HOWTO zu Nagios System- und Netzwerk-Überwachung

(C) 2008-2016 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Open Source Training and Consulting GmbH
http://www.ostc.de

$Id: nagios-HOWTO.txt,v 1.124 2019/11/26 19:37:07 tsbirn Exp $

Dieses Dokument beschreibt Grundlagen und Einsatz von Nagios unter UNIX/Linux
(Überwachung von Ressourcen, Systemparametern, Diensten und Netzwerk).

INHALTSVERZEICHNIS

0) Allgemein wichtig
0.1) Konfiguration
0.2) Sicherheit
0.3) Zugriffsrechte
0.4) Hinweise zur Syntax in Konfigurations-Dateien
0.5) Schnelle Bedienung auf der Kommandozeile
1) Einführung
2) Zweck von "Monitoring"
3) Was ist überwachbar? -- ALLES!
4) Was bietet Nagios?
5) Vor- und Nachteile von Nagios
6) Anmerkungen zu Nagios
7) Architektur von Nagios
8) Web-GUI von Nagios
8.1) Zugriffssteuerung auf Web-GUI
9) Alarmierung (Benachrichtigungen)
10) Weitere Eigenschaften von Nagios
11) Was ist notwendig um Nagios laufen zu lassen?
12) Wie als Anfänger beginnen?
12.1) Nagios Dokumentation (Core Documentation)
13) Zum Verständnis von Nagios wichtige Punkte
13.1) Check-Typen
13.2) Status-Typ SOFT/HARD
13.3) Host-Abhängigkeit ("parents")
14) Konfiguration von Nagios
15) Warum keine GUI zur Konfiguration?
16) Kommunikationsarten Nagios-Server <-> Plugins (GRAFIK)
17) Check-Agent Konfiguration
17.1) SSH (check_by_ssh)
17.2) NRPE (Nagios Remote Plugin Executor)
17.3) NSCA (Nagios Service Check Acceptor)
17.4) NSCP (Nagios Service Check Protocol)
17.5) NWSC (Nagios Windows Service Checker)
18) Was ist bei Plugins zu beachten?
18.1) Plugin-Schnittstelle
19) Auswahl von Plugins
20) Standard-Plugins
21) Nagios Objekte und ihre Beziehungen
22) Konfigurations-Dateien -- Struktur
23) Konfigurations-Dateien -- Inhalt
23.1) Haupt-Konfigurations-Datei "nagios.cfg"
23.2) Resource-Konfigurations-Datei "resource.cfg"
23.3) GUI-Konfigurations-Datei "cgi.cfg"
23.4) Objekt-Konfigurations-Dateien
23.5) Vordefinierte Objekte
24) Benachrichtigungen
25) Makros
26) Event-Handler
27) Flap Detection
28) External Commands (externe Kommandos)
29) Performance-Daten mit PNP4Nagios
30) Logging
31) Debugging
32.1) Erweiterungen von Nagios 3 (26.2.2008)
32.2) Erweiterungen von Nagios 4 (20.9.2013)
33) Glossar
34) Links
34.0) Nagios Dokumentation
34.1) Deutsche Nagios Webseiten
34.2) Englische Nagios Webseiten
34.3) Nagios Bücher

34.4) Firmen mit Nagios-Expertise
34.5) Quellen für Nagios-Plugins
34.6) GUI/Web-Interfaces für Nagios
34.7) AddOns für Nagios
34.8) SNMP (Simple Network Management Protocol)
34.9) Mobile Device Interfaces für Nagios (iOS/Android Apps)
34.10) Nagios Distributionen
34.11) Nagios-Foren/Blogs
34.12) Nagios-basierte Alternativen ("Nagios Powered")
34.13) Nagios-Alternativen
34.14) Kommerzielle Monitoring-Produkte
34.15) Wikipedia Artikel zu Nagios
35) Hinweise und Tipps

---

0) Allgemein wichtig

0.1) Konfiguration

* Nagios-Verz.- und Kommando-Namen heißen je nach Nagios-Version:
  + nagios   # Allgemein V1/V2
  + nagios2   # V2-Plugins (auch bei V3/V4)
  + nagios3   # V3
  + nagios4   # V4

* Die Haupt-Konfigdatei "nagios.cfg" liegt unter folgendem Verz.
  (enthält Verweise auf weitere einzulesende Konfig.-Dateien):
    /etc/nagios3   # Wenn aus Repository installiert
    /usr/local/nagios/etc   # Wenn aus Quellcode installiert

* VOR Änderungen IMMER aktuelle Konfigurations-Dateien SICHERN:
  + Für späteren Vergleich
  + Zum Zurückspielen funktionierender Version bei nicht behebbaren Fehlern
  + Am besten per Versionsverwaltung (RCS/CVS/SVN/GIT/...)
  + Gesamten Nagios-Konfig.Ordner "/etc/nagios3" (aus Repository inst.) oder
    "/usr/local/nagios/etc" (aus Quellcode inst.) in TAR/ZIP-Archiv sichern

      tar czvf nagios-backup.tgz /etc/nagios3
      tar czvf nagios-backup.tgz /usr/local/nagios/etc

* Nach Änderungen einen "PRE-FLIGHT CHECK" (als root!) durchführen (-v=verify),
  da keine GUI die korrekte Syntax sicherstellt (analog "testparm" bei Samba).
  Konfiguration nach JEDER Änderung "verifizieren" (nur Syntaxcheck),
  BEVOR Nagios-Dienst neu gestartet wird --> sonst Nagios-Betrieb unterbrochen!

    sudo /usr/sbin/nagios3   -v /etc/nagios3/nagios.cfg
    sudo /usr/local/nagios/sbin/nagios -v /usr/local/nagios/etc/nagios.cfg

* Erst nach erfolgreichem Test Nagios-Dienst neu starten!
      service nagios3 restart   # Kurzzeitige Unterbrechung (einige ms)
  + Konfigurations-Datei neu einlesen genügt meist (kill -HUP bzw. reload)
      service nagios3 reload   # Keine Unterbrechung
  + "restart" macht mehr als "reload" (z.B. Dateien anlegen und verrechten)
    - Manchmal notwendig!
    - Unterbricht aber kurzzeitig Nagios-Dienst!

* Log- und Debug-Dateien unbedingt auf Fehlermeldungen von Nagios überprüfen!
    /var/log/nagios/*   # Meldungen von Nagios-Dienst V2
    /var/log/nagios3/*   # Meldungen von Nagios-Dienst V3
    /var/log/nagios4/*   # Meldungen von Nagios-Dienst V4
    /var/log/syslog   # System-Meldungen
    /var/log/system   # Alle Log-Meldungen (abhängig vom System)
    /var/log/messages   # Alle Log-Meldungen (abhängig vom System)
    /var/log/allmessages   # Alle Log-Meldungen (abhängig vom System)
    /var/log/localmessages   # Alle Log-Meldungen (abhängig vom System)
    /var/log/auth.log   # Anmeldevorgänge und -versuche

* Plugins IMMER auf Kommandozeile als Benutzer "nagios" austesten!
  + Per User "nagios", NICHT als "root"!
      sudo -u nagios PLUGIN ...
  + Plugin-Usage-Meldung per -h/--help lesen
  + ALLE: Standard, Download, Selbstgeschriebene

* ERST kompletten Plugin-Aufruf inkl. Aussenrum (SSH, NRPE, ...) mit Parametern
  auf Kommandozeile zum Laufen bringen (testen bis funktionsfähig!),
  DANN in Nagios-Konfiguration einbauen und erneut testen bis funktionsfähig!

0.2) Sicherheit

* ZWEITEN KOMMUNIKATIONSWEG für Benachrichtigungen einrichten (z.B. per SMS
  über per USB angeschlossenes GSM-Modul), da bei Identität des zu
  überwachenden Netzwerks mit dem Versandnetzwerk der Benachrichtigungen
  Problemnachrichten nicht mehr verschickt werden können.

* "Principle of least privilege"
  + Nagios läuft ohne "root"-Rechte unter (normalem) Benutzer "nagios"
  + Plugins laufen somit unter Benutzer "nagios" (ohne "root"-Rechte)
  + Bei Bedarf kann per "sudo" auch Plugins das "root"-Recht gegeben werden
    (für Zugriff auf "/proc", "/sys", "/dev",
     für Kommando "nmap" zum Zugriff auf Netzwerkkarten,
     für Zugriff auf die privilegierten Ports 0-1023,
     für Kommando "smartctl" zum Zugriff auf Hardware).
    - Dazu notwendige Einträge in "/etc/sudoers":
      ```
      nagios ALL=(root) NOPASSWD: /usr/sbin/smartctl, /usr/bin/nmap, ...
      ```
    - Dazu notwendiger Aufruf des Plugins in Kommando-Definition:
      ```
      define command {
        command_name  check_smart
        command_line  /usr/bin/sudo \          # Zeilenumbruch maskiert
                      /usr/sbin/smartctl ...
      }
      ```

* NICHT als Benutzer "root" arbeiten, wenn nicht unbedingt notwendig!
  + Statt dessen "sudo" als normaler Benutzer verwenden
  + "/etc/sudoers" geeignet konfigurieren (Abfrage Benutzerpasswort!)
  + Eigenen Account in Gruppe "admin", "wheel" oder "sudo" aufnehmen

* Benutzer "root" KEIN Passwort geben (fehlt sowieso unter vielen Linuxen)
  (Umweg über Anmeldung als normaler Benutzer notwendig).

* Benutzer "root" KEINE (schlüsselbasierte) SSH-Anmeldung ermöglichen
  (Umweg über Anmeldung als normaler Benutzer notwendig).

* Besitzverhältnisse und Zugriffsrechte an Verz. und Dateien (ganz. Verz.pfad)
  vor und nach folgenden Operationen prüfen/vergleichen:
  + chmod
  + chown
  + chgrp
  + mv
  + cp
  + scp
  + tar
  + zip
  + ...

* Verschlüsselte Kommunikation zwischen Nagios Core und GUI + Plugins einrichten
  + SSL/TLS (HTTPS)
  + SSH-Tunnel
  + OpenVPN
  + stunnel
  + NRPE
  + NSCA
  + NSCP
  + ...

0.3) Zugriffsrechte

* Besitzer(gruppe) und Zugriffsrechte an Verz. und Dateien (ganz. Verz.pfad)
  so einstellen, dass Benutzer oder Gruppe "nagios" zugreifen kann (lesen und
  schreiben), Verz. müssen für "nagios" ausführbar sein, damit sie betreten/
  durchquert werden können.

* Set-Group-ID-Recht für Verz. setzen (chmod g+s ...), falls eine Datei darin
  bei jedem Dienst-Start neu angelegt wird. Damit wird die Besitzer-Gruppe
  des Verz. für die neu angelegte Datei übernommen.

* Auf bestimmte Verz./Dateien ist Zugriff von Nagios und dem Webserver (Apache)
  nötig, der unter Benutzer "www/wwwrun" + Gruppe "www-data" läuft.
  Daher diesen Benutzer in Gruppe "nagios" aufnehmen oder auf Verz./Datei
  nötige Rechte für "Other" setzen (chmod o+rx ... bzw. chmod o+r ...).

* HINWEIS: Nagios wird zwar mit "root"-Rechten gestartet,
  stellt aber zum Betrieb auf Benutzer "nagios" + Gruppe "nagios" um
  (siehe Variablen "nagios_user" + "nagios_group" in "nagios.cfg").
  D.h. "sudo nagios3 -v nagios.cfg" verwendet "root"-Rechte nur kurzzeitig!

0.4) Hinweise zur Syntax in Konfigurations-Dateien

* Mit "#" und ";" beginnende Zeilen sind KOMMENTAR und werden ignoriert
  --> Leerraum davor ist erlaubt
  --> Per "\#" bzw. "\;" schützen, wenn an externes Kommando übergeben
  --> Am Zeilenende nach Anweisung vermeiden (evtl. als Wert interpretiert)

* Variablennamen müssen am Zeilenanfang stehen (KEIN Leerraum davor erlaubt)

* Alle Namen sind CASE-SENSITIVE (d.h. GROSS/kleinschreibung wird beachtet)
  + Außer Objekt-Hostnamen (da DNS-Hostnamen das auch nicht machen)!

* Umlaute "äöüÄÖÜß", Sonderzeichen und Leerzeichen in Service-Descriptions und
  Kommando-Namen vermeiden, da sie als externe Namen verwendet werden und der
  Aufruf von der Nagios-GUI oder vom Nagios-Kern aus fehlschlagen kann
  (Zeichensatz von Editor und Linux/Nagios kann verschieden sein: UTF-8/Latin1)
  --> Nur A-Z a-z 0-9 _ - verwenden (BEZEICHNER)
  --> Auch alle andere Objektnamen sollten nur diese Zeichen enthalten (BEZEICHNER)

* Leerzeichen in Namen werden berücksichtigt
  --> keine Leerzeichen in durch "," getrennten Listen verwenden!

* Auf Tippfehler, Quotierungsfehler mit "...", '...', Klammerfehler, Leerräume
  und Abschlusszeichen ";" in Konfigurations-Dateien achten.

* Die Quotierungszeichen "..." und '...' haben in Nagios KEINE Bedeutung,
  werden aber beim Aufruf externer Kommandos an diese weitergegeben.
  D.h. damit können und müssen Sonderzeichen vor der Shell geschützt werden.
  Ebenso ist Backslash "\" evtl. notwendig zur Übergabe von Sonderzeichen.

* IMMER Hochkomma '...' um Parameter $ARG1$ in externen Kommandos setzen
  (wg. evtl. vork. Shell-Sonderzeichen)

* Lange Zeilen per "\" am Zeilenende auf mehrere Zeilen umbrechen
  + Z.B. externe Kommandos
  + Nur "Newline" (Zeilenvorschub) direkt danach erlaubt!

* Pfadnamen sind absolut oder relativ angebbar
  + Relative Pfadnamen sind relativ zum Startverz. der Haupt-Konfigdatei "nagios.cfg"

0.5) Schnelle Bedienung auf der Kommandozeile

* Befehle nicht ständig neu tippen, sondern wiederverwenden
  + History nutzen
  + Maus nutzen

* TAB-Taste zum Vervollständigen von Kommandos + Datei/Verz.namen nutzen!

* Versehentlichen Aufruf eines Kommandos ohne "sudo" mit "sudo" wiederholen
```
    sudo !!   # Bash: !! = vorheriges Kommando
```

* Im Editor "Vi(m)" vergessenes "sudo" nach Änderungen an Konfigurations-Datei
  "KKK" kompensieren (oder in irgend einem anderen Editor):
```
    sudo vi(m) /etc/KKK        # So wäre es korrekt gewesen!
    vi(m) /etc/KKK             # So wurde Editor aufgerufen ("sudo" vergessen)
    ...                        # Datei "KKK" editiert
    :w                         # "write" (geht nicht!)
    :w!                        # "write" erzwingen (geht auch nicht!)
    :w /tmp/KKK                # "write" Kopie in Verz. "/tmp" (geht immer!)
    :q                         # "quit" (geht nicht!)
    :q!                        # "quit" erzwingen (geht immer!)
    ls -l /etc/KKK             # Zugriffsrechte+Besitzer anzeigen
```

```
    sudo mv /tmp/KKK /etc/KKK  # Neue Version an alte Stelle verschieben (sudo!)
    ls -l /etc/KKK             # Zugriffsrechte+Besitzer kontrollieren
```

1) Einführung

* Aufgabe: Überwachen von Netzwerken und ihren Komponenten
  + Verfügbarkeit (aktueller Zustand --> keine Historie)
  + Performanz (Messwert-Historie --> über extra Tools)

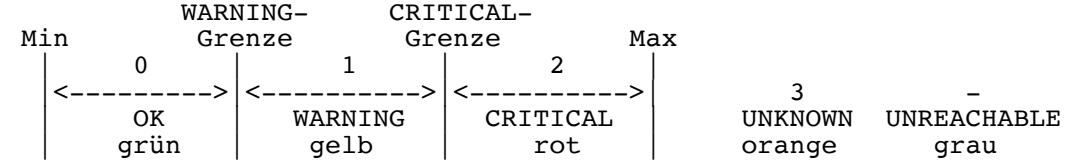
* Nagios = Network + (H)Agios (griechisch: "Heiliger")
  + Acronym: "Nagios Ain't Gonna Insist On Sainthood" ;-)
  + Autor: Ethan Galstad (Amerikaner)
  + 1999 begonnen (als "NetSaint") --> 15 Jahre Entwicklungszeit
  + 2003 umbenannt in "Nagios" wg. Namenskollision

* Beobachtet permanent beliebige Bestandteile eines Netzwerks und
  schlägt Alarm, sobald etwas "aus dem Ruder läuft"
  + Periodischer Aufruf von CHECK-Programm pro zu überwachender Komponente
    -> Liefert Exit-Code + Beschreibungstext
    -> In Status-Datenbank eingetragen + in GUI angezeigt (als Farbe)

* Stellt Netzwerk-Struktur mit logischen ABHÄNGIGKEITEN dar
  Stellt auf Messdaten-Basis AKTUELLEN ZUSTAND der Netzwerk-Komponenten dar

* STATUSORIENTIERT --> Liefert QUALITATIVE Aussagen gemäß AMPELSYSTEM
  + "Schwelle" nötig für Grenzwert-Definition: OK <-> WARNING <-> CRITICAL

Farbe	Status	Code	Messwert
Grün	OK	0	... < WARNING-Grenze
Gelb	WARNING	1	WARNING-Grenze <= ... < CRITICAL-Grenze
Rot	CRITICAL	2	... >= CRITICAL-Grenze
Orange	UNKNOWN	3	Unbekannt (Messwert fehlt)
Grau	UNREACHABLE	-	Unbekannt (wg. Ausfall eines "parents")



* Schwellwerte PRO CHECK flexibel einstellbar (per -w=warning, -c=critical):
  + check_smtp -w 5.0   -c 8.0   localhost   # 1 (Antwortzeit in Sek)
  + check_disk -w 10%   -c 5%   -p /var   # 1 (freier Plattenplatz)
  + check_procs -w 150   -c 250   --metrics=PROCS # 1 (Anz. Prozesse)
  + check_icmp -w 200.0,40%   -c 1000.0,80% localhost   # 2 (ms Laufzeit,Paketverlustrate)
  + check_load -w 2.0,1.0,0.5 -c 4.0,2.0,1.0 localhost   # 3 (1/5/15 Min Durchschn.)

* Liefert KEINE QUANTITATIVEN Aussagen (zeitlicher Verlauf der Messwerte)
  + Momentansicht (ändert sich ständig)
  + Live-Status = letzter Messwert
  + Kann allerdings solche Daten aufzeichnen (NDO/IDO = Nagios/Icinga Data Out)
    (z.B. NDOutils/IDOutils zeichnet alle Werte extern auf)
  + Auswertung mit Zusatztools (AddOns): NagVis, PNP4Nagios, NagiosGrapher, ...

* Fokus: Funktionalität und Stabilität, nicht Design (d.h. "schön" ist anders!)

