

Was ist neu in Nagios 3.0?

Oldenburg, 2008-06-11

Änderungen bei Objekten

- Gleitkomma-Intervalle
- Einzelne Objekte
 - `host, service`
 - `*groups`
 - `contact`
 - `timeperiod`
 - `*dependency`
 - `*escalation`
- Vererbung

Objekte: Intervalle

- Gleitkoma-Intervalle möglich:
 - `check_interval 2.5`
 - `2,5 * interval_length` (meist 60 sec.)

Nagios 2.x		Nagios 3.0
Host	Service	Host/Service
<code>check_interval</code>	<code>normal_check_interval</code>	<code>check_interval</code>
<code>retry_interval</code>	<code>retry_check_interval</code>	<code>retry_interval</code>

Unter Nagios 3.0
weiterhin möglich

Objekte: host

```
display_name      für_zukünftiges_Webinterface
contacts          einzelner_kontakt,noch_einer
...
first_notification_delay 20
flap_detection_options  o,d,u
notification_options    d,u,r,f,s
initial_state         o,d,u
retry_interval        3 # siehe Host-Logik
```

...

+hostextinfo-Parameter

alias ist optional (default: host_name)

Objekte: service (1)

display_name für_zukünftiges_Webinterface
contacts einzelner_kontakt,noch_einer

...

first_notification_delay 20

flap_detection_options o,c,w,u

notification_options w,u,c,r,f,**s**

initial_state o,w,c,u

check_interval 5

retry_interval 3

...

+serviceextinfo-Parameter

Objekte: service (2)

- jetzt optional:
 - `contact_groups`
 - `notification_interval`
 - `notification_period`
 - werden bei Bedarf vom Host geerbt
- Tipp:
 - Verantwortlichkeiten (`contact_groups`) und Notifications generell beim Host setzen
- Parameter fällt weg:
 - `parallelize_checks`
 - Service-Checks in 3.0 immer parallel

Objekte: *_group

```
define hostgroup {
    hostgroup_members hg1,hg2,hg3

define servicegroup {
    servicegroup_members sg1,sg2,sg3

define contactgroup {
    contactgroup_members cg1,cg2,cg3

... alle ...
    notes          irgendwelchersinnvollertext
    notes_url      url
    action_url     url
}
```

Objekte: contact

```
host_notification_enabled      1
service_notification_enabled   1
can_submit_commands            1
```

- mit den beiden `_enabled`-Parametern ist auch die (De-)Aktivierung über die Schnittstelle möglich:
 - `DISABLE_CONTACT_HOST_NOTIFICATION`
 - `DISABLE_CONTACT_SVC_NOTIFICATION`
 - dito mit Kontaktgruppe

Objekte: timeperiod

2008-06-09 00:00-24:00

june 1 00:00-24:00

day 2 00:00-24:00

monday 1 00:00-24:00

monday 2 june 00:00-24:00

monday 00:00-24:00

2008-06-09 - 2008-06-12 00:00-24:00

2008-06-09 - 2008-06-15 / 2 00:00-24:00

exclude tp1, tp2, tp3

Objekte: *dependency

- host-, servicedependency:
 - dependency_period *timeperiod*

- *Same Host Dependency:*

```
define servicedependency {  
    host_name                1x01  
    service_description      Memory  
    dependent_service_description NRPE  
    ...  
}
```

