

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

PlantUML > Language specification > State Diagram >

sponsors 20

patreon 122

liberapay 10

paypal 296

1000000

Диаграмма состояний

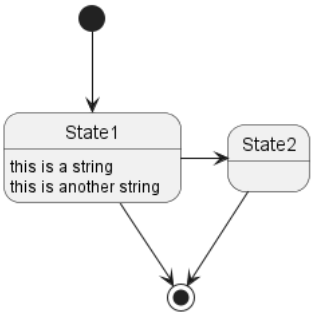
Простое состояние

Для изображения начального и конечного псевдосостояний используется [*] .

Используйте --> для изображения переходов.

```
@startuml
[*] --> State1
State1 --> [*]
State1 : this is a string
State1 : this is another string

State1 -> State2
State2 --> [*]
@enduml
```



- Простое состояние
- Change state rendering
- Составное состояние
- Длинные имена
- History [[H], [H*]]
- Ветвление [fork, join]
- Параллельные состояния
- Условия [choice]
- Stereotypes full example [start, choice, fork, join, end]
- Point [entryPoint, exitPoint]
- Pin [inputPin, outputPin]
- Expansion [expansionInput, expansionOutput]
- Направления стрелок
- Изменение цвета и отображения
- Заметки
- Заметки о переходах
- Еще о заметках
- Inline color
- Skinparam
- Changing style
- Change state color and style (inline style)
- Alias
- Display JSON Data on State diagram

Change state rendering

You can use hide empty description to render state as simple box.

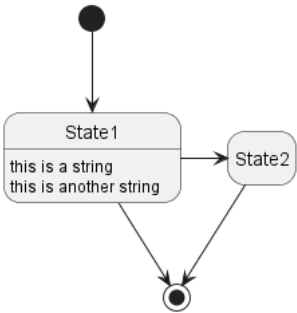


-
-

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
hide empty description
[*] --> State1
State1 --> [*]
State1 : this is a string
State1 : this is another string

State1 -> State2
State2 --> [*]
@enduml
```



Составное состояние

Также можно изображать составные состояния. Для этого его следует объявить, используя конструкцию `state`.

Внутренние состояния

-
-

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

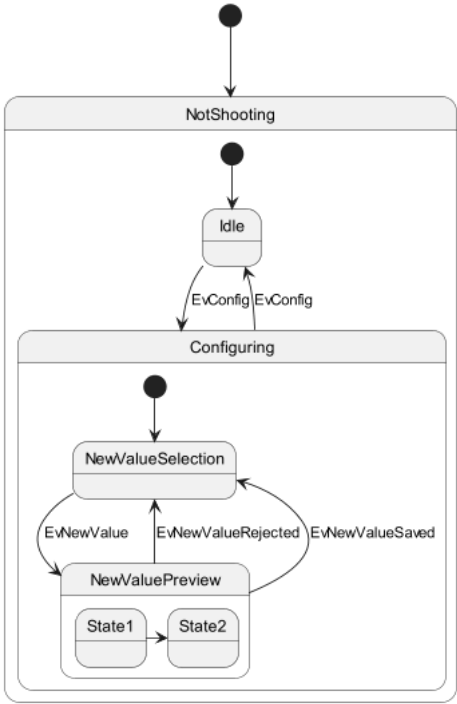
```
scale 350 width
[*] --> NotShooting

state NotShooting {
  [*] --> Idle
  Idle --> Configuring : EvConfig
  Configuring --> Idle : EvConfig
}

state Configuring {
  [*] --> NewValueSelection
  NewValueSelection --> NewValuePreview : EvNewVa
  NewValuePreview --> NewValueSelection : EvNewVa
  NewValuePreview --> NewValueSelection : EvNewVa
}