* Solider Monitoring BAUKASTEN (nicht bloß GUI), d.h. notwendig sind:
  + Editieren von Konfigurations-Dateien   (Editor, z.B. Vim)
  + Installation von Zusatzsoftware   (Administrieren)
  + Schreiben/Ändern/Testen von Check-Plugins (Programmieren)

2) Zweck von "Monitoring"

* Permanenter Gesamtüberblick über Infrastruktur-Zustand

* Probleme/Fehler zeitnah erkennen
  + Frühzeitig (VOR den Anwendern ;-)
  + Auch wenn keiner vor dem Rechner sitzt
  + Nicht so offensichtliche
    - Rechenzentrums-Umzug
    - Rechner ersetzen
    - OS- oder SW-Updates
    - Zertifikat-Ablauf
    - Klimaanlage im Rechenzentrum ausgefallen --> 70 Grad!
    - Wassereinbruch
    - Kurzeitiger Stromausfall --> USV angesprungen
    - Einbruch/Diebstahl/Sabotage
    - Kabel unterbrochen/versehentlich gezogen
    - ...

* GEZIELTE Fehlersuche unterstützen
  (nicht an falscher Stelle suchen!)

* Unnötige Nachrichten vermeiden (99.9% OK-Meldungen),
  damit wichtige Ereignisse nicht im Wust von Meldungen untergehen!

* Routineaufgaben automatisieren

* DOKUMENTATION der System- und Netzwerkstruktur!
  (Infrastruktur ändert sich ständig -- wächst ;-)

* Dokumentation der VERFÜGBARKEIT gegenüber Kunden
  (Einhaltung von SLA = Service Level Agreements)

* Langfristige Trends ermitteln --> Kapazitätsplanung

* Monitoring (Nagios) ist nicht Graphing (MRTG, Cacti)!
  (Visualisierung von Messwerten über längere Zeiträume)
  Aufzeichnung der Messwerte ebenfalls notwendig

3) Was ist überwachbar? -- ALLES!

* Hardware
  + Netzwerk-Komponenten per SNMP (Simple Network Management Protocol)
  + Server per Herstellertools (HP OpenView, IBM Director, ServerView, OpenManage, ...)
  + Physikalische Messwerte (Stromverbrauch, Wärmeverbrauch, Helligkeit, ...)
  + Umwelt (Temperatur, Luftfeuchtigkeit, Niederschlag, Windstärke+richtung,
    PH-Gehalt, Durchflussmenge, ...)
  + Haushaltsgeräte
  + Maschinen

* Betriebssystem (viele fertige Plugins vorhanden!)
  + Ressourcen: CPU, Speicher, Dateisysteme, Platten, RAID/SMART
  + Prozesse und Dienste
  + Performanz
  + Logfiles (Analyse, Mustererkennung)

* Netzwerk-Dienste (viele fertige Plugins vorhanden!)
  + Alle gängigen per Simulation eines Client-Zugriffs
    (DNS, FTP, HTTP, LDAP, SMTP, SSH, SMB, POP3, IMAP)
  + TCP/UDP-Ports
  + ICMP

* Anwendungen
  + SAP
  + Datenbanken (MySQL, PostgreSQL, BDB, ORACLE, DB2, SQL-Server)
  + Messaging-Systeme (Mail-Server)
  + Web- und Applikations-Server (Apache, Tomcat)
  + Verz.-Dienste (AD, LDAP, NDS)
  + Telefonieserver
  + ...

* Weiteres
  + Eigene Skripte und Plugins
    (Shell, Perl, VBScript, Java, Python, PHP, C#/.NET, PowerShell, ...)

4) Was bietet Nagios?

* Überwachen
  + PRIVATE Rechner-Betriebsmittel: CPU, RAM, Disk, User, Prozesse, ...
    --> Agent auf Gerät notwendig: SSH, NRPE, NSCA, NSCP, NWSC, SNMP, ...
  + ÖFFENTLICHE Netzwerk-Dienste: SMTP, POP3, IMAP, HTTP, FTP, ...
  + Netzwerkfähige GERÄTE: Switch, Hub, Router, Drucker, VoIP, ...

* Alle Betriebssysteme überwachbar (Linux, UNIX, Windows, MacOS, Host, ...)
  (läuft aber selbst nur unter Linux/UNIX!)

* Streng MODULAR -- erweiterbar über PLUGINS
  --> Sehr viele ERWEITERUNGEN verfügbar

* Berücksichtigt ABHÄNGIGKEITEN (Dependencies) von Netzwerk-Komponenten
  --> Nachrichten-Unterdrückung abhängig vom Zustand anderer Hosts/Services

* Alarmierung/Benachrichtigung (Notification) bei Problem-Auftreten/Behebung
  --> Ampel, Sound/Gong/Sirene, Email, SMS, Anruf, Chat, ...

* Eskalationsmanagement (falls keine Reaktion erfolgt)
  --> Information anderer/weiterer Kontaktgruppen

* Ereignisbehandlung (Event-Handler) zur automatischen Problemlösung
  --> Kleinere Problem ohne Administrator-Eingriff beheben ("Selbstheilung")
      (angenehm + GEFÄHRLICH, da Problem evtl. nicht wahrgenommen!)

* Grafische Visualisierung des aktuellen Zustandes (per Webinterface)
  --> In jedem Browser anzeigbar

* Aufzeichnung des zeitlichen Verlaufs (Logdateien)
  --> Auswertung durch andere Tools

5) Vor- und Nachteile von Nagios

* Vorteile
  + Textbasierte Konfiguration ;-)
  + Sehr große Verbreitung
  + Flexibel und beliebig individualisierbar
  + Viel Erfahrung auch in großen Projekten
  + Stabil und ausgetestet (Grund: langsame Entwicklungsgeschwindigkeit)
  + Konstante und konservative Weiterentwicklung
  + Einfache Erweiterbarkeit durch Plugins
  + Viele AddOns (unter GPL)
  + Professioneller Support erhältlich (Netways, ConSol, Heinlein, Thomas-Krenn,
Mathias Kettner)
  + Große aktive deutsche Community
  + Einige Forks (Schnittstellen + Konfiguration gleich bzw. erweitert)
    - Shinken: Nachprogrammierung in Python (POC = Proof of Concept,
                 Scaling to large environments, Dez 2009)
    - Icinga(1): Am 6.5.2009 durch Firma Netways (Quellcode-Kopie in C)
                 --> 1. stabile Version 28.10.2009
    - Icinga 2: Okt 2012 begonnen (parallel zu Icinga 1, C++ statt C)
    - Naemon: Okt 2013, Andreas Ericsson, Firma Op5, Mitglied Nagios Core Team
    - Kommerzielle Ableger: NetEye, Groundwork Op5, OpenITCockpit, ...
  + Keine (Lizenz)Kosten
  + Open Source (GPLv2)
  + Umfangreiche, vollständige und gute Dokumentation (nur Englisch)

* Nachteile
  + Textbasierte Konfiguration 8-(
  + Schwer überschaubare Konfiguration
    - Komplexität hoch (Baukasten)
    - Syntax kompliziert
  + Grafische Darstellung "old-fashioned" --> Icinga 2, TKmon, ...
  + Performance niedrig bei vielen Checks --> Icinga 1/2, Shinken
  + Hardware/Software-Hersteller stellt kein Plugin zur Verfügung
    (z.B. nur teuren Java-Client)
    --> Suchen oder selbst schreiben --> Programmiererfahrung notwendig
  + Hardware/Software-Hersteller stellt auch keine API-Doku zur Verfügung
    --> Pech gehabt!
  + Frei verfügbare Plugins fehlerbehaftet/nicht allgemein genug (zu speziell)
    - Auswahl unter mehreren Möglichen dauert zu lange und endet unbefriedigend
  + Updates kritisch
    - Nagios-Konfiguration + Nagios-Kern + Plugins + AddOns + Datenbank +

nötige Programmiertools, ... gleichzeitig schwierig zu aktualisieren
--> parallel neue Version aufsetzen bis Gleichstand, dann umschalten

* Probleme
  + Ausführung der Checks/Plugins erzeugt:
    - Prozess-Forks
    - Double-Prozess-Forks
    - Memory Overhead
  + Checkresultate auf Filebasis erzeugt I/O Last
  + Event-Handler blockieren "Dispatcher" im Nagios-Core (NERD)
    - Steuert zeitl. Reihenfolge der Plugin-Aufrufe
    - Inzwischen gegen bessere austauschbar: Bronx, DNX

6) Anmerkungen zu Nagios

* Nagios ist ein "FRAMEWORK/BAUKASTEN" und kann erst mal kaum etwas
  Muss zu 100% gesagt bekommen, was es tun soll (nichts automatisch!)
  --> Q: "Nagios is Not for Dummies"
  --> Q: "Nagios ist ein sehr guter Schüler,
         braucht aber auch einen guten Lehrer!"

* Nagios kümmert sich NUR um
  + Plugin-Aufrufe (Check aufgrund Konfiguration)
  + Aufzeichnen + Auswerten der Plugin-Ergebnisse
  + Reaktionen darauf (Notification/Event-Handler wiederum über Plugins!)
  --> Q: "Made im Speck"
  --> Q: "Nagios ist nur ein Scheduler/Dispatcher"

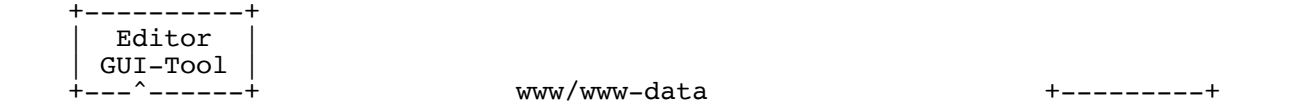
* Wenige fest eingebaute Funktionen (CORE) +
  genau festgelegte Schnittstellen (APIs)
  Rest machen Plugins (+ UNIX/Linux-Kommandos!)
  --> Streng modular + erweiterbar
  --> Skalierung durch Entkopplung der Komponenten
  --> Hoher Ressourcenverbrauch wg. Kommunikationsaufwand + viele Prozesse

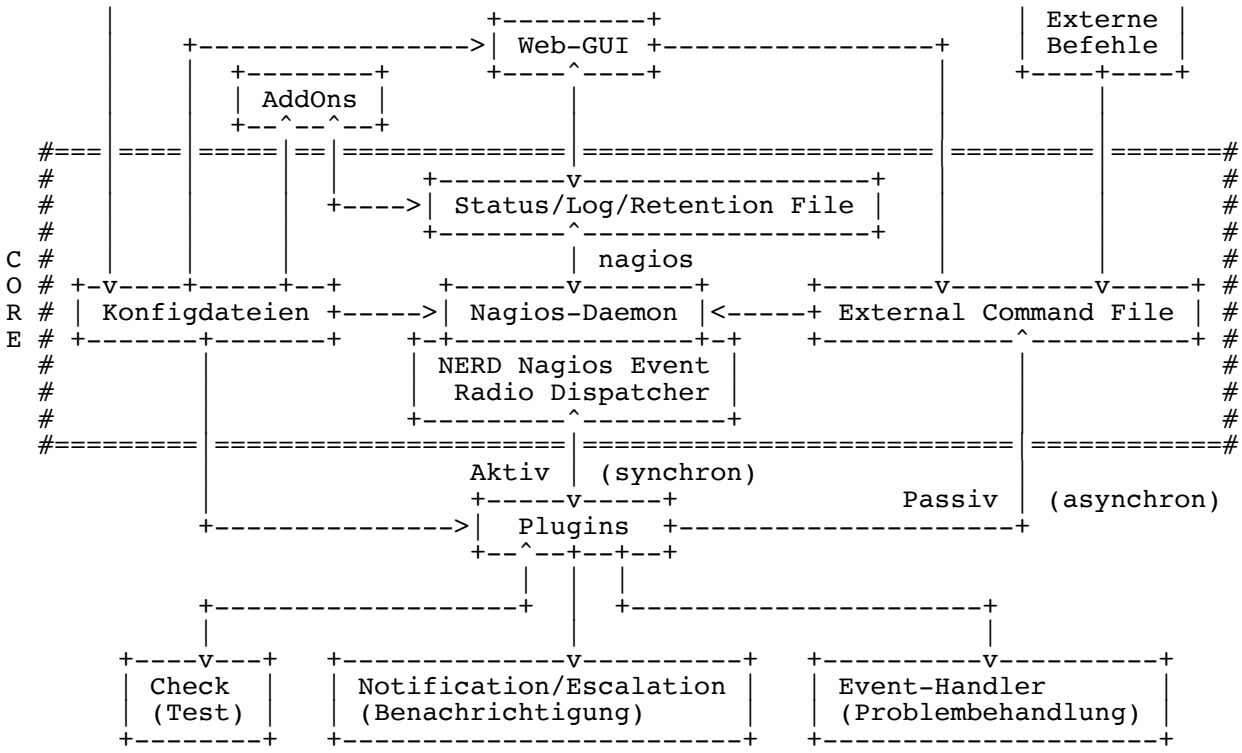
* Aussagen von vergleichbarer Aussagekraft
  + Q: "Nagios kann alles überwachen"
  + Q: "Produkt kann per XML Daten austauschen"
  + Q: "Produkt kann per SOAP Daten austauschen"
  + Q: "Produkt ist per SNMP auslesbar und steuerbar"
  + Q: "Produkt ist ein Transaktionsmonitor"
  + Q: "Produkt ist ein relationales Datenbanksystem"

7) Architektur von Nagios

* Saubere Trennung in
  + Nagios-Daemon (Core)
    - Zentrales Framework (API festgelegt)
    - Konfiguration (Dateien)
    - Status-Speicher (binär im RAM + auf Platte)
    - Command/Event-Queue (Warteschlange)
    - Scheduling (Planung und Anstoßen von Checks)
    - Notification (Benachrichtigen)
    - Event-Handling (Problembehandlung)
    - Logging (direkte Dateien + Syslog)
  + Nagios Plugins
    - Überwachung einzelner Hosts/Dienste (Checks)
    - Skripte oder Binärprogramme (Kommandos)
    - Zustandsmeldungen an Nagios-Daemon (OK, WARNING, CRITICAL, UNKNOWN)
    - Statusmeldungen zusätzlich (z.B. "0 kB (0%) free on /dev/sda1")

* Kommunikation in Nagios
  + Client-Server-Prinzip
  + Eigene Nagios-Protokolle (NRPE, NSCA, NSCP = NSClient++, NWSC)
  + Standard-Protokolle (SSH, SNMP, WMI)
  + Aktive Abfrage = Erst Server tätig, dann Plugin (synchron)
    Passive Meldungs-Entgegennahme = Erst Plugin tätig, dann Server (asynchron)





* Editor/GUI-Tool: Konfiguration von Nagios und Addons (Konfig.-Dateien)

* Web-GUI: Übersicht Systemstatus
  Reportgenerierung (URLs definieren für häufig nötige Reports)
  Trendanalyse
  Tests + Benachrichtigungen ein/ausschalten
  Wartungsfenster ein/ausschalten
  Kommentare eintragen/löschen ("in Bearbeitung")
  Rescheduling von Checks

* Nagios-Daemon: Führt periodisch (aktiv) Tests aus
  Nimmt (passiv) Testergebnisse entgegen
  Wertet Testergebnisse aus
  Löst Benachrichtigungen (Alarm) aus

* NERD (Nagios Event Radio Dispatcher): Scheduler für Checks, Notifications, Ecalations und Event-Handler (möglichst blockierungsfrei)

* Status File: Speichert aktuelle Zustände (für GUI, Status-Übergänge)
  Log File: Sammelt Plugin-Meldungen (Messwerte)
  Retention File: Speichert Status File periodisch (für Nagios-Restart)

* Konfigdateien: Beschreibung zu beobachtender Geräte/Dienste/Ressourcen
  + ihre Abhängigkeiten + Plugin-Aufrufe
  In Form von Dateien (auf Verz. verteilt)

* Plugins: Externe Kommandos (egal in welcher Programmiersprache!)
  Von Nagios für bestimmte Aktionen aufgerufen
  (Check, Notification/Escalation, Event-Handler)

* Check-Plugin: Durchführung Einzeltest (direkt/indirekt, aktiv/passiv)
  Rückmeldung an Nagios-Daemon (Status + Text)

* Notification-Pl: Versenden von Benachrichtigungen bzw.
  Escalation-Pl: Eskalation von Benachrichtigungen

* Event-Handler: Problembehandlung vor Benachrichtigung bzw.
  bei fehlender Reaktion ("Self-Healing")

* External Cmdfile: Liste von Befehlen an Nagios-Daemon
  (von GUI, Passive Checks, externen Kommandos, Benutzer)

* ADDONS: Erweiterungen (GUI-Konfiguration, Performance-Grafik, ...)
  Ebenfalls per Konfigurations-Dateien konfiguriert
  (RIESENGEBIET!)

8) Web-GUI von Nagios

* Aufruf (URL): http://HOST/nagios # z.B. localhost, V2
  http://HOST/nagios3 # V3
  http://HOST/nagios4 # V4
  http://USER:PASSWORD@HOST/nagios # mit Account+Passwort
  http://USER:PASSWORD@HOST/nagios3 # V3
  http://USER:PASSWORD@HOST/nagios4 # V4

* Einrichten Web-GUI Accounts + Passworte
  --> 8.1) Zugriffssteuerung auf Web-GUI

* Darstellung aktueller Zustand je Host und Service (Dienst) in GUI:

Host		Farbe		Service
UP	0	Grün	0	OK
-	-	Gelb	1	WARNING
DOWN	1	Rot	2	CRITICAL
UNREACHABLE	2	Orange	3	UNKNOWN
PENDING	-	Grau	-	PENDING

* Unterschiedliche Detail-Ebenen
  + Taktische Übersicht = Zusammenfassung (Tactical Overview)
  + Matrixansicht
  + Gruppenansicht (von Hosts oder Services, frei definierbar)
  + Detailansicht
  + Statusmap (Grafische Netzwerkübersicht mit definierbaren Bildern)
  + Topologie
  + Reports und Logdateien-Zugriff
    - Statusänderungen
    - Statustrends
    - Benachrichtigungen
    - Verfügbarkeit in % + Zeit
    - Performancewerte per Schnittstelle

* Steuerung des Nagios-Daemon
  + Checks ein/ausschalten
  + Benachrichtigungen ein/ausschalten
  + Störungsmeldungen quittieren (Acknowledegement = "in Bearbeitung")
  + Kommentare hinterlegen
  + SCHEDULING der Abfragen beeinflussen
  + Daemon hoch/runterfahren

* CGI-Skripte der GUI (/usr/lib/cgi-bin/nagios3/*)