- dependent_host_name weglassen

Objekte: *escalation

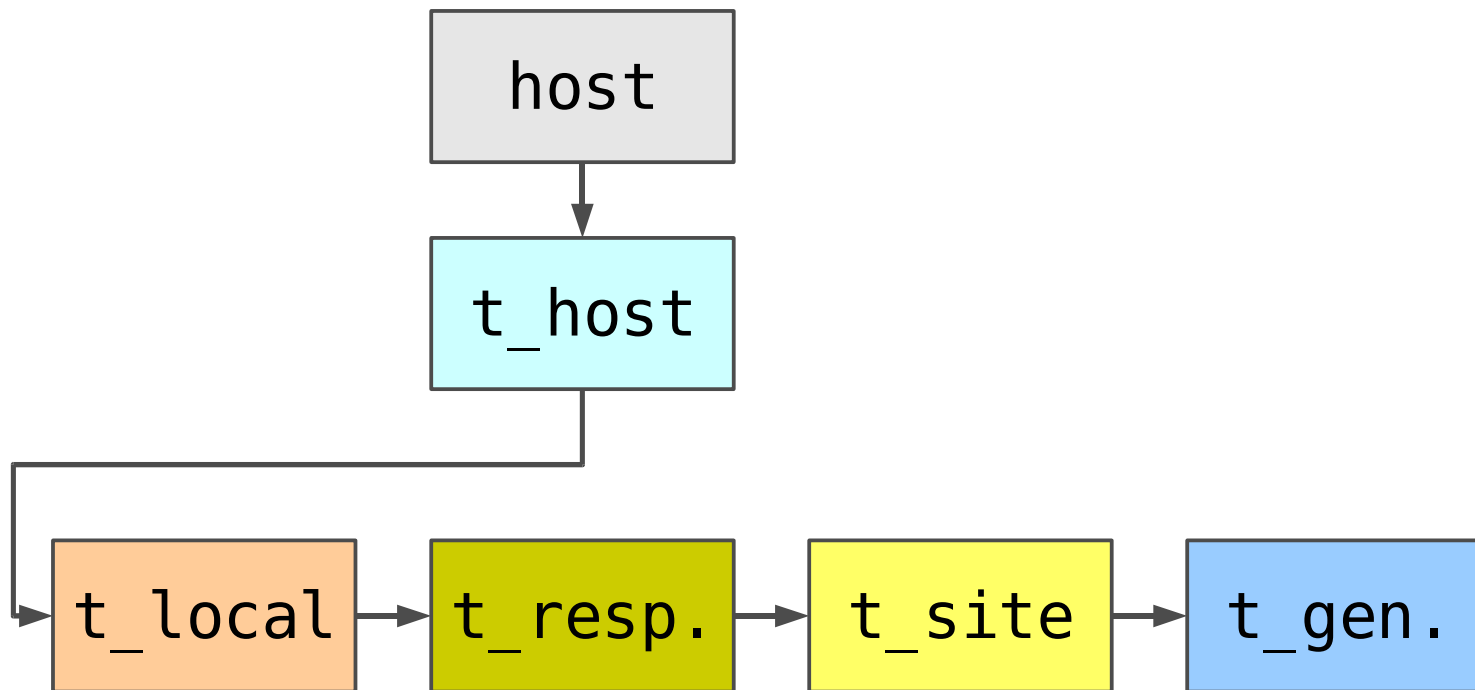
- erlauben jetzt die direkte Angabe von Kontakten:

```
define service_escalation {  
    ...  
    contacts    c1,c2,c3  
    ...  
}
```

Vererbung (1)

```
define host{
    use          template-host-ahz
    host_name    AHDC01
    alias        AHDC01 (1.Domaincontroller AH )
    address      172.17.129.160
    hostgroups   +eliWINDOWS,eliADS,eliDISK_D
    parents      eli-sw01
}
define host{
    name         template-host-ahz
    use          host_local_t,host_responsible_ahz_t,\
                host_site_ahz_t,host_generic_t
    register 0
}
```

Vererbung (1.5) ;-)



Erster Treffer zählt

Vererbung (2)

```
define host{
    name                                host_generic_t
    ...
    check_period                        24x7
    check_interval                      5
    retry_interval                      1
    max_check_attempts                 5

    first_notification_delay           15

    hostgroups                          UNDEFINED
    notes_url http://mynagios/dokuwiki/\
        doku.php?id=nagios:hosts:$HOSTNAME$
}
```

Vererbung (3)

```
define host{
    name                host_site_ahz_t
    # -- site group
    hostgroups          AH,AH_Zentrale
    ...
    # -- notifications
    notifications_enabled 1
    notification_interval 480
    notification_period   24x7
    notification_options  d,r
    # -- custom variables
    _NSCLIENT_PORT       1248
    _PING_PARAM           -w 200.0,20% -c 500.0,40%
}
```

Vererbung (4)

```
define host{
    name                host_responsible_ahz_t
    register            0
    # -- responsibility
    contact_groups     eli-admins,mhk-admins
}
```

```
define host{
    name                host_local_t
    register            0
}
```

Vererbung (5)

```
define service{
    host_name                elix01,mintra
    service_description      lx-Load
    normal_check_interval    1
    check_command             check_nrpe!check_load

    use    service_perfdata_t,template-service-eli
}
define service{
    name                    service_perfdata_t
    register                0
    process_perf_data      1
    action_url              /nagios/pnp/index.php?\
                            host=$HOSTNAME&srv=$SERVICEDESC$
}
}
```

Custom Macros (1)

- möglich bei host, service, contact
- werden wie normale Variablen erlaubt
- beginnen mit einem Unterstrich, z.B.
 `_NSCLIENT_PORT 12489`
 `_PING_PARAM -w 200.0,20% -c 500.0,40%`
- Referenzierung:
 `$_HOSTNSCLIENT_PORT`
 `$_HOSTPING_PARAM`
 dito mit Service und Kontakt

Custom Macros (2)

```
define command{
    command_name check_nt
    command_line $USER1$/check_nt -H $HOSTADDRESS$\
        -p $_HOSTNSCLIENT_PORT$ -v $ARG1$ $ARG2$
}
define service{
    hostgroup_name      eliLINUX,eliWINDOWS
    service_description PING
    check_command       check_icmp!$_HOSTPING_PARAM$
    normal_check_interval 1
    use service_perfdata_t,template-service-eli
}
```

Makros

- viele Neuerung, zuviel für den Vortrag ;-)
- Makros kosten Performance (insbesondere Summenmakros), daher:
- `large_installation_tweaks = 1`
 - schaltet Summenmakros ab
- `enable_environment_macros = 0`
 - schaltet alle Umgebungsvariablen ab
 - Achtung: verschiedene Skripte sind auf das Environment angewiesen (... ganz abschalten ist daher nicht immer eine gute Idee ...)