state NewValuePreview {
  State1 --> State2
}

}
@enduml
```



Внутреннее состояние к внутреннему состоянию

```
@startuml
state A {
  state X {
  }
  state Y {
  }
}

state B {
  state Z {
  }
}

X --> Z
Z --> Y
@enduml
```

[Ref. [QA-3300](#)]



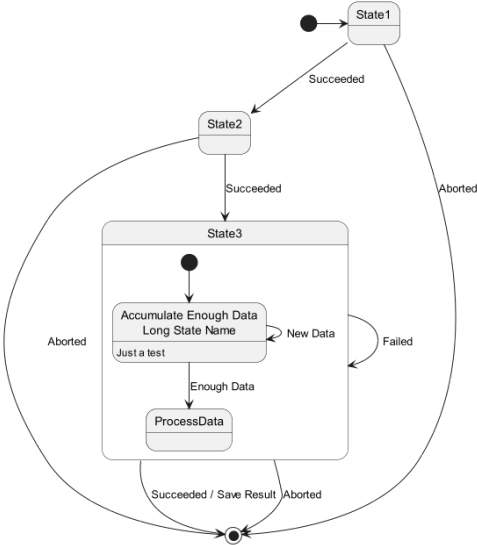
Длинные имена

Вы также можете использовать ключевое слово `state` для сокращения длинного имени состояния.

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
scale 600 width
[*] -> State1
State1 --> State2 : Succeeded
State1 --> [*] : Aborted
State2 --> State3 : Succeeded
State2 --> [*] : Aborted
state State3 {
  state "Accumulate Enough Data\nLong State Name"
  long1 : Just a test
  [*] --> long1
  long1 --> long1 : New Data
  long1 --> ProcessData : Enough Data
}
State3 --> State3 : Failed
State3 --> [*] : Succeeded / Save Result
State3 --> [*] : Aborted

@enduml
```

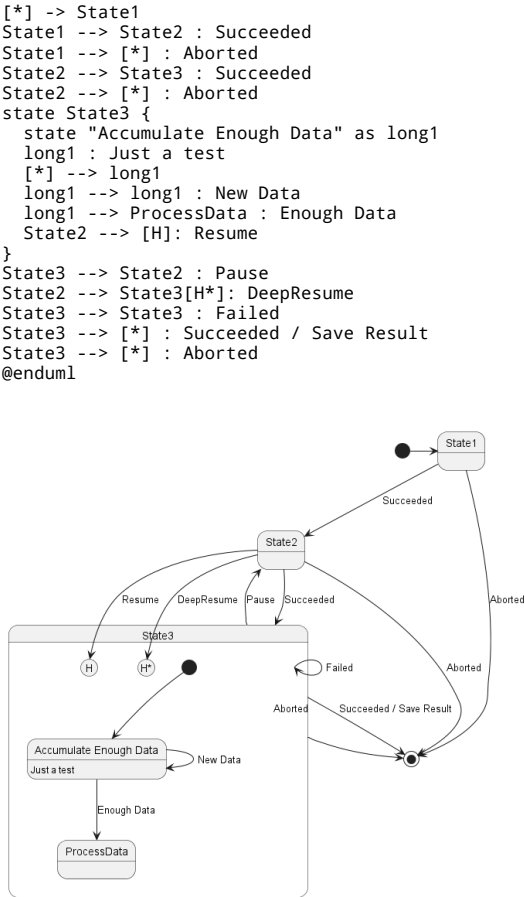


History **[[H], [H*]]**

You can use **[H]** for the history and **[H*]** for the deep history of a substate.