CGI-Skript	Aufgabe
avail.cgi	Verfügbarkeitsstatistik
*cmd.cgi	Externe Kommandos durchführen (etwa 160 Stück)
*config.cgi	Objekt-Konfiguration anzeigen (readonly)
*extinfo.cgi	Erweiterte Info + Kmdo ausführen für H/HG/S/SG
histogram.cgi	Ereignis-Übersicht (Anzahl eingetretener Ereignisse)
history.cgi	Logeinträge nach Ereignissen filtern + anzeigen
notifications.cgi	Verschickte Nachrichten anzeigen
outages.cgi	Für Teilnetz-Ausfall verantwortliche NW-Knoten anz.
showlog.cgi	Logfile-Einträge anzeigen
*status.cgi	Zustände anzeigen (parametrisierbar, WICHTIG!)
statusmap.cgi	Topologische NW-Karte ("parents")
statuswml.cgi	Statusabfrage via Handy (WML = Wireless Access Prot.)
statuswrl.cgi	Topologische NW-Karte in 3D (VRML-Browser nötig)
*summary.cgi	Wunschauswertungen per Auswahlmaske (H/S/Fehler/TP)
tac.cgi	Tactical Overview (Startseite)
trends.cgi	Zustände über Zeitachse anzeigen
graphs.cgi	
rrd2-graph.cgi	
rrd2-system.cgi	
mrtg-rrd.cgi	

```
  +------------------+-----------------------------------------------------+

* TIPP: CGI-Skript per Kommandozeile ausführen
        (root-Recht notwendig!, vermeidet wget, curl):

    REQUEST_METHOD="GET" \                  # Art der HTTP-Abfrage
    QUERY_STRING="." \                      # Query (TODO)
    REMOTE_USER="nagiosadmin" \             # Anmeldung
      /usr/lib/cgi-bin/nagios3/status.cgi   # Abfrage-Skript
        tidy |                              # HTML normieren (verzichtbar)
        w3m \                               # HTML in Text konvertieren
            -T "text/html" \                #   Content-Type
            -I "UTF-8" \                    #   Input-Zeichensatz
            -O "latin1" \                   #   Output-Zeichensatz
            -dump |                         #   Auf STDOUT ausgeben
        grep -wi "CRITICAL" |               # Ergebnis filtern (w=word,
        grep -wi "load" |                   # Ergebnis filtern  i=ignorecase)
        less                                # Seitenweise anzeigen

* Parameter der CGI-Skripte:

  +-------------------------------------------------------------------------+
  | status.cgi                                                              |
  +-------------------------+-----------------------------------------------+
  | host=H                  | H: all, Host                                  |
  | hostgroup=HG            | HG: all, Hostgroup                            |
  | servicegroup=SG         | SG: all, Servicegroup                         |
  | style=TYP               | TYP: overview, summary, grid, detail, hostdetail
  | hoststatustypes=NUM     | NUM: 1 PENDING, 2 UP, 4 DOWN, 8 UNREACHABLE   |
  | servicestatustypes=NUM  | 1 PENDING, 2 OK, 4 WARNING, 8 UNKNOWN, 16 CRITICAL
  | hostprops=NUM           | NUM: 1/2 Scheduled Downtime/No ...            |
  | serviceprops=NUM        | NUM: 4/8 Acknowledgement/No ...               |
  |                         | 16/32 Host/Service-Check deakt./aktiviert     |
  |                         | 64/128 Eventhandler deakt./aktiviert          |
  |                         | 256/512 Flap Detection deakt./aktiviert       |
  |                         | 1024/2048 Host/Service oszilliert/nein...     |
  |                         | 4096 Host/Service von Benachrichtigung ausgeschl.
  |                         | 8192 Benachrichtigung aktiviert               |
  |                         | 16384/32768 Passive Host/Service Check deakt./akt.
  |                         | 65536 Host/Service mind. 1 Result aus pass. Test
  |                         | 131072 Host/Service mind. 1 Result aus akt. Check
  |                         | 252144 Host/Service in Hard State (V3)        |
  |                         | 524288 Host/Service in Soft State (V3)        |
  +-------------------------+-----------------------------------------------+
  | extinfo.cgi                                                             |
  +-------------------------+-----------------------------------------------+
  | type=0                  | Globale Parameter anzeigen                    |
  | type=1&host=H           | Kommandos und Infos zu Host H                 |
  | type=2&service=S        | Kommandos und Infos zu Service S              |
  | type=3                  | Vorhandene Host/Service-Kommentare anzeigen   |
  | type=4                  | Nagios-Perform.-Info (Host/Service/aktiv/passiv)
  | type=5&hostgroup=HG     | Kommandozentrale für Hostgruppe HG            |
  | type=6                  | Wartungszeiträume für Hosts/Service anzeigen  |
  | type=7                  | Geplante Tasks sortiert nach Ausführungszeitpunkt
  | type=8&servicegroup=SG  | Kommandozentrale für Servicegruppe SG         |
  +-------------------------+-----------------------------------------------+
  | cmd.cgi                                                                 |
  +-------------------------+-----------------------------------------------+
  | cmdtype=NUM             | Kommando ausführen (NUM = 1..156)             |
  | host=H                  | Host H auswählen                              |
  | hostgroup=HG            | Hostgruppe HG auswählen                       |
  | service=S               | Service S auswählen                           |
  | servicegroup=SG         | Servicegruppe SG auswählen                    |
  +-------------------------+-----------------------------------------------+
  | statusmap.cgi                                                           |
  +-------------------------+-----------------------------------------------+
  | cmdtype=NUM             | Kommando ausführen (NUM = 1..122)             |
  +-------------------------+-----------------------------------------------+
  | histogram.cgi                                                           |
  +-------------------------+-----------------------------------------------+
  | history.cgi                                                             |
  +-------------------------+-----------------------------------------------+
  | host=H                  | Host H auswählen                              |
  | hostgroup=HG            | Hostgruppe HG auswählen                       |
  | service=S               | Service S auswählen                           |
  | servicegroup=SG         | Servicegruppe SG auswählen                    |
```

```
  +-------------------------+-----------------------------------------------------+
  | notification.cgi                                                              |
  +-------------------------+-----------------------------------------------------+
  | host=H                  | Host H auswählen                                    |
  | service=S               | Service S auswählen                                 |
  | contact=C               | Kontakt C auswählen                                 |
  +-------------------------+-----------------------------------------------------+
  | summary.cgi                                                                   |
  +-------------------------------------------------------------------------------+

* Beispiel für CGI-Webabfragen (Syntax):
    http://localhost/nagios/cgi-bin/status.cgi?FILTER
  FILTER z.B.:
    host=all&type=detail&hoststatustypes=3&serviceprops=10&servicestatustypes=28
    ...&sorttype=1&sortoption=6

* Navigationsbereich (linke schwarze Seitenleiste)
    /usr/local/nagios/share/side.html

* In C geschrieben (Sic!)
  + Schnell
  + Schlecht wartbar/erweiterbar


8.1) Zugriffssteuerung auf Web-GUI
----------------------------------

* Anmeldung
  + Zunächst NUR per HTTP von "localhost" erlaubt
    - Erweiterbar auf beliebige Rechner (-->
    - Verschlüsselter Zugriff per HTTPS möglich (SSL-Zertifikat)
  + Zunächst nur "Basic Authentication" (per "htaccess.passwd"/"htaccess.users")
    - Standard-Account "nagiosadmin" darf alles
  + Beeinflusst
    - Sichtbarkeit und Bedienbarkeit von Hosts/Services
      (vollständig, Read-Only, nur wenn im Contact)
    - Erlaubte Tätigkeiten (Lesen, Ändern)

* Zugriff auf CGI-Interface (Web-GUI):
    http://localhost/nagios    # Lokal  Name V2
    http://127.0.0.1/nagios3   # Lokal  IP   V3
    http://HOSTNAME/nagios     # Remote Name V2
    http://IP/nagios3          # Remote IP   V3
  Anmeldung mit Account-Name + Passwort erforderlich:
    nagiosadmin + PASSWORD     # Standard-Name mit allen Zugriffsrechten
    nagiosadm   + PASSWORD     # Evtl.
    nagios      + PASSWORD     # Evtl.

* Für Standard-Account "nagiosadmin" sind in "cgi.cfg" ALLE Zugriffe erlaubt
  (weitere solche Benutzer per "," getrennt angeben):
    authorized_for_system_information        = nagiosadmin   # Systeminfo ansehen
    authorized_for_configuration_information = nagiosadmin   # Konfig. ansehen
    authorized_for_system_commands           = nagiosadmin   # Kmdos starten
    authorized_for_all_services              = nagiosadmin   # Services ansehen
    authorized_for_all_hosts                 = nagiosadmin   # Hosts ansehen
    authorized_for_all_service_commands      = nagiosadmin   # Service-Kmdo st.
    authorized_for_all_host_commands         = nagiosadmin   # Host-Kmdo starten

* Accounts die nur LESEZUGRIFF haben, aber nichts ändern können sollen,
  als "Read Only" in "cgi.cfg" eintragen:
    authorized_for_read_only                 = user1, user2, ...

* ALLE ANDEREN im Apache in der Datei "htaccess" eingerichteten Accounts sehen
  NUR die Host/Service-Objekte, bei denen ihr Account-Name unter "contacts" bzw.
  "contact_groups" in Host/Hostgruppe/Server/Servergruppe eingetragen ist!
    contacts         user1, ...
    contact_groups   group1, ...

* Zugriffssteuerung: Apache Benutzer-Anmeldung + -Passwort einrichten
  (ohne Option "-b" wird Passwort interaktiv abgefragt --> bessere Methode):
    /etc/apache2/conf.d/nagios3.conf
      # --> /etc/nagios3/apache2.conf      # Symbol. Link
      # --> /etc/nagios3/htpasswd.users    # Account+Passwort-Datei
  + Htaccess-Datei NEU erzeugen (-c=Datei erz., -b=Passwort auf Kmdozeile):
      KMDO      OPTS HTACCESS-DATEI              USER        PASSWORD
      htpasswd -c -b /etc/nagios3/htpasswd.users nagiosadmin geheim
```

```
      chmod o+r /etc/nagios3/htpasswd.users  # Leserecht für andere setzen!
  + Htaccess-Datei Eintrag hinzufügen (-b=password on cmdline):
      KMDO     OPT HTACCESS-DATEI              USER    PASSWORD
      htpasswd -b  /etc/nagios3/htpasswd.users visitor geheim


9) Alarmierung (Benachrichtigungen)
-----------------------------------

* Flexibel per Skripte definierbar
  + E-Mail
  + SMS, Pager
  + Instant Messenger (ICQ, Jabber, ...)
  + Windows-Message (NET SEND Popup)
  + VoIP-Anruf (Asterisk)
  + Echte "Ampel" am Serverschrank ;-)

* Umfangreich konfigurierbar
  + Personen + Gruppen
  + Zeitfenster + Intervalle
  + Eskalation bei fehlender Reaktion
    --> Kette definierbar

* Automatische Unterdrückung bei Folgefehlern und Abhängigkeiten

* Schnittstelle zu anderen Anwendungen


10) Weitere Eigenschaften von Nagios
------------------------------------

* Distributed Monitoring
  + Mehrere Nagios-Server überwachen Teilbereiche
  + Weitermeldung ihrer Ergebnisse an zentralen Server
  + Einsatzbereich
    - Logische Netzstruktur
    - Lastverteilung
    - Geschützte Bereiche (DMZ, Intranet)
    - End2End-Monitoring

* High-Availability (HA) relativ einfach

* Automatische Maßnahmen per "Event-Handler" bei Zustandswechsel OK <-> Fehler
  --> Self-Healing

* Erfassung von Downtimes (Ausfallzeiten, Wartungsfenster)

* Generierung von Performancewerten

* Vermeidung von Fehlalarmen durch Re-Checks & Flap-Detection
  + "Status Flapping" = schnell wechselnde Zustände (oszillierend)

* Unterscheidung Ausfall <-> Nicht-Erreichbarkeit aufgrund Abhängigkeiten
  --> Automatisches Unterdrücken unnötiger Checks

* Läuft unter Linux-Benutzer "nagios" und Linux-Gruppe "nagios" (nicht root)
  + Evtl. auch Gruppe "nagioscmd", "ncmd", "nagcmd" (Nagios Command Group)


11) Was ist notwendig um Nagios laufen zu lassen?
-------------------------------------------------

* Mindestens
  + UNIX/Linux-Rechner -- Windows leider nicht ;-(
  + Binärcode von Nagios-Daemon, Nagios-Plugins und Nagios-GUI (C)
  + Quellcode von ... + C-Compiler + Bibliotheken (nur zum selbst übersetzen)
  + Netzwerkkarte + konfiguriertes TCP/IP-Netzwerk

* Für GUI
  + Apache2 Webserver
  + gd-library

* Für Plugin-Ausführung
  + Shell (!)
  + Perl (!)
  + Python
```

  + C-Compiler (Binär = Standalone)
  + PHP
  + nmap
  + snmpget
  + ping
  + smartctl
  + ...

* Für Benachrichtigung
  + Mailclient (mail/mailx)
  + Mailserver Postfix/Exim/Sendmail
  + SMS-Gateway
  + Instant Messenger Gateway
  + VoIP-Telefonanlage
  + Echte "Ampel" am Serverschrank ;-)

* Für einige AddOns
  + Quellcode AddOns
  + Datenbank MySQL (evtl. auch sqlite, BDB, PostgreSQL, ORACLE, RRD-DB)
  + Weitere Bibliotheken

* Lizenz ;-)
  + GNU General Public Licence V2 (GPLv2, analog Linux)
  + Open Source (sehr wichtig!)

* Zeit + Geld 8-(
  + Hardware
  + Einarbeitung
  + Konfiguration
  + Dienstleistung
  + Beobachten + Problem-Bearbeitung (!)
  + Ständige Pflege der Konfiguration (!)
  + Updates

12) Wie als Anfänger beginnen?

* UNIX/Linux-Wissen notwendig!
  + Grundlagen Kommandozeile (Shell)
  + Systemadministration auf Kommandozeile
  + Editor auf Kmdozeile (vim, emacs, mc, gedit, ...)
  + Shell-Programmierung für Plugin-Anpassung/eigene Plugins
  + Perl-Programmierung für Plugin-Anpassung/eigene Plugins

* "Geduld ist eine Tugend"
  + Nagios ist mächtig und flexibel
  + Konfiguration wie gewünscht dauert wegen
    - Erlernen von Nagios-Begriffen / -Verhalten / -Optionen
    - Eigenes Netzwerk verstehen + beschreiben
    - Konfiguration designen + erstellen + testen
    - Plugins suchen/installieren/testen/einbinden (fremde)
      bzw. entwickeln/testen/einbinden (eigene)

* Sinnvolle Vorgehensweise
  + Installation aus Quellen (immer gleich!)
  + Zunächst nur Nagios-Maschine lokal monitoren
  + Netzwerk-Dokumentation schrittweise erarbeiten
  + Überwachung schrittweise erweitern
    - Rechner für Rechner mit seinen Diensten dazunehmen
      (Art der Kommunikation auswählen)
    - Abhängigkeiten zw. Rechnern und Diensten dazunehmen
    - Benachrichtigungen dazunehmen
    - Event-Handler dazunehmen
    - Neue Fehler --> Neue Überwachung zur Fehlererkennung dazunehmen

* Benachrichtigungen auf WIRKLICH relevante "eindampfen"
  Q: "40.000 Benachrichtigungen in Mailbox/Ordner gesammelt und nie angesehen"

* Saubere Strukturierung gerade für größere Umgebungen extrem wichtig
  + Aufteilung: Dateinamen + Verz.namen gut wählen
  + Hilfsmittel: Templates, Makros, Host/Service/Kontaktgruppen, ...
  + Kriterien:
    - Abstrakt: Host, Service, Contact, Timeperiod, ...
    - Konkret: Server, Standort, Projekt, ...

* "Nagios Core Documentation" lesen --> 12.1) Nagios Core Dokumentation

12.1) Nagios Dokumentation (Core Documentation)
---

* Links
  --> http://nagios.sourceforge.net/docs/nagioscore/3/en/toc.html Nagios V3 Core Doku
  --> http://nagios.sourceforge.net/docs/nagioscore/4/en/toc.html Nagios V4 Core Doku
  --> http://nagios.sourceforge.net/docs/3_0/toc.html   Nagios V3 Core Doku

* Inhaltsverzeichnis
  + toc.html   # Table of Contents

* Advice for Beginners
  + beginners.html   # Advice for Beginners

* Release Notes
  + about.html   # About Nagios Core (What is, Requirements, Licence)
  + whatsnew.html   # What's New in Nagios Core 3.x/4.x
  + knownissues.html   # Known Issues (bekannte Probleme)

* Quickstart Installation Guide
  + quickstart.html   # Quickstart
  + quickstart-fedora.html   # Fedora
  + quickstart-opensuse.html   # OpenSuSE
  + quickstart-ubuntu.html   # Ubuntu/Debian
  + upgrading.html   # Upgrading Nagios

* How to monitor a ...
  + monitoring-windows.html   # Windows machine
  + monitoring-linux.html   # Linux/Unix machine
  + monitoring-netware.html   # Netware server
  + monitoring-publicservices.html # Publicly available service (HTTP, FTP, SSH, etc)
  + monitoring-routers.html   # Router/Switch
  + monitoring-printers.html   # Network Printer

* Configuration Overview + Main Configuration File Options
  + config.html   # Configuration Overview
  + configmain.html   # Main Configuration File Options

* Object Configurations Overview
  + configobject.html   # Object Configuration Overview
  + objectdefinitions.html   # Object Definitions
  + objectinheritance.html   # Object Inheritance
  + objecttricks.html   # Time-Saving Tricks for Object Definitions
  + customobjectvars.html   # Custom Object Variables (ALT)

* Verifying Your Configuration + Starting and Stopping Nagios
  + verifyconfig.html   # Verifying your Configuration
  + startstop.html   # Starting and Stopping Nagios

* Understanding Macros and How They Work + Standard Macros in Nagios
  + macros.html   # Understanding Macros and How They Work
  + macrolist.html   # List of available Macros

* Nagios Plugins
  + plugins.html   # Nagios Plugins
  + pluginapi.html   # Nagios Plugin API
  + epnplugins.html   # Developing Plugins For Use With Embedded Perl
(ALT)
  + embeddedperl.html   # Using The Embedded Perl Interpreter
  + perfdata.html   # Performance Data

* Nagios AddOns
  + addons.html   # Nagios AddOns (NRPE, NSCA, NSCP, NWSC,
NDOUtils, ...)

