

# object's

Definition der object's mit Optionen

## Host

Beispiel:

```
object Host NodeName {  
  display_name = "Local host on this node"  
  address = "127.0.0.1"  
  address6 = "::1"  
  
  groups = [ "all-hosts" ]  
  
  check_command = "hostalive"  
}
```

Name	Optional	Beschreibung
display_name	ja	A short description of the host (e.g. displayed by external interfaces instead of the name if set).
address	ja	The host's address. Available as command runtime macro \$address\$ if set.
address6	ja	The host's address. Available as command runtime macro \$address6\$ if set.
groups	ja	A list of host groups this host belongs to.
vars	ja	A dictionary containing custom attributes that are specific to this host.
check_command		The name of the check command.
max_check_attempts	ja	The number of times a host is re-checked before changing into a hard state. Defaults to 3.
check_period	ja	The name of a time period which determines when this host should be checked. Not set by default.
check_interval	ja	The check interval (in seconds). This interval is used for checks when the host is in a HARD state. Defaults to 5 minutes.
retry_interval	ja	The retry interval (in seconds). This interval is used for checks when the host is in a SOFT state. Defaults to 1 minute.
enable_notifications	ja	Whether notifications are enabled. Defaults to true.
enable_active_checks	ja	Whether active checks are enabled. Defaults to true.
enable_passive_checks	ja	Whether passive checks are enabled. Defaults to true.
enable_event_handler	ja	Enables event handlers for this host. Defaults to true.
enable_flapping	ja	Whether flap detection is enabled. Defaults to false.
enable_perfdata	ja	Whether performance data processing is enabled. Defaults to true.
event_command	ja	The name of an event command that should be executed every time the host's state changes or the host is in a SOFT state.

Name	Optional	Beschreibung
flapping_threshold	ja	The flapping threshold in percent when a host is considered to be flapping.
volatile	ja	The volatile setting enables always HARD state types if NOT-OK state changes occur.
zone	ja	The zone this object is a member of.
command_endpoint	ja	The endpoint where commands are executed on.
notes	ja	Notes for the host.
notes_url	ja	Url for notes for the host (for example, in notification commands).
action_url	ja	Url for actions for the host (for example, an external graphing tool).
icon_image	ja	Icon image for the host. Used by external interfaces only.
icon_image_alt	ja	Icon image description for the host. Used by external interface only.

## Hostgroup

Beispiel:

```
object HostGroup "my-hosts" {
    display_name = "My hosts"
}
```

Name	Optional	Beschreibung
display_name	ja	A short description of the host group.
groups	ja	An array of nested group names.

## Notification

Beispiel:

```
object Notification "localhost-ping-notification" {
    host_name = "localhost"
    service_name = "ping4"

    command = "mail-notification"

    users = [ "user1", "user2" ]

    types = [ Problem, Recovery ]
}
```

Name	Optional	Beschreibung
host_name		The name of the host this notification belongs to.
service_name	ja	The short name of the service this notification belongs to. If omitted this notification object is treated as host notification.
vars	ja	A dictionary containing custom attributes that are specific to this notification object.

Name	Optional	Beschreibung
users	ja	A list of user names who should be notified.
user_groups	ja	A list of user group names who should be notified.
times	ja	A dictionary containing begin and end attributes for the notification.
command		The name of the notification command which should be executed when the notification is triggered.
interval	ja	The notification interval (in seconds). This interval is used for active notifications. Defaults to 30 minutes. If set to 0, re-notifications are disabled.
period	ja	The name of a time period which determines when this notification should be triggered. Not set by default.
zone	ja	The zone this object is a member of.
types	ja	A list of type filters when this notification should be triggered. By default everything is matched.
states	ja	A list of state filters when this notification should be triggered. By default everything is matched.

## Service

Beispiel:

```
object Service "uptime" {
    host_name = "localhost"

    display_name = "localhost Uptime"

    check_command = "check_snmp"

    vars.community = "public"
    vars.oid = "DISMAN-EVENT-MIB::sysUpTimeInstance"

    check_interval = 60s
    retry_interval = 15s

    groups = [ "all-services", "snmp" ]
}
```

display_name	ja	A short description of the service.
host_name		The host this service belongs to. There must be a Host object with that name.
name		The service name. Must be unique on a per-host basis (Similar to the service_description attribute in Icinga 1.x).
groups	ja	The service groups this service belongs to.
vars	ja	A dictionary containing custom attributes that are specific to this service.
check_command		The name of the check command.
max_check_attempts	ja	The number of times a service is re-checked before changing into a hard state. Defaults to 3.
check_period	ja	The name of a time period which determines when this service should be checked. Not set by default.

