

# monitorp

*Monitor protocol.*

author:

Paulo Moura

version:

1.0

date:

2000/7/24

compilation:

static, context\_switching\_calls, events

*(no dependencies on other files)*

## Public interface

### monitor\_activated/0

*True if monitor is currently active.*

compilation:

static

mode - number of solutions:

monitor\_activated - zero\_or\_one

### activate\_monitor/0

*Activates all spy points and start monitoring.*

compilation:

static

mode - number of solutions:

activate\_monitor - one

### suspend\_monitor/0

*Suspends monitoring, deactivating all spy points.*

compilation:

static

mode - number of solutions:

suspend\_monitor - one

### reset\_monitor/0

*Resets monitor, deactivating and deleting all spy points.*

compilation:

static

mode - number of solutions:

reset\_monitor - one

### spy\_point/4

*Current spy point.*

compilation:

static

template:

```
spy_point(Event, Object, Message, Sender)
```

mode - number of solutions:

```
spy_point(?event, ?object, ?callable, ?object) - zero_or_more
```

### **set\_spy\_point/4**

*Sets a spy point.*

compilation:

```
static
```

template:

```
set_spy_point(Event, Object, Message, Sender)
```

mode - number of solutions:

```
set_spy_point(?event, ?object, ?callable, ?object) - one
```

### **del\_spy\_points/4**

*Deletes all matching spy points.*

compilation:

```
static
```

template:

```
del_spy_points(Event, Object, Message, Sender)
```

mode - number of solutions:

```
del_spy_points(@event, @object, @callable, @object) - one
```

## **Protected interface**

*(none)*

## **Private predicates**

*(none)*