* Security Considerations + Enhanced CGI Security and Authentication
  + security.html   # Security Considerations
  + cgisecurity.html   # Enhanced CGI Security and Authentication
  + int-tcpwrappers.html   # TCP Wrapper Integration
  + int-snmptrap.html   # SNMP Trap Integration

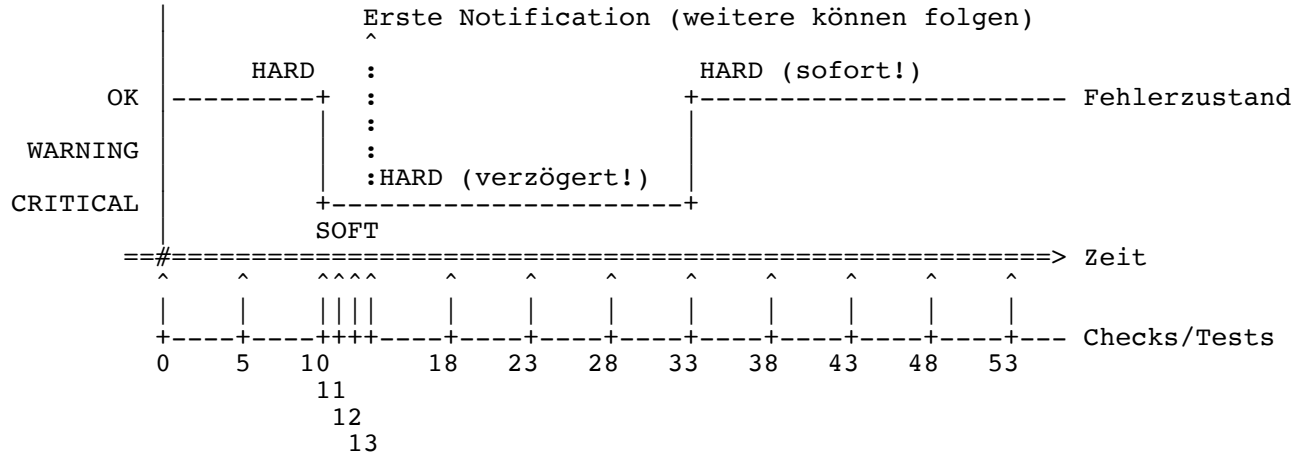
* CGI Configuration
  + configcgi.html   # CGI Config File

```
  + cgiauth.html                   # CGI-Authentication
  + cgis.html                      # Information On The CGIs
  + cgiincludes.html               # Custom CGI Headers and Footers

* Sonstige Teile der Nagios Dokumentation
  + activechecks.html              # Active Checks
  + passivechecks.html             # Passive Checks
  + servicechecks.html             # Service Checks
  + hostchecks.html                # Host Checks
  + checkscheduling.html           # Check Scheduling
  + dependencychecks.html          # Predictive Dependency Checks
  + cachedchecks.html              # Cached Checks
  + dependencies.html              # Dependencies
  + freshness.html                 # Service and Host Freshness Checks

  + statetypes.html                # State Types
  + eventhandlers.html             # Event-Handlers
  + notifications.html             # Notifications
  + escalations.html               # Escalations
  + timeperiods.html               # Time Periods (or... "Is This a Good Time?")
  + oncallrotation.html            # On-call notification rotations
  + flapping.html                  # Detection and Handling of State Flapping
  + stalking.html                  # State Stalking
  + volatileservices.html          # Volatile Services

  + adaptive.html                  # Adaptive Monitoring
  + extcommands.html               # External Commands

  + networkreachability.html       # Determining Status and Reachability of Network
Hosts
  + passivestatetranslation.html   # Passive Host State Translation
  + distributed.html               # Distributed Monitoring
  + redundancy.html                # Redundant and Failover Network Monitoring

  + clusters.html                  # Monitor Service and Host Clusters
  + downtime.html                  # Scheduled Downtime

  + tuning.html                    # Tuning Nagios For Maximum Performance
  + largeinstalltweaks.html        # Large Installation Tweaks
  + faststartup.html               # Fast Startup Options
  + mrtggraphs.html                # Graphing Performance Info With MRTG
  + nagiostats.html                # Using The Nagiostats Utility   !!!

  + integration.html               # Integration Overview

* "System Requirements" in "About Nagios" lesen
  + http://nagios.sourceforge.net/docs/nagioscore/3/en/about.html#requirements

* Nagios Core 3.x/4.x Version History
  + http://www.nagios.org/projects/nagioscore/history/core-3x
  + http://www.nagios.org/projects/nagioscore/history/core-4x


13) Zum Verständnis von Nagios wichtige Punkte
----------------------------------------------

13.1) Check-Typen
-----------------

  +----------+-------------------------------------------------------------------+
  | Typ      | Beschreibung                                                      |
  +----------+-------------------------------------------------------------------+
  | Active   | Plugin vom Nagios-Daemon ausgelöst.                               |
  |          | In regelmäßigen Abständen SYNCHRON ausgeführt (einstellbar,       |
  |          | d.h. Nagios wartet auf Antwort --> Evtl. Timeout)                 |
  |          | Plugin wird aufgerufen und Rückgabewert ausgewertet (Timeout)     |
  | Passive  | Durch externes Programm selbständig (periodisch) durchgeführt.    |
  |          | Liefert Status ASYNCHRON an Nagios-Daemon (z.B. SNMPTrap).        |
  |          | Ergebnis landet im "External Command File"                        |
  +----------+-------------------------------------------------------------------+
  | Service  | Regelmäßig Verbindungsversuch --> Service antwortet (korrekt?)    |
  |          |                                   --> Impliziert Host-Check!      |
  | Host     | Nur bei Bedarf --> d.h. alle Service-Checks schlagen fehl!        |
  |          |                    Meist per Ping (ICMP) --> Host antwortet       |
  +----------+-------------------------------------------------------------------+
  | Local    | Plugin startet auf Nagios-Server (Direct)                         |
```

```
  | Agent    | Plugin startet weiteres Plugin (Helper, Vermittler, Bindeglied) |
  | Remote   | Plugin startet auf überwachter/vermittelnder Maschine (Indir.)  |
  +----------+-----------------------------------------------------------------+
```

13.2) Status-Typ SOFT/HARD

* Alarmstatus-Typ (um falschen Alarm zu vermeiden)
  + SOFT (Übergang von OK --> WARNING/CRITICAL/...)
    Bemerkt, aber noch nicht sicher (1..N-1 Mal fehlgeschlagen)
    - JA: In Logdatei aufgezeichnet
    - JA: In Weboberfläche angezeigt
    * NEIN: Benachrichtigung (vor Anwender "verborgen")
    - EVTL: Event-Handler ausführen
    - CONFIG: check_interval   = 5   # 5 x interval_length = 5 x 60s = 300s
  + HARD = Sicher genug erkannt (mind. N Mal fehlgeschlagen, typisch: N=4)
    - JA: In Logdatei aufgezeichnet
    - JA: In Weboberfläche angezeigt
    * JA: Benachrichtigung ausführen
    - EVTL: Event-Handler ausführen
    - Typische max. Zeit bis Benachrichtigung: 9 Min = (5 Min + 4 x 1 Min)
    - CONFIG: retry_interval   = 1   # 1 x interval_length = 60s
              max_check_attempts = 10   # 4 x retry_interval = 4 x 60s = 240s



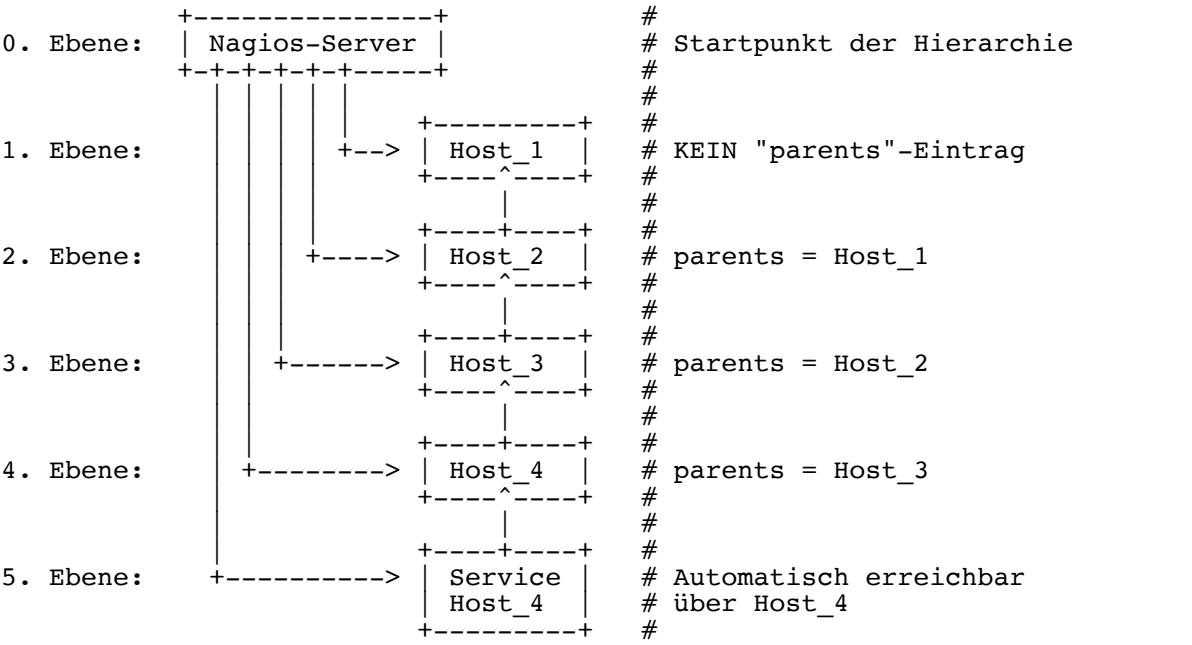
* SOFT: Status-Typ nach Status-Wechsel "OK" --> ...
  HARD: Def. Anzahl (schneller) Checks ergeben "nicht OK"

* Status-Wechsel "HARD ..." --> "SOFT OK" gibt es nicht!
  Übergang WARNING/CRITICAL/... --> OK wird sofort als HARD akzeptiert
  (wenn Host/Gerät/Dienst wieder funktioniert, dann sofort "OK")

13.3) Host-Abhängigkeit ("parents")

* Unterscheidung
  + Service-Check: Regelmäßig ausgeführt --> Impliziter Host-Check!
  + Host-Check: ALLE Service-Checks fehlgeschlagen --> nur bei Bedarf!
  + Host nicht erreichbar
    --> Parents des Host versuchen zu erreichen
    --> Parents der Parents versuchen zu erreichen
    --> ...

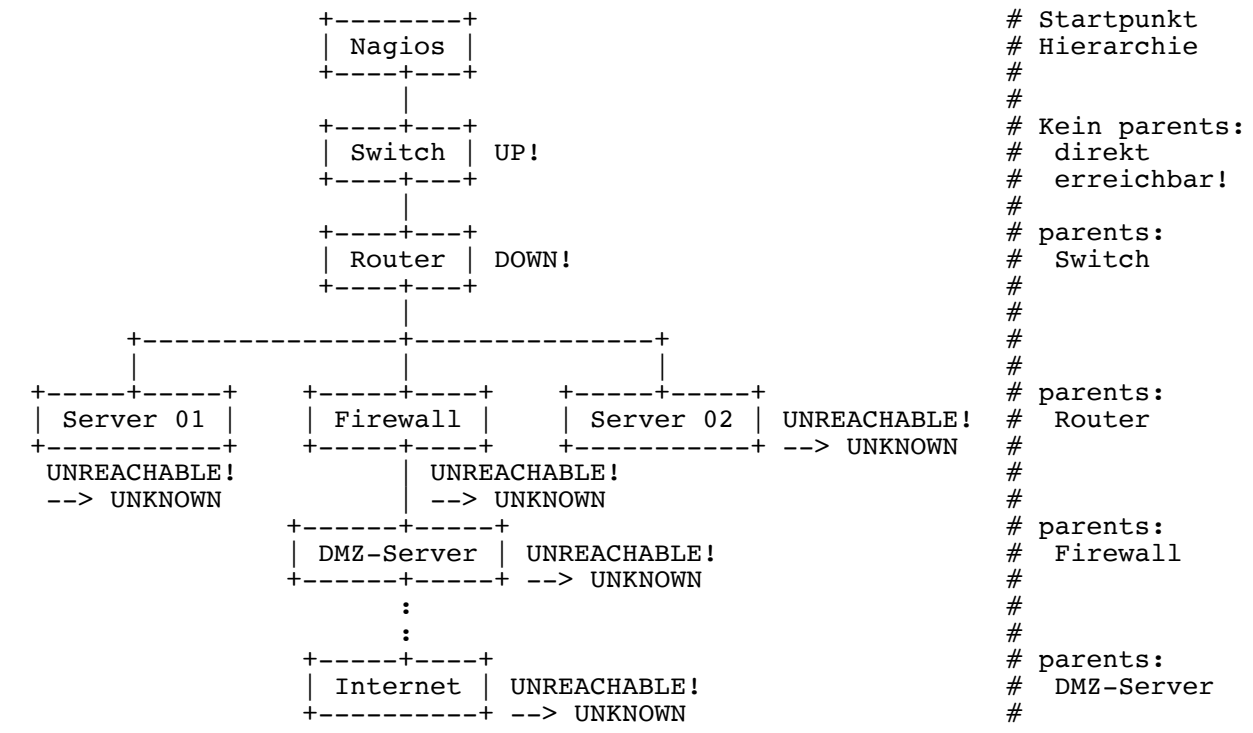
* Topologische Abhängigkeit von Host/Services per "parents" definierbar
  --> Eltern-Kind-Beziehung
  --> Gesamte (physikalische) Netzwerk-"Topologie" daraus abgeleitet
  + Definiert PHYSIKALISCHEN Weg von Hosts zu Nagios-Host
    --> Spiegelt sich in automatisch erzeugter Grafik in GUI wider!
  + Kein MUSS, Standard ist KEINE parents-Einträge:
    --> Alle Hosts/Services direkt Nagios-Server untergeordnet
    --> Nur eine Ebene in GUI-Grafik (ein "Haufen" um Nagios-Server herum)
  + Definition
    - Nagios-Server = "Startpunkt" der Hierarchie (oberste Ebene)
    - Hosts OHNE "parents"-Eintrag = "Kinder" des "Nagios-Servers" (1. Ebene)
    - Hosts MIT "parents"-Eintrag = Zeigen auf ihr Eltern-Objekt
    - MEHRERE "parents" pro Objekt erlaubt (V3)
    - Zyklus NICHT erlaubt!



* Aus "parents"-Angaben abgeleitet:
  + Netzwerk-Topologie (physikalische Erreichbarkeit)
  + Grafische Form der Status Map
  + UNREACHABLE-Meldungen (grau)

* Abhängigkeiten deaktivieren Checks bei Nichterreichbarkeit
  + Nur DIREKTE Host-Abhängigkeiten einzutragen --> Gesamthierarchie abgeleitet

* Schlussfolgerung "UNREACHABLE" aus Netzwerk-Topologie (physikalische Struktur)
  + Definiert mittels Child-Parents-Abhängigkeiten
  + UNREACHABLE-Server nicht mehr geprüft + keine Fehler mehr für sie gemeldet
    --> Status "UNKNOWN"



14) Konfiguration von Nagios
---

* Abstrakte OBJEKTE bilden (zu überwachende) Realität ab
  + Jedes bekommt frei wählbaren INTERNEN Namen --> BEZEICHNER (analog Prog.)
    --> BAUKASTEN basierend auf BEZEICHNERN
  + Muss nicht mit EXTERNEM Namen übereinstimmen (z.B. Benutzer, Gruppe, Host)
  + 12/14 Objekttypen möglich (z.B. Host, Service, Contact, ...)

  + Jedes Objekt hat bestimmte ATTRIBUTE, die seine Eigenschaften beschreiben
    (Attribute haben FESTE Namen, z.B. "host_name", "service_description")
  Q: "In Nagios ist alles ein Objekt" (analog "In Linux ist alles eine Datei")
  Q: "In Nagios hat alles einen Namen" (analog "Jede Datei hat einen Namen")
  --> DSL = Domain Specific Language
      Einheitliche SCHNITTSTELLE zw. Nagios, GUI, Datenbank, AddOns, ...

* Per EDITOR in Form von TEXTDATEIEN konfiguriert (ursprünglich keine GUI!)
  (z.B. aus bereits vorhandenen Inventarlisten per Skript generieren ;-)
  + NAMEN der Textdateien sind irrelevant
    (spiegeln aber oft Internen Namen oder Typ von Objekten wider)
  + Einlese-Reihenfolge und Reihenfolge der Objekt-Definitionen irrelevant
  + Entscheidend ist, dass nach dem Einlesen aller Konfigurationsdateien
    ALLE verwendeten Namen aufgelöst werden können (d.h. definiert sind)

* Konfiguration wg. Vielzahl an Einstellungsmöglichkeiten oft kritisiert
  + Objekte haben sehr viele Attribute
  + Zusammenhänge zwischen Objekten komplex (implizit + explizit)
  + Analoges Problem bei:
    - Samba mit "smb.conf" (SWAT = Samba Web Admin Toolkit)
    - Apache mit "httpd.conf"
    - ...

* GRÜNDLICHE PLANUNG der Konfiguration passend zur eigenen Struktur wichtig
  + "Mal schnell loslegen" führt ins Chaos!
  + Besser:
    - AUFTEILUNG auf Konfigurations-Dateien gut überlegen
    - GUTE NAMEN für alles vergeben (sprechend/intuitiv)
    - GRUPPEN von Hosts/Services bilden und gemeinsam behandeln
    - TEMPLATES für gleichartige Geräte/Dienste einsetzen + vererben
    - MAKROS für oft verwendete Werte verwenden --> änderungsfreundlich!
    - Check-Kommandos generisch definieren
    - ...

* VORHER nachdenken über
  + Netzwerkstruktur?   # Bandbreite, Firewall, Kundenanbindung, ...
  + Was überwachen?   # Server, Router, Switches, Measures, ...
  + Wie detailliert überwachen?   # Erreichbarkeit, OS-Daten, Spez. Aufgaben, ...
  + Worüber kommunizieren   # SSH, NRPE, NSCA, NSCP, SNMP, ...
  + Womit überwachen?   # Plugins suchen, auswählen, programmieren ...
  + Wann überwachen?   # Servicezeiten, Service Level Agreements, ...
  + Wen wann wie benachrichtigen? # Contact, Contactgroup, Escalation, ...
  + Wo überwachen?   # lokal, agent, remote, verteilt, ...
  + Prioritäten?   # Services, Dependencies, Escalations, ...
  + Sonderfälle und Ausnahmen   # ...

* Schlanke Konfiguration sinnvoll --> Weniger Text --> Templates + Gruppen!
  + Erleichtert Übersicht
  + Spart Zeit und Arbeit
  + Weniger fehleranfällig
    --> Konfiguration entschlacken -- Ballast auslagern!

* Stellt gleichzeitig System- und Netzwerk-DOKUMENTATION dar!
  --> Rechner neu aufsetzen!
  --> Rechenzentrumsumzug!

* Konfiguration von Nagios hört NIE auf!
  --> Man kann nicht alles voraussehen!
  --> Genug Zeit/Geld zum Planen und Sammeln der Daten nehmen!