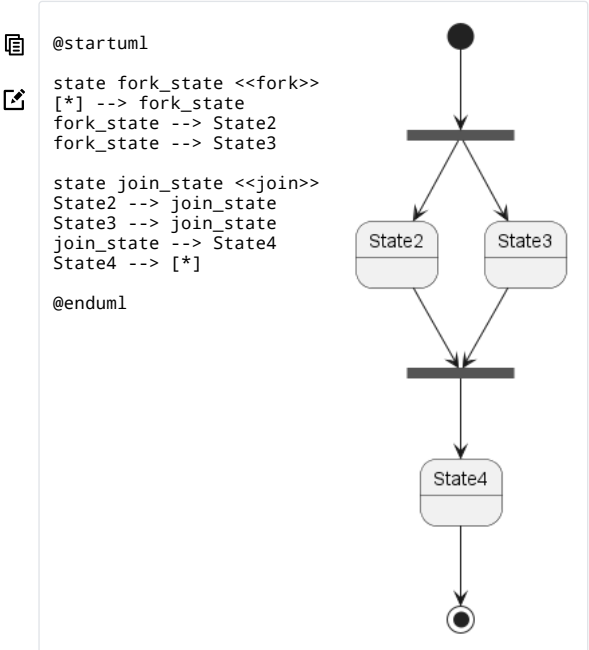


- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide



Ветвление [fork, join]

Также можно использовать стереотипы `<<fork>>` и `<<join>>` для создания ветвления и слияния обратно.



- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

Используя -- или ||, вы можете объявлять параллельные подсостояния внутри составного состояния.

Горизонтальное разделение --

```
@startuml
[*] --> Active

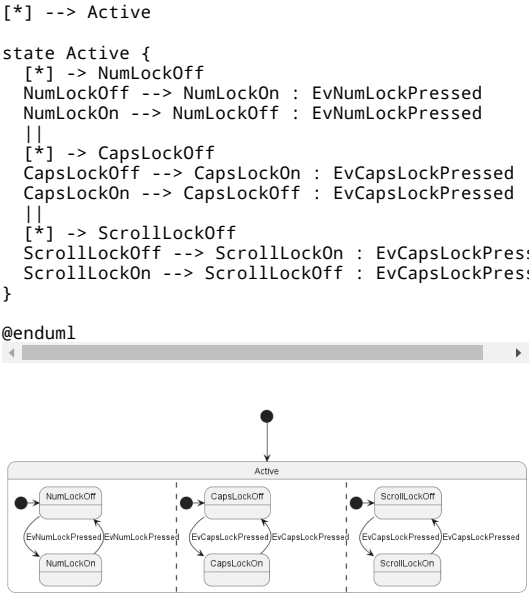
state Active {
    [*] -> NumLockOff
    NumLockOff --> NumLockOn : EvNumLockPressed
    NumLockOn --> NumLockOff : EvNumLockPressed
    --
    [*] -> CapsLockOff
    CapsLockOff --> CapsLockOn : EvCapsLockPressed
    CapsLockOn --> CapsLockOff : EvCapsLockPressed
    --
    [*] -> ScrollLockOff
    ScrollLockOff --> ScrollLockOn : EvCapsLockPressed
    ScrollLockOn --> ScrollLockOff : EvCapsLockPressed
}

@enduml
```

Вертикальное разделение ||

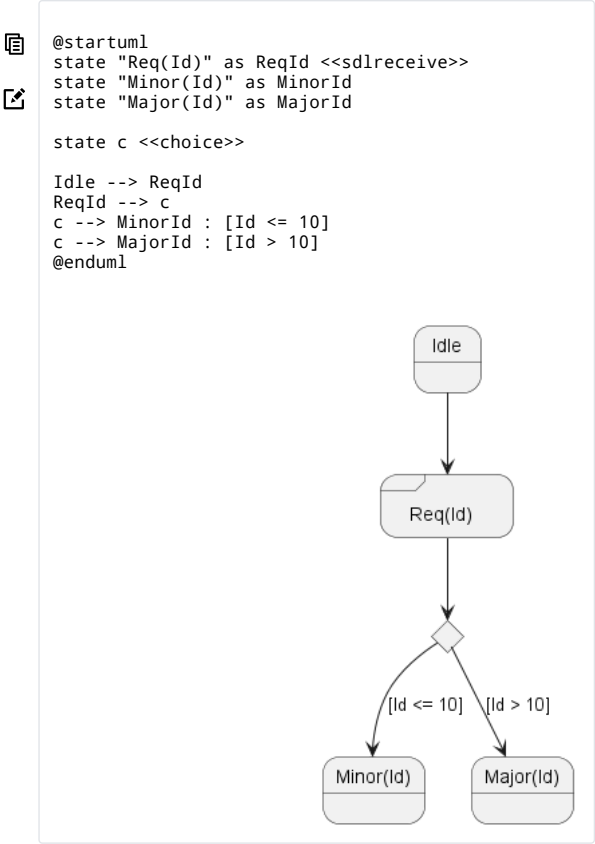
-
-

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide



Условия [choice]