check_interval	ja	The check interval (in seconds). This interval is used for checks when the service is in a HARD state. Defaults to 5 minutes.
retry_interval	ja	The retry interval (in seconds). This interval is used for checks when the service is in a SOFT state. Defaults to 1 minute.
enable_notifications	ja	Whether notifications are enabled. Defaults to true.
enable_active_checks	ja	Whether active checks are enabled. Defaults to true.
enable_passive_checks	ja	Whether passive checks are enabled. Defaults to true.
enable_event_handler	ja	Enables event handlers for this host. Defaults to true.
enable_flapping	ja	Whether flap detection is enabled. Defaults to false.
enable_perfdata	ja	Whether performance data processing is enabled. Defaults to true.
event_command	ja	The name of an event command that should be executed every time the service's state changes or the service is in a SOFT state.
flapping_threshold	ja	The flapping threshold in percent when a service is considered to be flapping.
volatile	ja	The volatile setting enables always HARD state types if NOT-OK state changes occur.
zone	ja	The zone this object is a member of.
command_endpoint	ja	The endpoint where commands are executed on.
notes	ja	Notes for the service.
notes_url	ja	Url for notes for the service (for example, in notification commands).
action_url	ja	Url for actions for the service (for example, an external graphing tool).
icon_image	ja	Icon image for the service. Used by external interfaces only.
icon_image_alt	ja	Icon image description for the service. Used by external interface only.

## ServiceGroup

Beispiel:

```
object ServiceGroup "snmp" {
    display_name = "SNMP services"
}
```

Name	Optional	Beschreibung
display_name	ja	A short description of the service group.
groups	ja	An array of nested group names.

## TimePeriod

Beispiel:

```
object TimePeriod "24x7" {
    import "legacy-timeperiod"

    display_name = "Icinga 2 24x7 TimePeriod"

    ranges = {
        monday = "00:00-24:00"
```

```

    tuesday = "00:00-24:00"
    wednesday = "00:00-24:00"
    thursday = "00:00-24:00"
    friday = "00:00-24:00"
    saturday = "00:00-24:00"
    sunday = "00:00-24:00"
  }
}

```

Name	Optional	Beschreibung
display_name	ja	A short description of the time period.
update		The „update“ script method takes care of updating the internal representation of the time period. In virtually all cases you should import the „legacy-timeperiod“ template to take care of this setting.
ranges		A dictionary containing information which days and durations apply to this timeperiod.

## User

### Beispiel:

```

object User "icingaadmin" {
  display_name = "Icinga 2 Admin"
  groups = [ "icingaadmins" ]
  email = "icinga@localhost"
  pager = "icingaadmin@localhost.localdomain"

  period = "24x7"

  states = [ OK, Warning, Critical, Unknown ]
  types = [ Problem, Recovery ]

  vars.additional_notes = "This is the Icinga 2 Admin account."
}

```

Name	Optional	Beschreibung
display_name	ja	A short description of the user.
email	ja	An email string for this user. Useful for notification commands.
pager	ja	A pager string for this user. Useful for notification commands.
vars	ja	A dictionary containing custom attributes that are specific to this user.
groups	ja	An array of group names.
enable_notifications	ja	Whether notifications are enabled for this user.
period	ja	The name of a time period which determines when a notification for this user should be triggered. Not set by default.
types	ja	A set of type filters when this notification should be triggered. By default everything is matched.
states	ja	A set of state filters when this notification should be triggered. By default everything is matched.

# UserGroup

Beispiel:

```
object UserGroup "icingaadmins" {  
    display_name = "Icinga 2 Admin Group"  
}
```

Name	Optional	Beschreibung
display_name	ja	A short description of the user group.
groups	ja	An array of nested group names.

## Quellen

- <http://docs.icinga.org/icinga2/snapshot/doc/module/icinga2/toc#!/icinga2/snapshot/doc/module/icinga2/chapter/object-types>

```
sed -e 's/\t//g' text | sed -e 's/ Optional. /| ja |/' | sed -e 's/  
Required. /| |/' | sed -e 's/^/|/' | sed -e 's/$/|/' | sed -e 'li\^Name^  
Optional ^Beschreibung^'
```

From:

<https://wiki.da-checka.de/> - **PSwiki**

Permanent link:

<https://wiki.da-checka.de/doku.php/wiki/programme/icinga/object-definition>

Last update: **2015/07/08 16:34**