* Makros einsetzen

Bedeutung	Syntax	Einsatzort
Installationsspezifisch (resource.cfg)	$USER1$ .. $USER32$	Pfad, Account Passwort, ...
Standard (Host/Service/Contact/...)	$HOSTNAME$, $HOSTADDRESS$, $SERVICEDESC$, $SERVICESTATE$	Aus Objekt automatisch
Argument/Parameter (command argument)	$ARG1$ .. $ARG32$	Datenübergabe Cmd --> Plugin
Benutzerdefiniert (custom variable macro)	$_(HOST\|SERVICE\|CONTACT)...$	Globale Variab. (ab V3!)

```
* Basis-Makros aus "resource.cfg" (nur für Nagios lesbar)
    $USER1$ IMMER Pfad zu Nagios-Plugin-Verz. --> verwenden!
    $USER2$ oft Pfad zu Event-Handler-Verz.   --> verwenden!
    $USER3$ oft Benutzername                  --> verwenden!
    $USER4$ oft Passwort                      --> verwenden!

* Parameterübergabe von internem auf externes Kommando
  + Durch "!" getrennt (Leerraum zählt)
  (Reihenfolge --> Argument-Numerierung)

   Internes Kommando                  Externes Kommando
   CHECK...!ARG1!ARG2!ARG3!...   -->  CMD... $ARG1$ ... $ARG2$ ... $ARG3$
            |    |    |                      ^          ^          ^
            |    |    +----------------------|----------|----------+
            |    +---------------------------|----------+
            +--------------------------------+

* Aufteilung der Nagios-Bestandteile auf System-Verz.:

  +--------------+-------------------------------------------------------------+
  | Verzeichnis  | Inhalt                                                      |
  +--------------+-------------------------------------------------------------+
  | bin          | Nagios-Daemon und Hilfsprogramme                            |
  | etc          | Konfigurations-Dateien                                      |
  | etc/objects  | Weitere Konfigurations-Dateien (pro Objekt/Objektklasse)    |
  | etc/config   | Weitere Konfigurations-Dateien                              |
  | etc/conf.d   | Weitere Konfigurations-Dateien                              |
  | libexec      | Plugins                                                     |
  | sbin         | GUI: CGI-Skripte                                            |
  | share/htdocs | GUI: Dokumentation + statische Teile                        |
  | stylesheets  | GUI: CSS-Dateien                                            |
  | var/log      | Log-Dateien                                                 |
  | var/lib      | Status-Daten + Lock-Dateien                                 |
  | usr/lib      | Plugins                                                     |
  +--------------+-------------------------------------------------------------+

* Kontrolle des Nagios-Daemon
  + STARTPUNKT "nagios.cfg" beim START von Nagios-Daemon als Parameter angeben
      /usr/local/nagios/bin/nagios /usr/local/nagios/etc/nagios.cfg   # Quellcode
      /usr/sbin/nagios3            /etc/nagios3/nagios.cfg            # Ubuntu-Paket
  + Nagios-Daemon starten/stoppen/...
      /etc/init.d/nagios3 start     # oder: service nagios3 start
      /etc/init.d/nagios3 stop      # oder: service nagios3 stop
      /etc/init.d/nagios3 restart   # oder: service nagios3 restart
      /etc/init.d/nagios3 reload    # oder: service nagios3 reload
      /etc/init.d/nagios3 status    # oder: service nagios3 status
  + Konfiguration nach JEDER Änderung auf Korrektheit "verifizieren" (-v=verify):
      /usr/sbin/nagios3             -v /etc/nagios3/nagios.cfg
      /usr/local/nagios/sbin/nagios -v /usr/local/nagios/etc/nagios.cfg


15) Warum keine GUI zur Konfiguration?
--------------------------------------

* Verleitet zum Anfangen ohne Vorüberlegung und Nagios-Einarbeitung ;-)

* Zuviele Einstellungsmöglichkeiten in Konfigurations-Dateien machen
  Realisierung vollständiger GUI schwer (mit jeder Nagios-Version mehr)

* Namensvergabe für Objekte immens wichtig --> Keinem Tool überlassen!

* Bindung an GUI --> Was ist wenn GUI nicht mehr weiterentwickelt wird?
  (schon mehrfach passiert)

* Es gibt "professionelle" GUI-Ansätze (gut und schlecht gleichermassen)
  + Warum "weitere Abstraktionsschicht/Metaebene" einziehen,
    deren Logik/Denkweise man zusätzlich verstehen muss?
  + Verstehen muss man Struktur/Arbeitsweise von Nagios sowieso!
  + Viele Daten in GUI erfassen unangenehm + fehleranfällig
  + Datensicherung und Wiederherstellung schwieriger
  + Schließt Konfiguration per Editor in Datei aus
    (Reihenfolge, Struktur, Formatierung, Kommentare gehen verloren)

* Namen, Struktur und Kommentare der Konfigurations-Dateien
  + Gehen beim Einlesen/Generieren der Konfiguration verloren
```

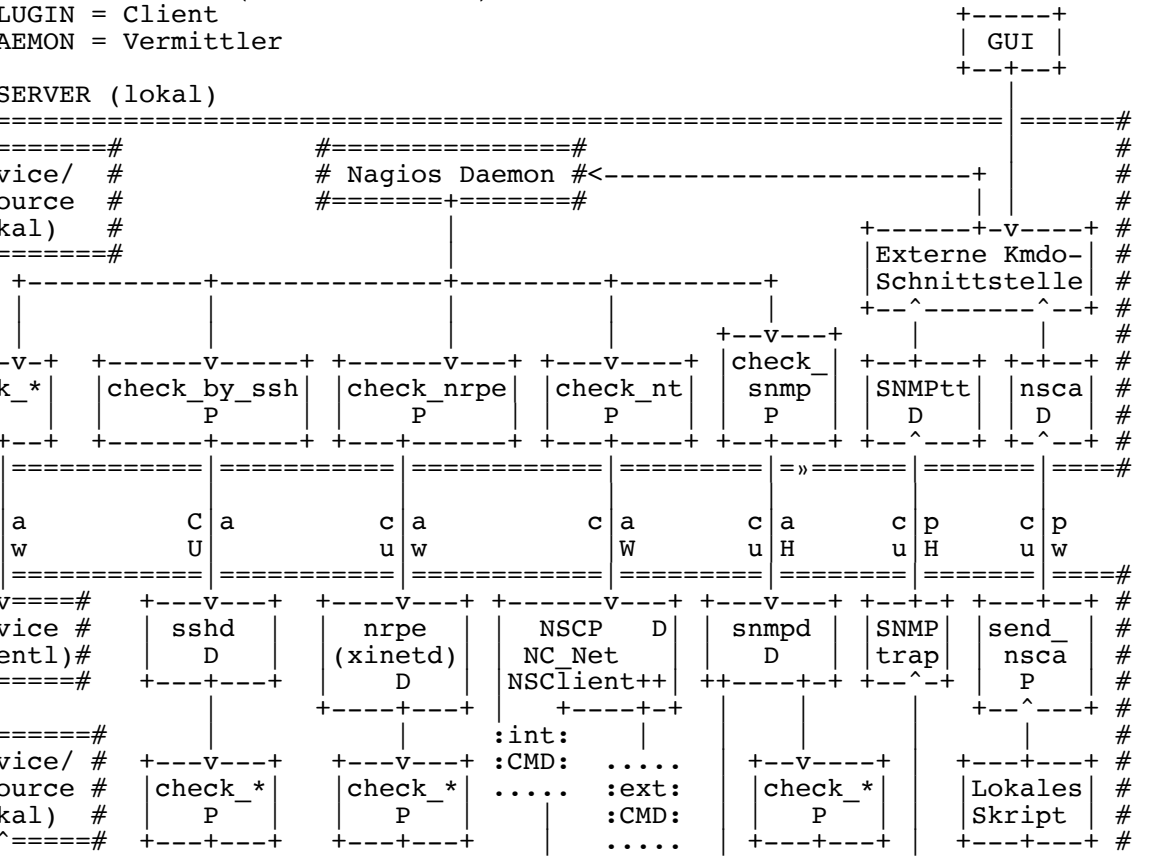
  + Automatisch nur starre Aufteilung machbar:
    - Nur eine Datei
    - Pro Objekttyp eine Datei?
    - Pro Objektgruppe eine Datei?

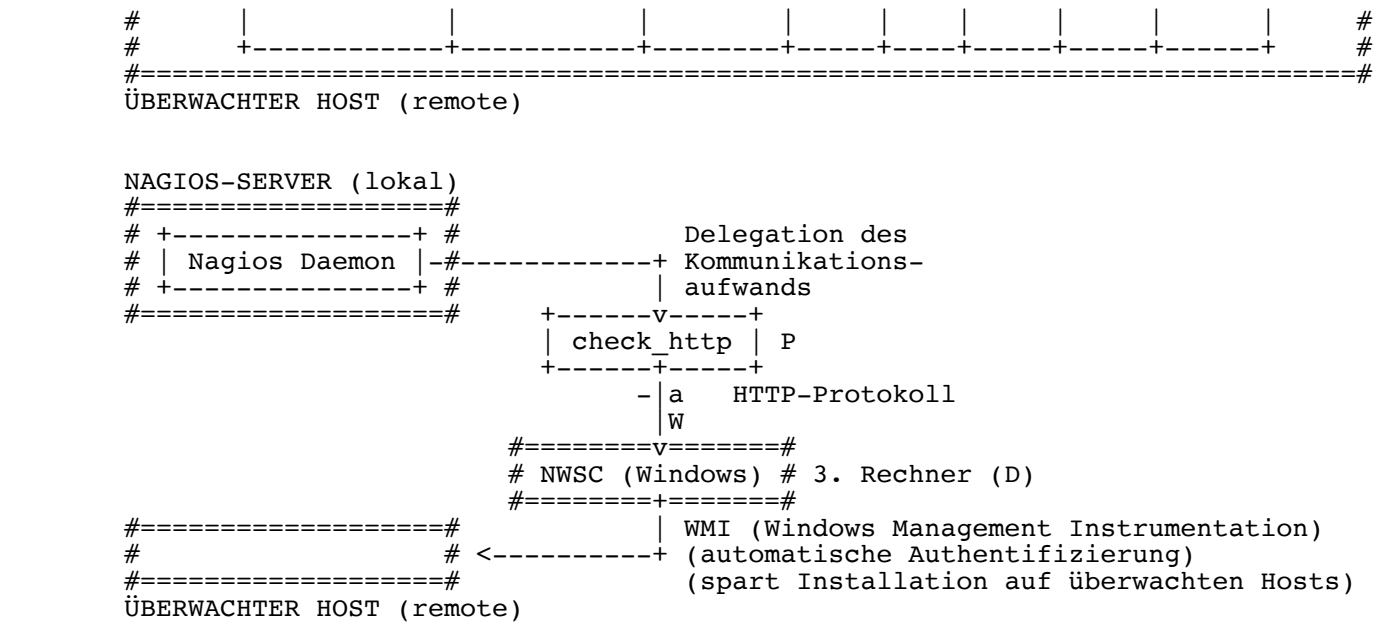
* Trotzdem hier eine Liste interessanter GUIs zur Nagios-Konfiguration:
  + Centreon
  + Fruity
  + Ignoramus (Configuration Generator)
  + LConf (LDAP-basiert)
  + Lilac
  + Monarch (Groundwork Monitor Agent)
  + Nag2Web
  + Nagat (Nagios Administration Tool)
  + Nagmin (Nagios Administration)
  + NagiosAdmin(istrator)
  + NagiosQL (PHP+MySQL-basiert)
  + Nagiosweb
  + NConf (Enterprise Nagios Configurator)
  + NCPL (Nagios Control Panel)
  + OneCMDB (Configuration Management Database)
  + OpCfg
  + Thruk (check_mk)
  + TKmon (Thomas Krenn)
  + WATO (Check_MK's Web Administration Tool)

16) Kommunikationsarten Nagios-Server <-> Plugins (GRAFIK)
---

* Auch remote von Nagios angestoßene Checks heißen "Plugin"!

* Übersicht Kommunikationsarten Nagios-Server <-> Plugins
    a = Active = Synchron = Ausgelöst vom Nagios-Daemon
    p = Passive = Asynchron = Extern ausgelöst (z.B. auf zu überwachendem System)
    K = Unverschlüsselt (Klartext)
    C = Verschlüsselt (encrypted)
    c = Wahlweise verschlüsselt
    H = Beliebige Hardware
    u = Unix/Linux-Rechner (meist)
    U = Unix/Linux-Rechner (ausschließlich)
    w = Windows-Rechner (meist)
    W = Windows-Rechner (ausschließlich)
    P = PLUGIN = Client
    D = DAEMON = Vermittler

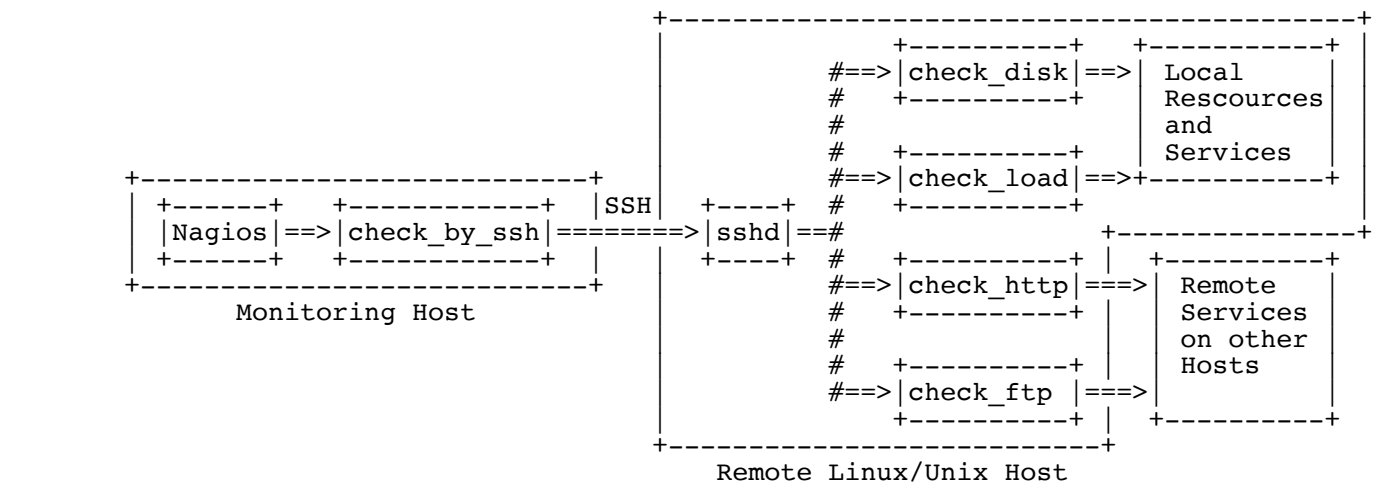




* Überwachung von Netzwerk-Diensten (öffentlicher Service) ist einfach und schnell einzurichten (Nagios ist auch nur Nutzer/Client eines solchen)
  + IP + Port
  + Evtl. Anmeldedaten notwendig
  + Evtl. Proxy notwendig
  + Verschiedene Prüfungsebenen:
    + Verbindung herstellbar (Dienst antwortet)
    + Protokoll-Startdialog korrekt (z.B. Dienst-Version ermitteln)
    + Protokoll-Abfrage mit korrekter Antwort (vollständige Kommunikation)

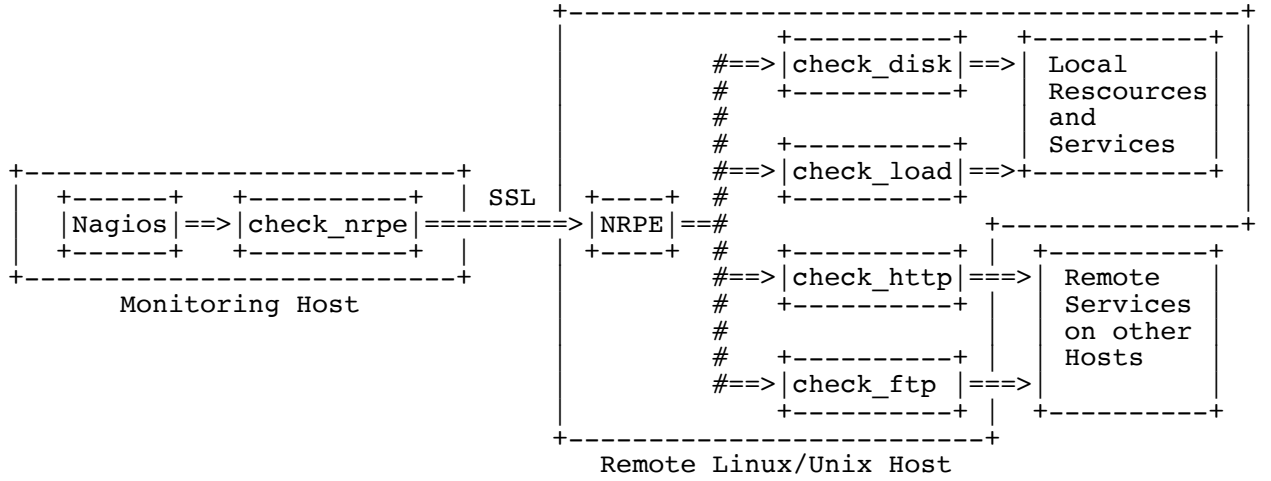
* Für Remote-Check lokaler Systemparameter/Dienste ist ein "Agent/Vermittler" (Helferlein, Bindeglied, Proxy) notwendig (SSH, NRPE, NSCP, SNMP, NSCA, NWSC)

* SSH
  + Daemon "sshd" oft bereits installiert (Linux)
  + Von Nagios angestoßen ("Active")
  + VORHER Schlüsselaustausch nötig für Anmeldung OHNE Passwort/Passphrase!
  + VORHER mind. 1x Verbindungsaufbau testen (evtl. Eingabe von "yes" notwendig)
  + Aufwendiger Verbindungsaufbau
    --> Q: "Da kommt die Brüh' teurer als das Fleisch"
  + Extrem sicheres "Rohr" auf Basis extrem unsicherer Konfiguration aufbauen
    - Durch das "Rohr" ist zunächst alles erlaubt
    - Missbrauchbar zum Ausführen beliebiger Kommandos (umkonfigurierbar)
  + Client "check_ssh" + Daemon "sshd" + Konfiguration "/etc/ssh/sshd_config"
  + sshd-Daemon + Plugins auf überwachtem Rechner notwendig

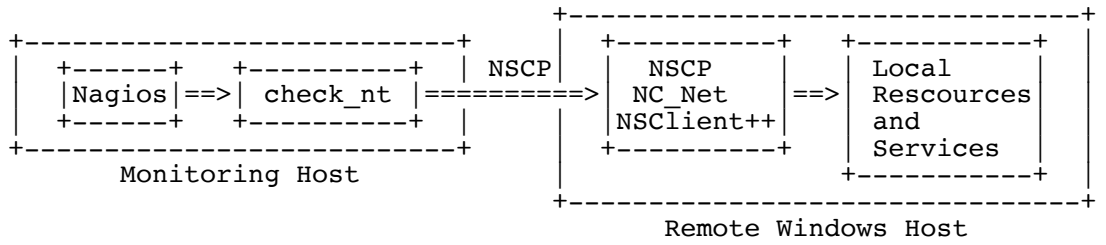


* NRPE (Nagios Remote Plugin Executor)
  + Analog SSH-Verbindung, aber "billiger"
  + Verfügbar für Linux, UNIX, BSD, MacOS, Windows
  + Von Nagios angestoßen ("Active")
  + Führt beliebig viele Tests aus ("indirekte Checks")
  + Reicht Ergebnis an Nagios zurück
  + Client "check_nrpe" + Daemon "nrpe" + Konfiguration "nrpe.cfg"

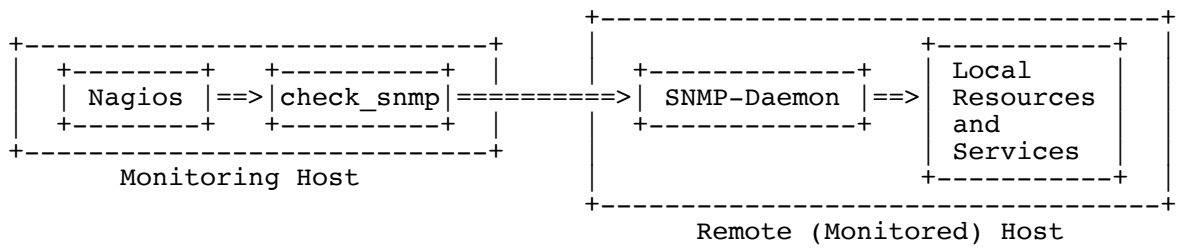
  + Verbindung unverschlüsselt und verschlüsselt (SSL) möglich
  + NRPE-Daemon/Dienst + Plugins auf überwachtem Rechner notwendig