Стереотип <<choice>> может быть использован для создания состояний-условий.



Stereotypes full example [start, choice, fork, join, end]

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
state start1 <<start>>
state choice1 <<choice>>
state fork1 <<fork>>
state join2 <<join>>
state end3 <<end>>

[*] --> choice1 : from start\nto choice
start1 --> choice1 : from start stereo\nto choice

choice1 --> fork1 : from choice\nto fork
choice1 --> join2 : from choice\nto join
choice1 --> end3 : from choice\nto end stereo

fork1 ---> State1 : from fork\nto state
fork1 ---> State2 : from fork\nto state

State2 --> join2 : from state\nto join
State1 --> [*] : from state\nto end

join2 --> [*] : from join\nto end
@endum1
```

[Ref. [QA-404](#), [QA-1159](#) and [GH-887](#)]

Point [entryPoint, exitPoint]

You can added **point** with <<entryPoint>> and <<exitPoint>> stereotypes:

-
-

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
state Somp {
  state entry1 <<entryPoint>>
  state entry2 <<entryPoint>>
  state sin
  entry1 --> sin
  entry2 -> sin
  sin -> sin2
  sin2 --> exitA <<exitPoint>>
}
```

```
[*] --> entry1
exitA --> Foo
Foo1 -> entry2
@enduml
```

Pin [inputPin, outputPin]

You can added **pin** with <<inputPin>> and <<outputPin>> stereotypes:

-
-

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
state Somp {
  state entry1 <<inputPin>>
  state entry2 <<inputPin>>
  state sin
  entry1 --> sin
  entry2 -> sin
  sin -> sin2
  sin2 --> exitA <<outputPin>>
}
```