Downtime, Comments, ACKs

- Speicherung zusammen mit Retention- und Statusinfos (keine eigenen Dateien mehr)
- Downtime-Notification: Option 's'
- Änderung bei non-persistent ACKs:
 - 2.x: löscht Kommentar beim Neustart
 - 3.0: löscht Kommentar, wenn Status wieder OK

Externe Kommandos

- Viele neue Kommandos
-> siehe Homepage
- **PROCESS_FILE**; *pfad; delete*
 - Verarbeitet Datei mit externen Kommandos
 - delete=1: Datei wird anschließend gelöscht
- **SEND_CUSTOM_SVC_NOTIFICATION**; \
host; svc; opts; name; comment
 - opts: Bitmaske
 - > 1=Broadcast an alle Kontakte + Eskalationen des betroffenen Services
 - > 2=erzwingen Versand, ignoriere Filter
 - > 4=erhöhe Benachrichtigungszähler (für Eskalationen)

Neue Host-Check-Logik

Host-Checks **aktiv** und **parallel**

- aktiv:
 - `active_host_checks_enabled 1`
 - `check_interval > 0`
- parallel:
 - `max_check_attempts > 1`
- caching:
 - `cached_host_check_horizon 15` (default)
- predictive:
 - `enable_predictive_host_dependency_checks 1`
- ... Und? Hilft es was? ...

Embedded Perl

- `configure --enable-embedded-perl`
- `nagios.cfg`:
 - `enable_embedded_perl=1`
 - `use_embedded_perl_implicitly=1`
- Plugin:
 - `# nagios: -epn`
 - Achtung: in den ersten 10 Zeilen!
- Performance:
 - 2000 Services, viel Perl: keine Probleme, wenn `large_installation_tweaks=1` gesetzt ist

Neustart

- Statistik:
 - nagios -s nagios.cfg
- Neustart:
 - nagios -vp nagios.cfg
 - /etc/init.d/nagios stop
 - nagios -udx nagios.cfg
 - > -v: verify
 - > -p: write precaching
 - > -u: read precaching
 - > -d: daemon
 - > -x: no circular checks
- ... Und? Nützt es was? ...

Erweiterte Plugin-Ausgabe

normale textausgabe | optionale perfddata

langtext, 1. zeile

langtext, 2. zeile

...

langtext, n. zeile | perfddata, 2. zeile

perfddata, 3. zeile

...

perfddata, n. zeile

\$HOSTOUTPUT\$, \$HOSTPERFDATA\$,
\$LONGHOSTOUTPUT\$

Weboberfläche

- `cgi.cfg`
 - `lock_author_names`
 - `escape_html_tags`
 - `action_url_target`
 - `notes_url_target`
- `status.cgi`
 - `hostprops, serviceprops:`
 - > 262144 = Hard State
 - > 524288 = Soft State

Upgrade 2.x auf 3.0

- Upgrade bei selbst übersetzten Nagios fast immer problemlos
- Mutige: im laufenden Betrieb
- Vorsichtige: außerhalb der Hauptkampfzeiten (z.B. am WE)
- Auf Nummer sicher: Testportierung
- im Notfall: make install der aktiven Version

Upgrade (2)

- Quellen wie gewohnt entpacken

```
cd /usr/local/src/nagios
tar xvzf /xyz/nagios-3.0.2.tar.gz
cd nagios-3.0.2
./configure
make all
```
- **NICHT** installieren
- Check der Konfiguration mit:

```
./base/nagios -v /etc/nagios/nagios.cfg
```

Upgrade (3)

- main.cfg:
 - `service_reaper_frequency` heißt jetzt `check_result_reaper_frequency`
 - `aggregate_status_updates` entfällt
- Makros:
 - `$NOTIFICATIONNUMBER$` ->
`$HOSTNOTIFICATIONNUMBER$`,
`$SERVICENOTIFICATIONNUMBER$`

Upgrade (4)

- Downtime (downtime_file):
 - Inhalt nach retention.dat kopieren
- Comments (comment_file):
 - Inhalt nach retention.dat kopieren
- Alle drei Dateien beginnen mit einem Info-Block, der aus den dazukopierten Daten entfernt werden muss:

```
info {  
    created=1234567890  
    version=2.0  
}
```

Upgrade (5)

```
/etc/init.d/nagios stop
```

```
cd /var/nagios
```

```
cat comments.dat downtime.dat >>  
retention.dat
```

```
# überflüssige Infoblocks rausnehmen
```

```
make install
```

```
make install-init
```

```
make install-commandmode
```