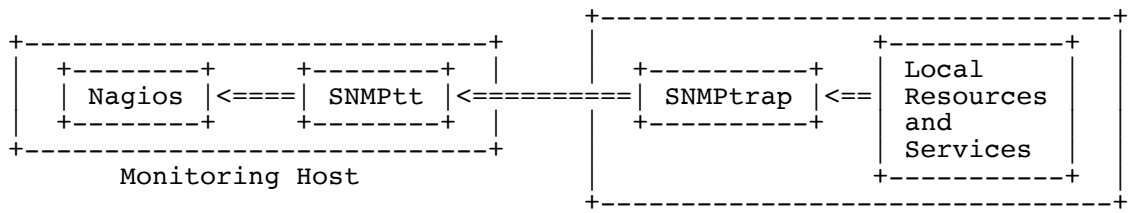
* NSCP (Nagios Service Check Protocol)
  + Für Windows (NC_Net, NSClient++) -- inzw. auch für Linux
  + Von Nagios angestoßen ("Active")
  + Keine Plugins auf Remote-Rechner notwendig
    - Daemon kann bereits alle Informationen ermitteln
    - Erweiterung kann auch beliebige Kommandos/Skripte remote ausführen
  + Client "check_nt" + Daemon "nscp" + Konfiguration "nsclient.ini/NSC.ini"
  + Verbindung unverschlüsselt und verschlüsselt (SSL) möglich
  + NSCP-Daemon/Dienst auf überwachtem Rechner notwendig



* SNMP (Simple Network Management Protocol)
  + Daemon "snmpd" oft bereits installiert (Linux)
  + (Fast) alle netzwerkfähige Geräte (Router, Switch, Drucker) dazu fähig
  + Neben Lesen von Daten auch Schreiben von möglich (Anweisungen/Konfiguration)
  + Existenz einer gewissen Anz. von Prozessen überwachen ist eingebaut
  + Implementierung oft nicht dokumentiert bzw. unvollständig/fehlerhaft
    (z.B. bei Cisco)
  + Kann beliebige Skripte auf Client-Seite aufrufen, d.h.
    missbrauchbar zum Ausführen von Kommandos (Linux: snmpd + exec)
  + Client "check_snmp" + Daemon "snmpd" + Konfiguration "/etc/snmp/snmpd.conf"
  + Verbindung unverschlüsselt und verschlüsselt (ab SNMPv3) möglich

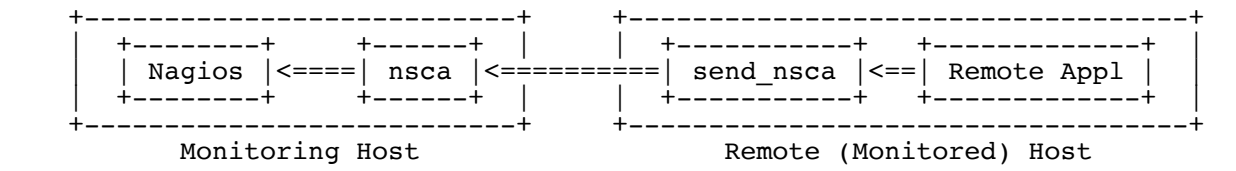


* SNMPtrap (Simple Network Management Protocol)
  + Wird von sich aus tätig
  + Liefert Ergebnisse an Nagios zurück ("Passive")

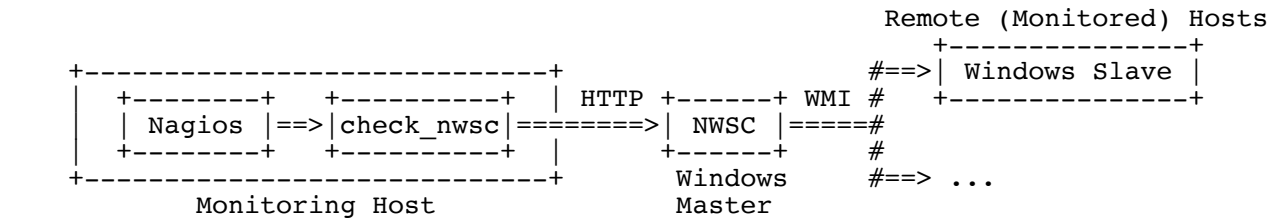


Remote (Monitored) Host

* NSCA (Nagios Service Check Acceptor)
  + Wird von sich aus tätig
  + Liefert Ergebnisse an Nagios zurück ("Passive")
  + Erlaubt verteilte und mehrstufige Nagios-Setups
  + Auslösen z.B. mit Cron/NSClient++/Windows-Taskmanager
  + Verbindung unverschlüsselt und verschlüsselt möglich
  + Sinnvoll z.B. für extreme Langläufer auf Remote-Seite
    (Ergebnis liegt irgendwann vor, z.B. Backup)
  + Aufwand gegenüber aktivem Checkaufruf niedriger
    (Nagios kann mehr überwachen)



* NWSC (Nagios Windows Service Checker)
  + Auf EINEM Windows Rechner zu installieren (Perl + Module)
  + Muss Domänenenadministrator-Rechte haben
  + WMI-Abrufe von anderen Windows-Rechner (Windows Management Instrumentation)
  + Aufträge per HTTP-Anfrage
  + Rückgabe der WMI-Daten per HTTP-Antwort



* Check_MK (Mathias Kettner --- teilweise kostenpflichtig)
  + Ersatz + Erweiterung von NRPE/NSClient/...
  + NRPE --> MRPE (MK Remote Plugin Executor)
  + Administration per Web-GUI "WATO" (Check_MK's Web Administration Tool)
  + Check-Klassifizierung
  + Für Windows (fertiges EXE) + Linux + ...
  + Windows Events monitorbar
  + Verschlüsselt
  + Erweiterbar mit beliebiger Programmiersprache
  + Nötig zum Übersetzen: g++, gdlib, xml2lib, ...
  + BÜNDELUNG aller Service-Checks für jeden Rechner (EINE Abfrage statt N)
  + Teil von OMD = Open Monitoring Distribution (kostenpflichtig, Support)
  + Keine Konfiguration nötig --> MK-Livestatus + MK-Multisite
    --> Konfigurations-Dateien automatisch erzeugt
    - Viele kleine
    - Nicht mehr editierbar weil wieder automatisch überschrieben
  + --> "Netzwerk-Monitoring mit OMD und Check_MK" in c't 24/2012, Seite 190

  Allzweck-Nagios-Plugin zum Lesen von Daten. Verfolgt neuen Ansatz zum Sammeln
  der Daten von Betriebssystemen und Netzwerk-Komponenten. Macht "NRPE",
  "check_by_ssh", "NSClient" und "check_snmp" überflüssig und hat viele
  Vorteile, insbesondere eine signifikante Reduktion des CPU-Verbrauchs auf dem
  Nagios-Host und eine automatische Inventarisierung der Objekte auf den zu
  prüfenden REMOTE-Hosts. Je größer die Nagios-Installation, desto hilfreicher.

* Weitere Vermittler/Bindeglieder/Plugins:
  + check_multi   #
  + NCPA   = Nagios Cross Platform Agent
  + NERD   = Nagios Event Radio Dispatcher (Query Handler)
  + NRDP   = Nagios Remote Data Processor
  + NSTI   = Nagios SNMP Trap Interface
  + SNMPTT = SNMP Trap Translator
  + ...

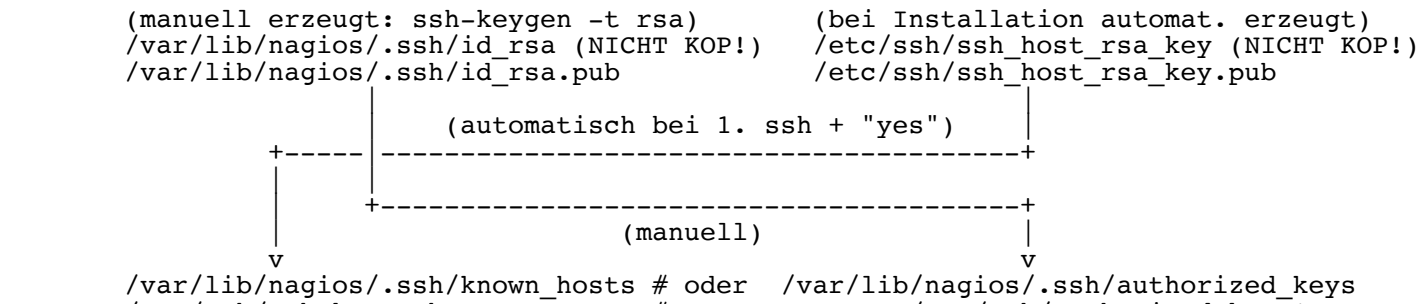
17) Check-Agent Konfiguration
---

* Check-Agent: Verbindet Nagios-Daemon und Remote-Plugin (Server-Client-Prinzip)
  + SSH (Secure Shell)   #   (Aktiv)

```
  + NRPE (Nagios Remote Plugin Executor)    # Analog snmpd             (Aktiv)
  + NSCA (Nagios Service Check Acceptor)    # Analog SNMPTrap          (Passiv)
  + NSCP (Nagios Service Check Protocol)    # Analog NRPE für Windows  (Aktiv)
  + NWSC (Nagios Windows Service Checker)   # WMI für Windows          (Aktiv)
  + SNMP     # Beliebige Skripte auf Client-Seite aufrufen            (Aktiv)
  + SNMPTrap # Automatisch Skript auf Client --> Nagios-Server Cmdfile (Passiv)
```

17.1) SSH (check_by_ssh)

```
* Einfachstes Werkzeug zum Überwachen von REMOTE-Linux-Systemen
  + "Kleiner Wrapper" rund um beliebiges Check-Plugin
    (führt vor und nach Plugin etwas zusätzliche Logik durch)
  + Auf REMOTE keine Installation weiterer Software notwendig
    außer dort aufzurufendes check_...-Skript (oder einfach alle Nagios-Plugins)

* Erfordert "echte Anmeldung" auf REMOTE-System als Benutzer "nagios"
  --> Vollwertiger Zugriff auf Kommandozeile
  --> Benutzeraccount mit schlüsselbasierter Anmeldung (ohne Passphrase) MUSS für
      Benutzer "nagios" auf REMOTE-System vorhanden sein (Login-Shell, Homeverz.)

* 5 Parameter
  + Benutzername           # Auf REMOTE-System (implizit "nagios" oder -l USER)
  + SSH-Port               # Auf REMOTE-System (implizit "22" oder -p PORT)
  + Zu nutzendes Keyfile   # Auf lokalem System (implizit, keine Passphrase)
  + Aufzurufendes Plugin   # Auf REMOTE-System
  + Parameter für Plugin   # Auf REMOTE-System

* Beispiel (als Benutzer "nagios" manuell durchgeführt!)
    ssh nagios@REMOTE
    /usr/lib/nagios/plugins/check_disk -w 10 -c 5 /
    exit
  ersetzbar durch (als Benutzer "nagios" durchgeführt!)
    sudo -u nagios \
      /usr/lib/nagios/plugins/check_by_ssh -H REMOTE \
        -C "/usr/lib/nagios/plugins/check_disk -w 10 -c 5 /"

* Einrichten auf NAGIOS:
   su - nagios            # --> /var/lib/nagios (Heimatverz. Nagios-Benutzer)
   ssh-keygen -t dsa      # --> /var/lib/nagios/.ssh/id_dsa
                          #     /var/lib/nagios/.ssh/id_dsa.pub
   ls -l ~/.ssh           # --> id_dsa
                          #     id_dsa.pub

* Einrichten auf REMOTE:
   ssh REMOTE                                                  # Auf REMOTE anmelden
   sudo useradd -m -d /var/lib/nagios -s /bin/bash             # User anlegen
   su - nagios                                                 # nagios anmelden
   mkdir .ssh                                                  # Verz. anlegen
   exit                                                        # nagios abmelden
   exit                                                        # REMOTE verlassen
   scp .ssh/id_dsa.pub nagios@REMOTE:.ssh/authorized_keys      # Öff. Key kopieren
   ssh-copy-id nagios@REMOTE                                   # (alternativ)

* Einrichten auf NAGIOS:
   su - nagios                                                 # Nagios-Benutzer
   ssh nagios@REMOTE                                           # 1x anm. + "yes"
   exit                                                        # REMOTE verlassen

* Ausprobieren auf NAGIOS:
   su - nagios                                                 # Nagios-Benutzer
   ssh nagios@REMOTE                                           # Muss
   sudo -u nagios \                                            #  ohne
     /usr/lib/nagios/plugins/check_by_ssh -H REMOTE \          #   Passwort
       -C '/usr/lib/nagios/plugins/check_disk -w 10 -c 5 /' #    funktionieren!

* Letzten Endes hat folgender Austausch des öffentlichen Schlüssels "*.pub" der
  jeweiligen Gegenstelle "über Kreuz" stattgefunden:

                              ANMELDUNG
          ssh nagios@REMOTE ==========================> sshd
            +---------+         SSH-Protokoll       +----------+
            | NAGIOS  |                             |  REMOTE  |
            +----+----+                             +-----+----+
                 |                                        |
```



17.2) NRPE (Nagios Remote Plugin Executor)

* Indirekte Checks analog SNMP

* Verglichen mit "check_by_ssh"
  + Keine echte Anmeldung, sondern Protokoll (Port 5666)
  + Minimal höherer Konfigurationsaufwand
    - NRPE-Dienst auf überwachten Rechnern installieren + konfigurieren
  + Erheblicher Sicherheitsgewinn
    - Liste der verfügbaren Kommandos auf überwachtem Rechner konfiguriert

* Detailliert steuerbar
  + Welche Kommandos Nagios auf überwachtem Rechner ausführen darf
  + Sogar welche Parameter Nagios dabei verwenden darf
  + Es sind aber auch Parameter übergebbar

* Paket "nrpe(-server)" (bzw. NSClient++ unter Windows) auf REMOTE und
  Plugin "check_nrpe" auf NAGIOS installieren

* Notwendige Tools
  + Client "check_nrpe" # Auf NAGIOS-Host
  + Server "nrpe" (NRPE-Dienst) # Auf überwachtem Rechner REMOTE

* Konfiguration in "nrpe.cfg" auf überwachtem Rechner
```
  /etc/nagios/nrpe.cfg            # Dienst auf überwachten Rechner
     allow_hosts = 192.168.1.10                    # Ein Rechner
     allow_hosts = 192.168.1.10,192.168,1.11,...   # Liste von Rechnern
     allow_hosts = 192.168.1.0/24                  # Netzwerk
     libexecdir = /usr/lib/nagios/plugins
     command[check_users] = @libexecdir@/check_users -w 5 -c 10
     command[check_load]  = @libexecdir@/check_load -w 15,10,5 -c 30,25,20
     command[check_disk1] = @libexecdir@/check_disk -w 20 -c 10 -p /dev/hda1
     command[check_disk2] = @libexecdir@/check_disk -w 20 -c 10 -p /dev/hdb1
  /etc/init.d/nrpe reload   # oder service nrpe reload
```

* Konfiguration in "nrpe.cfg" auf Nagios-Host
```
  /etc/default/nagios-nrpe-server  # Dienst auf Remote-Host
  /etc/nagios/commands/nrpe.cfg    # Client auf Nagios-Host
```
```
+-------------------------------------------------------------------+
| define command {                                                  |
|   command_name  check_nrpe                                        |
|   command_line  $USER1$/check_nrpe -H $HOSTADDRESS$ -c "$ARG1$"   |
| }                                                                 |
+-------------------------------------------------------------------+
| define service {                                                  |
|   use                   service-template-interval-1min            |
|   service_description   LOAD                                      |
|   host_name             linux1                                    |
|   check_command         check_nrpe!check_load                     |
| }                                                                 |
+-------------------------------------------------------------------+
```

* Alternative Konfigurations-Datei
```
  /etc/nagios/nrpe-local.cfg    # Client auf Nagios-Host
  /etc/nagios/nrpe.d/*          # Client auf Nagios-Host
```

17.3) NSCA (Nagios Service Check Acceptor)

* Paket "nsca(-server)" auf NAGIOS und "nsca-client" auf REMOTE installieren

```
  (bzw. NSClient++ auf Windows-REMOTE)

* Konfiguration auf Nagios-Server
  /etc/nagios3/nagios.cfg
    check_external_commands = 1             # Datei mit NSCA-Nachrichten lesen
    command_check_interval  = -1            # Möglichst oft
    command_file            = /var/lib/nagios3/rw/nagios.cmd  # Kommando-Datei
  /etc/nsca.cfg
    server_port    = 5667                                 # Default
    server_address =                                      # Alle Interfaces
    nsca_user      = nagios                               # Analog nagios-Dienst
    nsca_group     = nogroup                              # (irrelevant)
    command_file   = /var/lib/nagios3/rw/nagios.cmd       # Analog nagios-Dienst
    password       = geheim                               # Gemeinsames Geheimnis
    decryption_method = 3                                 # 3=3DES

* NSCA-Daemon manuell starten
  (automatischer Start beim Reboot bei Installation eingerichtet)
    sudo /etc/init.d/nsca start     # SysVinit - Dienst manuell aktiv
    service nsca start              # upstart  - Dienst manuell aktiv
    systemctl start nsca.service    # systemd  - Dienst manuell aktiv
    systemctl enable nsca.service   #          - Dienst automatisch aktiv
    systemctl disable nsca.service  #          - Dienst automatisch inaktiv

* Konfiguration auf Client-Rechnern (u1,u2,u3)
  + Linux "/etc/send_nsca.cfg":
      password          = geheim      # Gemeinsames Geheimnis
      decryption_method = 3           # 3=3DES
  + Windows "C:\Programs\NSClient++\nsclient.ini":
      [/settings/NSCA/client]
      hostname=win_${host}
      [/settings/NSCA/client/targets/default]
      address           = NAGIOS      # Nagios-Server
      password          = geheim      # Gemeinsames Geheimnis
      decryption_method = 3           # 3=3DES
      use ssl           = 0           # SSL nicht benutzen

* Testen auf NAGIOS (-d=delimiter, Status 3=CRITICAL)
    echo "localhost;3;CRITICAL TestNSCA..." |
       send_nsca -H localhost -d ";"
  Muss auf NAGIOS auftauchen in "/var/log/nagios.log":
    grep "TestNSCA..." /var/log/nagios3/nagios.log
    --> [1430904142] EXTERNAL COMMAND: PROCESS_HOST_CHECK_RESULT;localhost;3;CRITICAL
TestNSCA...

* Nachricht von Client an Nagios-Server schicken (per NSCA-Protokoll)
  + Zeile mit 4/3 per Trennzeichen (Std: Tabulator) getrennten Textfeldern
    (eine oder mehrere Zeilen erlaubt, z.B. am Ende eines Backup-Skriptes)
      HOST                STATUS_CODE  MESSAGE         # Host-Check
      HOST  SERVICE_DESC  STATUS_CODE  MESSAGE         # Service-Check
  + Beispiel (von REMOTE=10.0.2.90 an NAGIOS=10.0.2.1)
    - Host-Nachricht (Status 0 = UP)
        echo "10.0.2.90;0;OK Test..." |                # -H=NAGIOSHOST
          send_nsca -H 10.0.2.1 -d ";"                 # -d=DELIMITER
    - Service-Nachricht (Status 0 = OK)
        echo "10.0.2.90;TestService;0;OK Test..." |    # -H=NAGIOSHOST
          send_nsca -H 10.0.2.1 -d ";"                 # -d=DELIMITER

* Nachricht von Client an Nagios-Server schicken (per SSH-Protokoll)
  + Zeile mit 4/5 per Trennzeichen ";" getrennten Textfeldern
      PROCESS_HOST_CHECK_RESULT     HOST                STATUS_CODE  MESSAGE
      PROCESS_SERVICE_CHECK_RESULT  HOST  SERVICE_DESC  STATUS_CODE  MESSAGE
  + Beispiel (von REMOTE=10.0.2.90 an NAGIOS=10.0.2.1)
    - HOST-Nachricht (Status 0 = UP)
        echo -e "PROCESS_HOST_CHECK_RESULT;10.0.2.90;0;OK TestNSCA..." |
          ssh 10.0.2.1 "cat > /var/lib/nagios3/rw/nagios.cmd"
    - Service-Nachricht (Status 0 = OK)
        echo -e "PROCESS_SERVICE_CHECK_RESULT;10.0.2.90;Passive_Test;0;OK TestNSCA..."
|
          ssh 10.0.2.1 "cat > /var/lib/nagios3/rw/nagios.cmd"

* Service-Check PASSIV einstellen mit Timeout von 1 Stunde = 3600s
  + check_command + freshness_threshold: Kommando "check_dummy" nur dann
    1x ausgeführt, wenn länger als 3600s keine Service-Nachricht "Passive_Test"
    am Host u1/u2/u3 ankommt
  + is_volatile: Jede einzelne Statusänderung zählt (alle als HARD gewertet!)
```

```
+-------------------------------------------------------+
| define service {                                      |
|   use                     generic-service             |
|   check_freshness         1                           |
|  *freshness_threshold     3600                        |
|   host_name               u1,u2,u3                    |
|   service_description     Passive_Test                |
|   check_command           check-dummy!2!No Backup     |
|  *active_checks_enabled   0                           |
|  *passive_checks_enabled  1                           |
|  *is_volatile             1                           |
|   contact_groups          admins                      |
| }                                                     |
+-------------------------------------------------------+
```