```
[*] --> entry1
exitA --> Foo
Foo1 -> entry2
@endum1
```

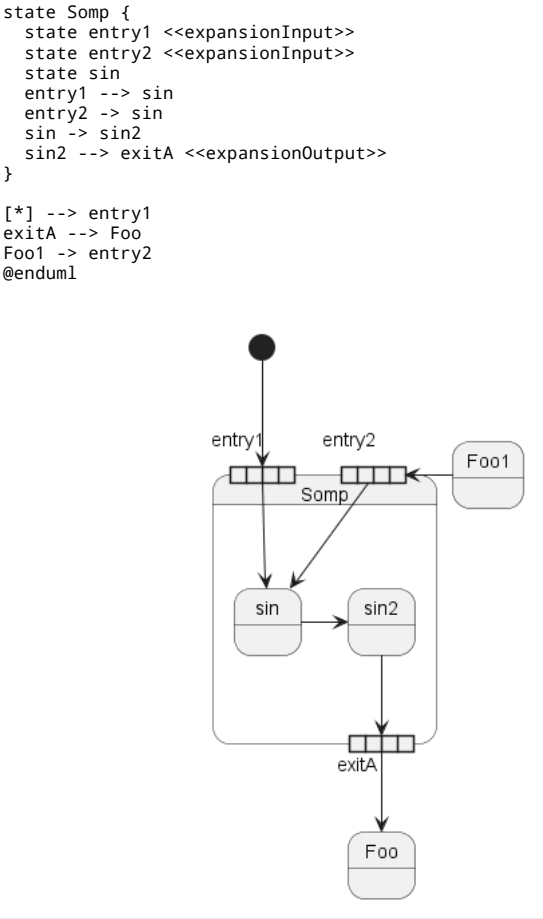
[Ref. [QA-4309](#)]

Expansion [expansionInput, expansionOutput]

You can added **expansion** with <<expansionInput>> and <<expansionOutput>> stereotypes:



- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

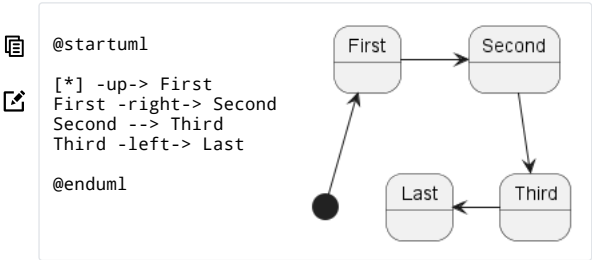


[Ref. [QA-4309](#)]

Направления стрелок

Для изображения стрелок перехода горизонтально используется оператор `->`. Следующий синтаксис позволяет задать другое направление.

- `-down->` или `-->`
- `-right->` или `->` (по умолчанию)
- `-left->`
- `-up->`



Вы также можете сокращать слова в описании стрелок (например, `-d->` или `-do->` вместо `-down->`).

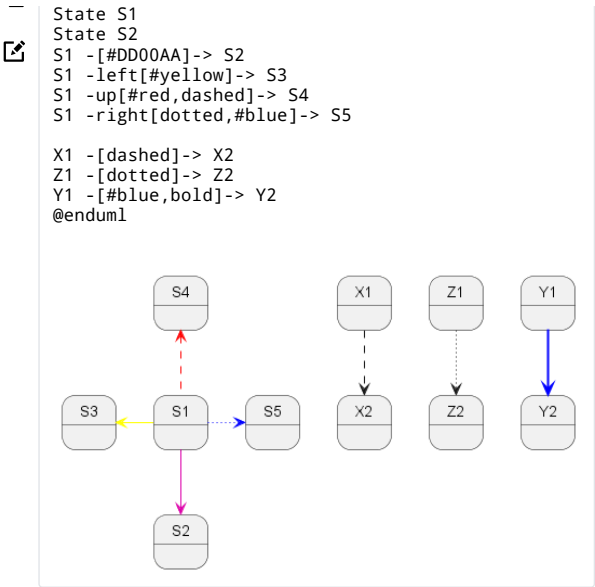
Не следует злоупотреблять этой функциональностью: *GraphViz* в большинстве случаев дает хороший результат без лишних манипуляций.

Изменение цвета и отображения

Можно использовать параметр [цвета](#) вместе со стилем.

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

последовательность прецедентов классов активности активности **UML** компонент состояний объ

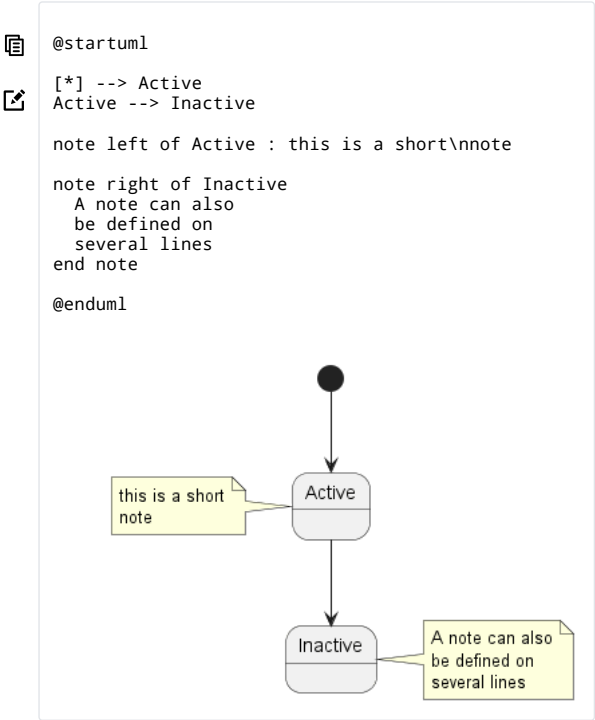


[Ref. [Incubation: Change line color in state diagrams](#)]

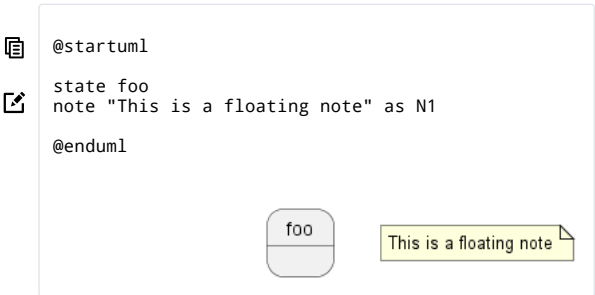
Заметки

К состоянию можно добавлять заметки, используя специальные ключевые слова: note left of , note right of , note top of , note bottom of .

Заметки можно определять в несколько строк.



Можно создавать заметки, не привязанные ни к какому объекту.



- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

Заметки о переходах

Можно оставлять заметки о переходах состояний или связей с помощью `note on link`.

@startuml
[*] -> State1
State1 --> State2
note on link
 Это заметка об изменении состояния
end note
@enduml

Еще о заметках

Также заметки можно прикреплять к составным состояниям.

@startuml
[*] --> NotShooting
state "Not Shooting State" as NotShooting {
 state "Idle mode" as Idle
 state "Configuring mode" as Configuring
 [*] --> Idle
 Idle --> Configuring : EvConfig
 Configuring --> Idle : EvConfig
}
note right of NotShooting : This is a note on a composite state
@enduml

Inline color

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
state CurrentSite #pink {
  state HardwareSetup #lightblue {
    state Site #brown
    Site -[hidden]-> Controller
    Controller -[hidden]-> Devices
  }
  state PresentationSetup{
    Groups -[hidden]-> PlansAndGraphics
  }
  state Trends #FFFF77
  state Schedule #magenta
  state AlarmSupression
}
@enduml
```

