



## Role

The receiver daemon receives passive check data and serves as a distributed passive command buffer that will be read by the arbiter daemon. There can be many receivers for load-balancing and hot standby spare roles. The receiver can also use modules to accept data from different protocols.

- Module for passive data collection: WS arbiter module

## Other daemon communications

The arbiter will get data from the receiver each seconds.

## Data

The receiver keep in memory a buffer of external commands. Theses external commands have host and checks names.

## Receiver connexion summary

None.

## Variable Descriptions

Property	Default	Description
receiver_name	N/A	This variable is used to identify the *short name* of the receiver which the data is associated with.
address	N/A	This directive is used to define the address from where the main arbiter can reach this receiver. This can be a DNS name or a IP address.
port	7773	This directive is used to define the TCP port used by the daemon.
spare	0	This variable is used to define if the receiver must be managed as a spare one (will take the conf only if a master failed). The default value is *0* (master).
realm	N/A	This variable is used to define the realm where the receiver will be put. If none is selected, it will be assigned to the default one.
direct_routing	0	If enabled, it will directly send commands to the schedulers if it knows about the hostname in the command.
modules	N/A	This variable is used to define all modules that the receiver will load.

## Example Definition

```
define receiver {
    receiver_name    receiver-master
    address          localhost
    port            7773
    spare           0
    timeout          3 ; Ping timeout
    data_timeout    120 ; Data send timeout
    max_check_attempts 3 ; If ping fails N or more, then the node is dead
    check_interval  60 ; Ping node every N seconds
    modules
    use_ssl         0
    hard_ssl_name_check 0
    direct_routing  0
    realm          All
}
```