17.4) NSCP (Nagios Service Check Protocol)

* Umfasst NC_Net, NSClient++

* Benutzt intern NRPE-Protokoll

```
  C:\Programme\NSClient++                # Installation
  C:\Programme\NSClient++\nsclient.ini   # Konfigurations-Datei (NSC.ini)
    +-------------------------------------------------------------------------+
    | [/modules]                                                              |
    | NRPEServer     = 1                  # NRPE Anfragen OK                  |
    | NSClientServer = 1                  # NSClient (check_mt) Anfragen OK   |
    | CheckSystem    = 1                  # Systemprüfungen durchführen       |
    | NSCAClient     = 1                  # Passive NSCA-Checks OK            |
    |                                                                         |
    | [/settings/NSCA/client/targets/default]                                 |
    | allowed hosts  = 10.10.2.0/24       # Rechner dürfen Verb. aufnehmen    |
    | address        = 10.10.2.1          # Nagios-Host Ziel                  |
    | encryption     = 3des               # Identisch zu NSCA-Dienst          |
    | password       = geheim             # Identisch zu NSCA-Dienst          |
    | use ssl        = false              # Identisch zu NSCA-Dienst          |
    +-------------------------------------------------------------------------+
```

* Dienst einrichten unter Windows

```
    C:\Programme\NSClient++\nscp service --install   # Dienst einrichten
    C:\Programme\NSClient++\nscp service --start     # Dienst starten
    C:\Programme\NSClient++\nscp service --stop      # Dienst starten
    C:\Programme\NSClient++ /install                 # Dienst einrichten
    net start nsclient++                             # Dienst starten
    net stop nsclient++                              # Dienst stoppen
    nsclient++ /start                                # Dienst starten
    nsclient++ /stop                                 # Dienst stoppen
```

* Prüfung ausführen auf Nagios-Server

```
    check_nt + check_nrpe                            # Auf Nagios-Server-Seite aufrufen
    check_nt -->
CLIENTVERSION,CPULOAD,UPTIME,MEMUSE,USEDDISKSPACE,SERVICESTATE,PROCSTATE
                Alle Counter von Windows-PerfMon (Leistungsprotokolle und Warnungen)
    # -H=hostname/ip -v=variable -p=port -s=secret -l=parameter
```

* Beispiel

```
    check_nt -H $HOSTNAME$ -p 12478 -v COUNTER -l "\\Leistungsobjekt(Instanz)\
\Indikator"
    check_nt -H $HOSTADDRESS$ -v COUNTER -l \
      "\\Auslagerungsdatei(_Total)\\Belegung (%),"Paging File usage is %.2f %%" \
      -w $ARG1$ -c $ARG2$"
    $USER1$/check_nt -H $HOSTNAME$ -v SERVICESTATE -p 12489 -s $USER4$ -l "$ARG1$"
```

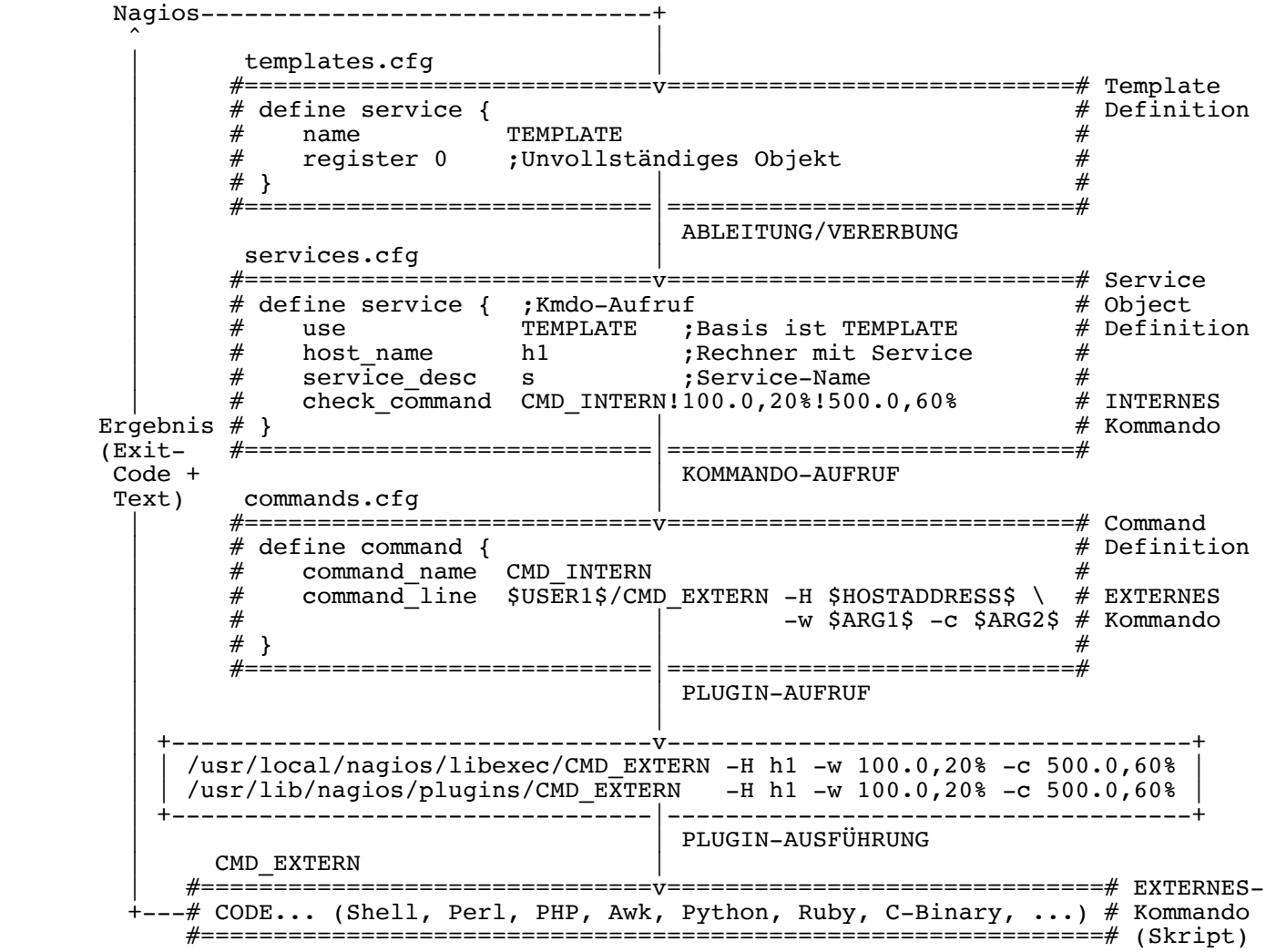
17.5) NWSC (Nagios Windows Service Checker)

* "Mittler", der Aufrufe von Windows-WMI (Windows Management Instrumentation)
  zur Verfügung stellt, die eine Linux-Maschine nicht bieten kann.
* Trotzdem muss NWSC nur auf auf EINEM Windows-Rechner installiert werden.
  --> http://www.ostc.de/projects.html#nwsc

18) Was ist bei Plugins zu beachten?

* Plugins IMMER auf Kommandozeile als BENUTZER "nagios" vollständig austesten
  + Unbedingt machen, um Fehlersuche zu vereinfachen!
    (Pfade, Parameter, Quotierung, Zugriffsrechte, ...)
      sudo -u nagios PLUGIN-KOMMANDO...   # Ausgabetext ansehen
      echo $?   # Exit-Status anzeigen
    --> Enormer Vorteil!

* Nagios-Kommando
  + Verbindet Nagios-Daemon und externe Plugins
  + Durchführung von
    - Check
    - Notification
    - Event-Handler
  + In ZWEI SCHRITTEN definiert
    - INTERNER Kommandoname für Kommando-Aufruf in OBJEKTEN
      . PARAMETER per "!" getrennt
      . Reihenfolge von 1..n numeriert
    - EXTERNER Kommandoname für Plugin-Aufruf in INTERNEM Kommando
      . PARAMETER per $ARG1$ $ARG2$ ... bezeichnet
      . Numerierung entspricht Reihenfolge beim Kommando-Aufruf
  + Grund: Mehrere interne Kommandos nutzen gleiches externes Kommando
  + Verwirrend: Interner + externer Kommandoname oft gleichlautend!
  + Makro $USER1$ verwenden, enthält Pfad zu Plugin-Verz.



* Wo Schwellwert-Parameter für Plugin eintragen?

Ort	Relevant für...
register.cfg	...Nagios-Gesamtsystem
Template	...jeden Host/Service, der Template benutzt (Default)
Host/Service	...diesen Host/Service
Externes Kommando	...jeden Aufruf externes Kommando (Quatsch?)
Plugin-Skript	...jeden Aufruf des Skriptes (Quatsch?)

* Plugin-Verz.: /usr/local/nagios/libexec/* # Quellcode-Installation
                /usr/lib/nagios/plugins/   # Paket

18.1) Plugin-Schnittstelle

* Nagios schreibt Kommunikation mit Plugins genau vor
  + Kommando mit bestimmten Optionen (Bedeutung fest definiert)
  + Exit/Returncode 0/1/2/3 je nach Status OK/WARNING/CRITICAL/UNKNOWN
  + Ausgabe ein/mehrzeiliger Text bestimmter Struktur (max. 300 Byte / 8 KByte)
  + Plugin-Ausgabe darf nach "|" Performance-Daten ("PerfData") enthalten
    --> Langfristige Trends erkennen
  + Usage-Meldung bei Aufruf ohne Parameter bzw. mit "-h/--help"
    --> Mini-Doku: in vielen Fällen ausreichend für Verständnis

* Struktur EINZEILIGER Plugin-Meldungen (V2, max. 300 Byte)
  (STATUS = OK, WARNING, CRITICAL, UNKNOWN bzw. UP, DOWN, UNREACHABLE)
  + Trenner " " und " - " vorgeschrieben
  + Trenner "|" darf fehlen, wenn keine Performance-Daten vorhanden

```
  +-----------------------------------------------------------+
  | CHECKNAME STATUS - INFOTEXT | OPTIONAL PERFORMANCEDATA    |
  +-----------------------------------------------------------+
   <--------------------------> <------------------------>
       Für Web-Interface            Performance-Data
         + Alarm-E-Mail
```

  + Beispiele
    SMTP OK - 0 seconds response time
    CHECKSAP OK - system sap123 available
    PROCS WARNING - 4 processes with command name 'pppoe'
    DISK CRITICAL - free space: /home 8627 MB (64% inode=73%);| /=4740MB;...

* Struktur MEHRZEILIGER Plugin-Meldungen (V3, max. 8192 Byte)
  (STATUS = OK, WARNING, CRITICAL, UNKNOWN bzw. UP, DOWN, UNREACHABLE)

```
  +-----------------------------------------------------------+
  | CHECKNAME STATUS - INFOTEXT | OPTIONAL PERFORMANCEDATA    | Trenner
  | LANGTEXT 1.ZEILE                                          | " ",  " - ", "|"
  | LANGTEXT 2.ZEILE                                          | vorgeschrieben
  | ...                                                       |
  | LANGTEXT N.ZEILE | OPTIONAL PERFORMANCEDATA 2.ZEILE       | Trenner " | "
  | OPTIONAL PERFORMANCEDATA 3.ZEILE                          | vorgeschrieben
  | ...                                                       |
  | OPTIONAL PERFORMANCEDATA N.ZEILE                          |
  +-----------------------------------------------------------+
```

  + Beispiel

* Sauber definierte einfache "Kommando-Schnittstelle" (Plugin-API)
  --> Plugins unabhängig vom Nagios-Kern
  --> Programmiersprache beliebig (UNIX-Kommando in C, Shell, Perl, PHP, ...)
  --> Plugins bleiben kompatibel (egal ob für Nagios V1/2/3/4/... realisiert)
  --> Plugins leicht weiterzuentwicklen, ergänzen, korrigieren
  --> Plugin-Skripte müssen sich aber EXAKT daran halten!

* Exit-Code-Bedeutung von Plugins (= Status)

Code	Service	Host	Farbe
0	OK	UP	grün
1	WARNING	UP or DOWN/UNREACHABLE	gelb
2	CRITICAL	DOWN/UNREACHABLE	rot
3	UNKNOWN	DOWN/UNREACHABLE	grau

* Optionen mit reservierter Bedeutung für Plugins
  (N=Zahl, können noch weitere spezifische Optionen haben)

Kurz	Lang	Bedeutung
-w N	--warning=N	Schwellwert für Warnung-Grenze(n)

-c N	--critical=N	Schwellwert für kritische Grenze(n)
-h	--help	Online-Hilfe ausgeben (ggf. kurz+lang)
-V	--version	Plugin-Versionsnummer ausgeben
-H N	--hostname=N	Rechner-Name/IP
-t N	--timeout=N	Wartezeit bis Check-Abbruch (Sek)
-4	--use-ipv4	IPv4 benutzen
-6	--use-ipv6	IPv6 benutzen
-v	--verbose	Mehr Ausgaben erzeugen (geschwätzig)
-v N	--verbose=N	Analog (Level 0..3)

* Häufig verwendete Optionen (nicht immer gleich)

Kurz	Lang	Bedeutung	
-C N	--community=N	Community-Name bei SNMP-Abfragen	("public")
-u N	--user(name)=N	Benutzer-Name für Anmeldungen	
-p N	--passw(or)d=N	Passwort für Anmeldungen	
-a N	--authentication=N	Passwort oder REALM (Kerberos, SNMPv3)	
-l N	--login/logname=N	Login-Name für Anmeldungen	
-u N	--url=N	URL	
-p N	--port=N	Portnummer	

* Allgemeines Format von Bereichsangaben (Grenzwerte/Thresholds für "-c" und "-w"):

```
[@]START:END
```

Vorgaben für START und END:
1. START <= END muss gelten
2. START: darf fehlen --> START = 0
3. END darf fehlen --> END = unendlich (inf)
4. "~" bedeutet negativ unendlich (-inf)
5. Std: Nachricht wenn Wert AUSSERHALB Bereich START..END (inkl. Grenzen)
6. "@" am Anfang --> Benachrichtigung wenn Wert INNERHALB (inkl. Grenzen)

Hinweis: Noch nicht alle Plugins akzeptieren Bereiche in diesem Format.
Für mehrere Grenzwerte gleichzeitig noch keine allgemeine Form festgelegt.

Beispiele für Bereichsformate:

Format	Bedeutung		
10	< 0 oder > 10	außerhalb Bereich	[0 .. 10]
:10	< 0 or > 10	" "	[0 .. 10]
10:	< 10	" "	[10 .. inf]
~:10	> 10	" "	[-inf .. 10]
10:20	< 10 oder > 20	" "	[10 .. 20]
@10:20	>= 10 und <= 20	innerhalb "	[10 .. 20]

Beispiele für Kommandozeilen:

check_...	Bedeutung
... -w10 -c20	CRITICAL wenn über 20, Warn wenn über 10 (CRITICAL wenn weniger als 0)
... -w~:10 -c~:20	Analog, negativer Werte --> OK
... -w10: -c20	CRITICAL wenn über 20, sonst Warn wenn unter 10 (CRITICAL wenn weniger als 0)
... -c1:	CRITICAL wenn weniger als 1
... -w~:0 -c10	CRITICAL wenn über 10, Warn wenn über 0
... -c@5:6	Noncritical Bereich ist 5:6, Rest CRITICAL
... -c10:20	CRITICAL wenn im Bereich 10 bis 20

* Plugin Verbosity Levels (-v)

Level	Ausgabeform
0	Eine Zeile, minimale Ausgabe: Zusammenfassung
1	Eine Zeile, zusätzliche Daten (z.B. Liste fehlerhafter Prozesse)
2	Mehrzeilig, Debugausgabe der Konfiguration (z.B. ps-Kommando)
3	Mehrzeilig, viele Details zur Problemsuche in Plugins

* Beispiel Partition "/home" auf freien Platz überprüfen
  (Meldung wenn WENIGER als NN% FREI)

```
    df -h /home
    # --> Filesystem Size  Used Avail Use% Mounted on
    # --> /dev/sda7   14G  4.7G  8.5G  36% /home   # --> Freier Platz: 64%

    check_disk -w 10% -c 5% /home
    # --> DISK OK - free space: /home 8627 MB (64% inode=73%);| /=4740MB;
12674;13378;0;14083
    #     da freier Platz 64% größer 10% (Warning) und größer 5% (Critical)
    echo $?   # --> 0 (OK)

    check_disk -w 70% -c 60% /home
    # --> DISK WARNING - free space: /home 8627 MB (64% inode=73%);| /=4740MB;...
    #     da freier Platz 64% kleiner 70% (Warning) aber größer 60% (Critical)
    echo $?   # --> 1 (WARNING)

    check_disk -w 90% -c 80% /home
    # --> DISK CRITICAL - free space: /home 8627 MB (64% inode=73%);| /=4740MB;...
    #     da freier Platz 64% kleiner 90% (Warning) und kleiner 80% (Critical)
    echo $?   # --> 2 (CRITICAL)

    check_disk -w 90% -c 80% /xyz
    # --> DISK UNKNOWN - /xyz is not accessible: No such file or directory
    echo $?   # --> 3 (UNKNOWN)
```

* Programmierung von Plugins (Minibeispiele für Dummy-Plugins):

+ Shell:
```
#!/bin/bash                                          # OK/0
echo "DISK OK - free space: ..."
exit 0
```

+ Perl:
```
#!/usr/bin/perl -w                                   # WARNING/1
use strict;
printf "DISK WARNING - free space: ...\n";
exit 1;
```

+ Awk:
```
#!/usr/bin/awk -f                                    # CRITICAL/2
BEGIN {
  printf("DISK CRITICAL - free space: ...\n")
  exit 2
}
```

+ PHP:
```
#!/usr/bin/php -f                                    # UNKNOWN/3
<?php
    printf("DISK UNKNOWN - no info ...\n");
    exit(3);
?>
```

+ Python:
```
#!/usr/bin/python                                    # OK/0
use sys
print "DISK OK - free space: ..."
sys.exit(0)
```

+ Tcl/Tk:
```
#!/usr/bin/tclsh                                     # WARNING/1
puts "DISK WARNING - free space: ..."
exit 1
```

+ BATCH
```
@ECHO OFF                                            # CRITICAL/2
ECHO DISK CRITICAL - free space: ...
EXIT 2
```

+ WSH
```
wscript.echo "DISK UNKNOWN - free space: ..." # UNKNOWN/3
wscript.quit 3                               # Aufruf:
                                             #   cscript FILE //nologo
```

* Dokumentation zur Plugin-Entwicklung:
  + Nagios plug-in development guidelines
    --> http://nagiosplug.sourceforge.net/developer-guidelines.html

19) Auswahl von Plugins

* Unterscheidung:
  + LOKAL: Auf Nagios-Server ausgeführt
  + REMOTE: Auf Ziel-Host ausgeführt
  + AGENT: Ruft andere Plugins auf (NRPE, NWSC, ...)