[Ref. [QA-1812](#)]

Skinparam

Вы можете использовать команду `skinparam` для изменения шрифтов и цветов диаграммы

Вы можете использовать данную команду :

- В определении диаграммы, как любую другую команду,
- В подключенном файле,
- В конфигурационном файле, указанном в командной строке в задании ANT.

Вы можете задавать цвета и шрифты для именованных шаблонов состояний.



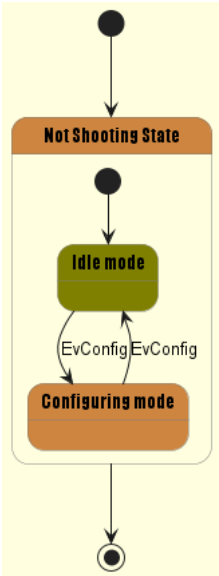
- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
skinparam backgroundColor LightYellow
skinparam state {
  StartColor MediumBlue
  EndColor Red
  BackgroundColor Peru
  BackgroundColor<<Warning>> Olive
  BorderColor Gray
  FontName Impact
}

[*] --> NotShooting

state "Not Shooting State" as NotShooting {
  state "Idle mode" as Idle <<Warning>>
  state "Configuring mode" as Configuring
  [*] --> Idle
  Idle --> Configuring : EvConfig
  Configuring --> Idle : EvConfig
}

NotShooting --> [*]
@enduml
```



WARNING This translation need to be updated. **WARNING**

Changing style

You can change [style](#).



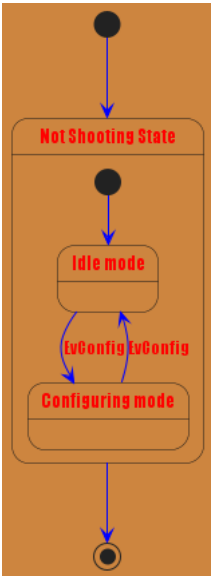
- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
<style>
stateDiagram {
  BackgroundColor Peru
  LineColor Gray
  FontName Impact
  FontColor Red
  arrow {
    FontSize 13
    LineColor Blue
  }
}
</style>

[*] --> NotShooting

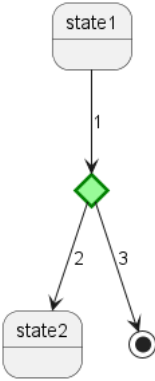
state "Not Shooting State" as NotShooting {
  state "Idle mode" as Idle <<Warning>>
  state "Configuring mode" as Configuring
  [*] --> Idle
  Idle --> Configuring : EvConfig
  Configuring --> Idle : EvConfig
}

NotShooting --> [*]
@enduml
```



```
@startuml
<style>
diamond {
  BackgroundColor #palegreen
  LineColor #green
  LineThickness 2.5
}
</style>
state state1
state state2
state choice1 <<choice>>
state end3 <<end>>

state1 --> choice1 : 1
choice1 --> state2 : 2
choice1 --> end3 : 3
@enduml
```



[Ref. [GH-880](#)]

Change state color and style (inline style)

You can change the color or style of individual state using the following notation:

- #color ##[style]color

With background color first (#color), then line style and line color (##[style]color).

- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

последовательность прецедентов классов активности активности **state** компонент состояний обь

```
state FooGradient #red-green ##00FFFF
state FooDashed #red|green ##[dashed]blue {
}
state FooDotted ##[dotted]blue {
}
state FooBold ##[bold] {
}
state Foo1 ##[dotted]green {
state inner1 ##[dotted]yellow
}

state out ##[dotted]gold

state Foo2 ##[bold]green {
state inner2 ##[dotted]yellow
}
inner1 -> inner2
out -> inner2
@enduml1
```

[Ref. [QA-1487](#)]

- #color;line:color;line.
[bold|dashed|dotted];text:color

FIXME text:color seems not to be taken into account **FIXME**

```
@startuml
@startuml
state FooGradient #red-green;line:00FFFF
state FooDashed #red|green;line.dashed;line:blue {
}
state FooDotted #line.dotted;line:blue {
}
state FooBold #line.bold {
}
state Foo1 #line.dotted;line:green {
state inner1 #line.dotted;line:yellow
}

state out #line.dotted;line:gold

state Foo2 #line.bold;line:green {
state inner2 #line.dotted;line:yellow
}
inner1 -> inner2
out -> inner2
@enduml1
@enduml1
```

```
@startuml
state s1 : s1 description
state s2 #pink;line:red;line.bold;text:red : s2 description
state s3 #palegreen;line:green;line.dashed;text:green : s3 description
state s4 #aliceblue;line:blue;line.dotted;text:blue : s4 description
@enduml1
```

[Adapted from [QA-3770](#)]



Alias

With State you can use `alias` , like:



- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

последовательность прецедентов классов активности активности **UML** компонент состояний объ

```
state alias1
state "alias2"
state "long name" as alias3
state alias4 as "long name"

alias1 : ""state alias1""
alias2 : ""state "alias2""""
alias3 : ""state "long name" as alias3""
alias4 : ""state alias4 as "long name""""

alias1 -> alias2
alias2 -> alias3
alias3 -> alias4
@enduml
```

or:

```
@startuml
state alias1 : ""state alias1""
state "alias2" : ""state "alias2""""
state "long name" as alias3 : ""state "long name"
state alias4 as "long name" : ""state alias4 as "
.

alias1 -> alias2
alias2 -> alias3
alias3 -> alias4
@enduml
```

Display JSON Data on State diagram

Simple example

```
@startuml
state "A" as stateA
state "C" as stateC {
state B
}

json jsonJ {
"fruit": "Apple",
"size": "Large",
"color": ["Red", "Green"]
}
@enduml
```

[Ref. [QA-17275](#)]

For another example, see on [JSON page](#).