* Plugin immer genau passend zum Einsatzzweck aussuchen
  (es gibt ein paar GENERISCHE, z.B. "check_udp", "check_tcp")

* Suche nach passendem Plugin aufwendig (und oft frustrierend 8-()
  Viele superspeziell auf EIN persönliches Problem hin geschrieben
  (z.B. Domänenname und Anmeldedaten fest einprogrammiert)
  --> Will man sie verwenden, muss man sie eigentlich NEU schreiben!

* Teilweise grottenschlechte Qualität, funktionieren nicht vernünftig
  --> Lange suchen oder lieber gleich selber schreiben?
  --> Q: "Da kennt jemand einen, der einen kleinen Bruder hat,
          der auch schon mal die Shell benutzt hat;
          und der schreibt dann Plugins für Nagios,
          welche die Qualität von Nagios insgesamt runterziehen."

* Teilweise stimmt Usage-Meldung/man-Page/Doku nicht mit Fähigkeiten überein
  --> Quellcode lesen

* Man MUSS programmieren können (Perl, Shell, PHP, ...)
  + Kombination Shell + UNIX-Kommandozeilentools einfach + leistungsfähig

* SNMP
  + SNMP-fähige Geräte oft schlecht dokumentiert
    --> MIB-OIDs und ihre Bedeutung zuerst ermitteln
  + SNMP-Implementierung oft fehlerhaft oder nicht vollständig
  + SNMP oft nicht vom Hersteller supported (Checkpoint-Firewalls!)
    --> Viel Ausprobieren nötig
    --> Update der Geräte-SW --> Alles wieder anders
  + Abfrage + Antwort oft UNverschlüsselt

* Datenausgabe von Geräten oft unstrukturiert und zu gross
  Relevante Daten im "Sumpf" zu suchen aufwendig
  --> Problem der Plugins, nicht von Nagios.
      Q: "Nagios kann nur soviel wie die zugrunde liegende Soft/Hardware"

* Man muss einen VERLÄSSLICHEN Mechanismus (auf der Kommandozeile) haben,
  um mit Geräten (z.B. Switch, Thermofühler) zu reden. Hat man den ---
  völlig unabhängig von Nagios --- kann man sie auch mit Nagios überwachen.

* Sind auf der Kommandozeile keine Daten von einem Gerät erhältlich,
  dann ist es mit Nagios nicht überwachbar
  --> Geräte mit wunderschöner Java/Flash-Weboberfläche (aber "geheimer" API)
      evtl. mit Nagios nicht überwachbar (IBM Raid --> GUI IBM Director nötig)

* Wenn möglich NICHTS auf Endgeräten installieren (Aufwand vermeiden)
  --> Oft wird "snmpd" unter Linux "vergewaltigt" und zum Mittler gemacht,
      da er Prozesse überwachen (vorhanden, Anzahl),
      beliebige Skripte aufrufen und
      Ergebnis via SNMP zurückgeben kann (z.B. "raidcheck"-Skript)

20) Standard-Plugins

* Manche sind symbolische Links auf anderes Plugin (nur anderer Plugin-Name)

* Liste: ls /usr/local/nagios/libexec/*   # Aus Quellcode installiert
         ls /usr/lib/nagios/plugins/*   # Als Paket installiert

Plugin check_	Beschreibung
apt	Ausstehende Updates von Debian/Ubuntu-Systemen

bgpstate	...
breeze	Signalstärke Breezcom Wireless Gerät per SNMP
by_ssh	Plugin remote per SSH ausführen (kapseln)
clamd	CLAMD-Verbindungen (Link --> check_tcp)
cluster	Erg. mehrerer Checks komb. zu OK/WARNING/CRITICAL (per On-Demand Makros)
dbi	Beliebige Datenbank abfragen (libdbi)
dhcp	DHCP-Server prüfen
dig	DNS-Server per "dig" prüfen
dir_age	Alter der ältesten Datei in Verz. prüfen
disk	Freie Festplattenkapazität
disk_smb	Samba-Share abfragen
dns	DNS-Nameserver prüfen
dummy	Test mit festem Returncode
file_age	Änderungsdatum einer Datei prüfen
flexlm	FLEXlm Lizenserver prüfen (Macrovision)
fping	Schneller als "check_ping" --> check_icmp besser)
ftp	FTP-Server prüfen (Link --> check_tcp)
host	Existenz, nicht Antwortzeit (Link --> check_icmp)
hpasm	Hardware von HP Proliant Server prüfen (Consol)
hpjd	HP Drucker mit JetDirect Karte prüfen
http	Web-Server prüfen
icmp	Per Ping erreichbar (schnelles check_fping/ping)
ide_smart	Lokale Platte per SMART prüfen
ifoperstatus	NW-Interface Status prüfen
ifstatus	NW-Interface Status prüfen
iftraffic	NW-Interface Status+Traffic prüfen (per SNMP)
imap	Mail-Server prüfen (Link --> check_tcp)
ircd	IRC/Chat-Daemon prüfen (Anz. Verbindungen)
jabber	Jabber-Daemon prüf (Anz.Verb., Link --> check_tcp)
ldap	LDAP-Verz.dienst prüfen
ldaps	LDAP-Verz.d. überw. (SSL/TLS, Link --> check_ldap)
linux_raid	Linux Multi Disk Geräte prüfen
load	Linux Systemlast prüfen (load)
log	Logfile auf Muster prüfen
logfile	Logfile auf Muster prüfen
mailq	Mail-Queue prüfen
memory	Freien Speicher prüfen
mrtg	MRTG-Logdatei prüfen
mrtgtraf	MRTG-Logdatei prüfen (Traffic)
mysql	Datenbank MySQL prüfen
mysql_query	Datenbank MySQL prüfen ...
nagios	Nagios prüft sich selbst
netsnmp	Monitor Disk, Swap, Load, Proc + Exec, Perl)
nntp	NNTP-Verb. prüfen (Link --> check_tcp)
nntps	NNTP-Verb. per SSL prüfen (Link --> check_tcp)
nrpe	Remote Plugin per NRPE ausführen
nt	Remote Plugin per NSClient ausführen
ntp	Systemzeit via NTP überprüfen (alt)
ntp_peer	Systemzeit via NTP überprüfen (neu)
ntp_time	Systemzeit via NTP überprüfen (neu)
nwstat	Novell MRTGEXT NML prüfen
oracle	Datenbank ORACLE prüfen
overcr	Over-CR Daemon prüfen
pgsql	Datenbank PostgreSQL prüfen
ping	Erreichbarkeitstest mit Ping
pop	Mail-Server POP-Verb. prüfen (Link --> check_tcp)
postgres	Datenbank PostgreSQL prüfen
procs	Prozesse prüfen
radius	RADIUS-Server prüfen
radius_ih	RADIUS-Server prüfen
raid	...
real	RTPS-Dienst (Real-Time Streaming Protocol) prüfen

rpc	Remote Procedure Call prüfen
rtamulti	... (Link --> check_icmp)
sensors	Status von Geräten im "lm_sensers" Paket prüfen
simap	SIMAP Daemon prüfen (Link --> check_tcp)
smart	SMART-Zugriff auf Festplatten-Daten (RAID)
smtp	Mail-Server SMTP-Verb. prüfen
snmp	Per SNMP prüfen (per snmpget)
spop	Mail-Server prüfen (Link --> check_tcp)
squid	Squid-Proxy prüfen
ssh	SSH prüfen
ssmtp	SSMTP Daemon prüfen (Link --> check_tcp)
swap	Auslastung des Swap-Bereichs prüfen
tcp	TCP-Ports prüfen (generisch)
time	Uhrzeit prüfen (NTP)
time_skew	Uhrzeit prüfen (Abweichung, kein NTP)
udp	UDP-Ports prüfen (gen., Link --> check_tcp)
ups	UPS-Dienst prüfen
users	Anzahl eingeloggter Benutzer prüfen
wave	...

* Hilfs-Plugins

Plugin	Beschreibung
urlize	Plugin-Text-Output --> <A...>...</A> --> Clickable HTML-Link
negate	Status eines Plugins negieren (OK <-> CRITICAL)
utils.sh	Definitionen für Plugins auf Basis von Shell-Skripten
utils.pm	Perl-Modul (veraltet/deprecated), "Nagios::Plugin" nutzen!

Hinweis: Inzwischen wird statt dem Perl-Modul "utils.pm" das Perl-Modul "Nagios::Plugin" eingesetzt (von Cevin Gar).

* Zusätzliche Plugins (--> 33.5 Quellen für Nagios-Plugins)

Plugin check_	Beschreibung
generic	Beliebige UNIX-Befehle nutzen
db2_health	Datenbank DB2 prüfen (ConSol)
mssql_health	Datenbank MSSQL prüfen (ConSol)
mysql_health	Datenbank MySQL prüfen (Consol)
oracle_health	Datenbank ORACLE prüfen (ConSol)
logfiles	Logfile/WinEventLog auf Muster prüfen (ConSol)
nagios_performance	Nagios-Performanz prüfen
many	Mehrere Checks kombinieren (all, first-ok/fail)
multi	Mehrere Checks logisch verknüpfen
mk	Ersatz für NRPE (von Mathias Kettner)
nwc_health	Netzwerkkomponenten per SNMP prüfen (Consol)
junos	Juniper
naf	NetApp Filer
aaccli	Adaptec SCSI Raid Controller (IBM ServerRAID)

* Konfigurations-Dateien für Plugins
  + Definieren INTERNE Kommandos für Aufruf EXTERNER Plugins
  + Verz.: /etc/nagios-plugins/config/*
           /usr/local/nagios3/plugins/config/*
  + Liste:

Konfigurations -Datei "*.cfg"	Plugin (externes Kommando)
apt	check_apt
breeze	check_breeze

check_nrpe	check_nrpe
dhcp	check_dhcp
disk-smb	check_disk_smb
disk	check_disk, check_by_ssh
dns	check_dns, check_dig
dummy	check_dummy
email_delivery	check_email_delivery
flexlm	check_flexlm
fping	check_fping
ftp	check_ftp
games	check_game
haproxy	check_haproxy
hppjd	check_hpjd
http	check_http
httpd_status	check_httpd_status
ifstatus	check_ifstatus, check_ifoperstatus
ipmi_sensor	check_ipmi_sensor
ldap	check_ldap, check_ldaps
lm_sensors	check_lm_sensors
load	check_load
mail	check_pop, check_smpt, check_ssmtp, check_imap
mailq	check_mailq
memcached	check_memcached
mrtg	check_mrtg, check_mrtgtraf
multipath	check_multipath
mysql	check_mysql
netware	check_nwstat
news	check_nntp
nt	check_nt
ntp	check_ntp_peer, check_time
pgsql	check_pgsql
ping	check_ping
printer	check_printer
procs	check_procs
radius	check_radius
rbl	check_rbl
real	check_real
rpc-nfs	check_rpc-nfs
snmp	check_snmp
snmp_cpfw	check_snmp_cpfw.pl
snmp_env	check_snmp_env.pl
snmp_int	check_snmp_int.pl
snmp_load	check_snmp_load.pl
snmp_mem	check_snmp_mem.pl
snmp_process	check_snmp_process.pl
snmp_storage	check_snmp_storage.pl
snmp_vrrp	check_snmp_vrrp.pl
snmp_win	check_snmp_win.pl
ssh	check_ssh
ssl_cert	check_ssl_cert
tcp_udp	check_tcp, check_udp
telnet	check_tcp
users	check_users

21) Nagios Objekte und ihre Beziehungen

* Abstrakte OBJEKTE bilden (zu überwachende) Realität ab
  + Jedes bekommt frei wählbaren INTERNEN Namen --> BEZEICHNER (analog Prog.)

    --> BAUKASTEN basierend auf BEZEICHNERN
    --> Frei wählbar im Rahmen von Nagios (hat nichts mit echter Außenwelt zu tun)
    --> GUTE NAMEN SIND EXTREM WICHTIG!
    --> Für Darstellung in GUI verwendet
  + Jedes hat bestimmte ATTRIBUTE (Name + Wert = Eigenschaft)
  + 12 Typen (bzw. 14 vor V3) + Templates
    --> Vererbung von gemeinsamen Eigenschaften von Objekten

1) Host (genau eine IP = DNS-HOSTNAME)
  + Eindeutiger Name = Host-Name (bzw. IP)
  + HW-Gerät im Netz
  + Host-Check per "ping" (ICMP)
  + Für JEDES zu überwachende Gerät ist ein Host-Objekt nötig!

2) Hostgroup
  + Zusammenfassung von Hosts zu "KATEGORIEN" (z.B. Linux, Windows, Switch)
  + Zur Darstellung in GUI, Auswertung, Reaktion
    ("Kosmetik" + Vereinfachung)
  + Zuordnung von Services, Kommandos, Kontakten, ... --> Vereinfachung der
Konfiguration
  + Host darf Mitglied in MEHREREN Hostgruppen sein
  + GUT ÜBERLEGEN!

3) Service (Ressource/Dienst/Daemon, beliebig viele pro Host)
  + Eindeutiger Name = Host-Name + Service-Description (NICHT Service-Name!)
  + Netzwerk-Dienste
    - Port + Protokoll
  + Daemon
    - Name
  + Lokale/Remote Ressourcen (Betriebsparameter)
    - Belegter/Freier Festplattenplatz
    - Neue Updates für Distribution
    - Wann läuft SSL-Zertifikat ab?
    - Anzahl Prozesse
    - ...

4) Servicegroup
  + Zusammenfassung von Services zu "KATEGORIEN" (z.B. Mail, DB, Web, SAP, ...)
  + Zur Darstellung in GUI, Auswertung, Reaktion
    ("Kosmetik" + Vereinfachung)
  + Zuordnung von Kommandos, Kontakten, ... --> Vereinfachung der Konfiguration
  + Service darf Mitglied in MEHREREN Servicegruppen sein
  + GUT ÜBERLEGEN!

5) Contact (Kontaktperson, Verantwortlicher)
  + Benachrichtigungen (Notification)
  + Sichtbarkeit in GUI (!)
  + Nachrichten-Zeitraum
  + Mail, SMS, ...

6) Contactgroup (Gruppe von Verantwortlichen)
  + Zusammenfassugn von Contacts --> Vereinfachung der Konfiguration
  + Benachrichtigungen
  + Sichtbarkeit in GUI (!)
  + Nachrichten-Zeitraum
  + Mail, SMS, ...
  + GUT ÜBERLEGEN!

7) Timeperiod (Zeitraum/Zeitfenster)
  + Überwachungszeitraum
    - Wartungszeitraum (Service Interval)
    - Problembehandlung
  + Benachrichtigungszeitraum (Notification Interval)

8) Command (Plugin-Aufruf)
  + Typen
    - Host/Service-Check mit Parametern (Statusprüfung)
    - Host/Service Notification (Benachrichtigung, Notify)
    - Host/Service Event-Handler (Ereignisbehandlung bei Statusänderung)
  + Interner Name + Externes Kommando (Pfad + Name)
  + Können Makros enthalten (Platzhalter,
    manuell beim Aufruf oder automatisch vom Host/Service/Contact/... gefüllt)
  + Verbindung: Nagios "Attribute" (Objekt Definition) --> Plugin Parameter
  + Interner + externer Name oft gleich (verwirrend oder auch nicht!)

9) Host Dependency (logische Abhängigkeit)

  + Abhängigkeit zw. Hosts OHNE Bezug zur Netzwerk-"Topologie"
  + Business-Objekte

10) Service Dependency (logische Abhängigkeit)
  + Abhängigkeit zw. Services OHNE Bezug zur Netzwerk-"Topologie"
  + Business-Objekte

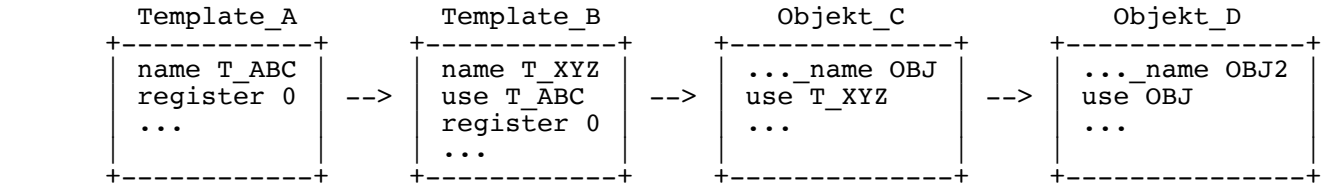
11) Host Eskalation (Eskalationskette)
  + Falls Kontakt(gruppe) nicht reagiert
  + Weitere Benachrichtigungen zu einem Problem anders behandeln
  + Mehrstufig

12) Service Eskalation (Eskalationskette)
  + Falls Kontakt(gruppe) nicht reagiert
  + Weitere Benachrichtigungen zu einem Problem anders behandeln
  + Mehrstufig

13) Extended Host Information (vor V3)
  + Aussehen von Status, Statusmap, Statuswml, Statuswrl und Extinfo festlegen
  + Inzwischen Teil von --> 1) Host Definition (V3)

14) Extended Service Information (vor V3)
  + Aussehen von Status und Extinfo festlegen
  + Inzwischen Teil von --> 3) Service Definition (V3)

15) Template (Vorlage, Schablone)
  + Generisches Attribut "name" legt Template-Namen fest (nicht host_name, ...)
  + UNVOLLSTÄNDIG definiertes Objekt (seit V3 auch VOLLSTÄNDIGE Definition erlaubt)
  + Nicht in GUI angezeigt (sinnvollerweise!)
  + Sogar fertige Objekte als Template verwendbar
      register 0   # In Objekt weglassen
      use OBJEKTNAME   # In davon abgeleitetem Objekt verwenden
  + Zweck
    - Ableiten von Objekten EINHEITLICHER STRUKTUR
    - Mehrfache + hierarchische VERERBUNG
    - Arbeitsersparnis
    - Redundanz vermeiden (änderungsfreundlicher)
    - Konsistenz erhöhen (fehlerfreier)
  + Beispiel: Objekt_D aus Objekt_C abgeleitet,
              Objekt_C aus Template_B abgeleitet,
              TEMPLATE_B aus Template_A abgeleitet,

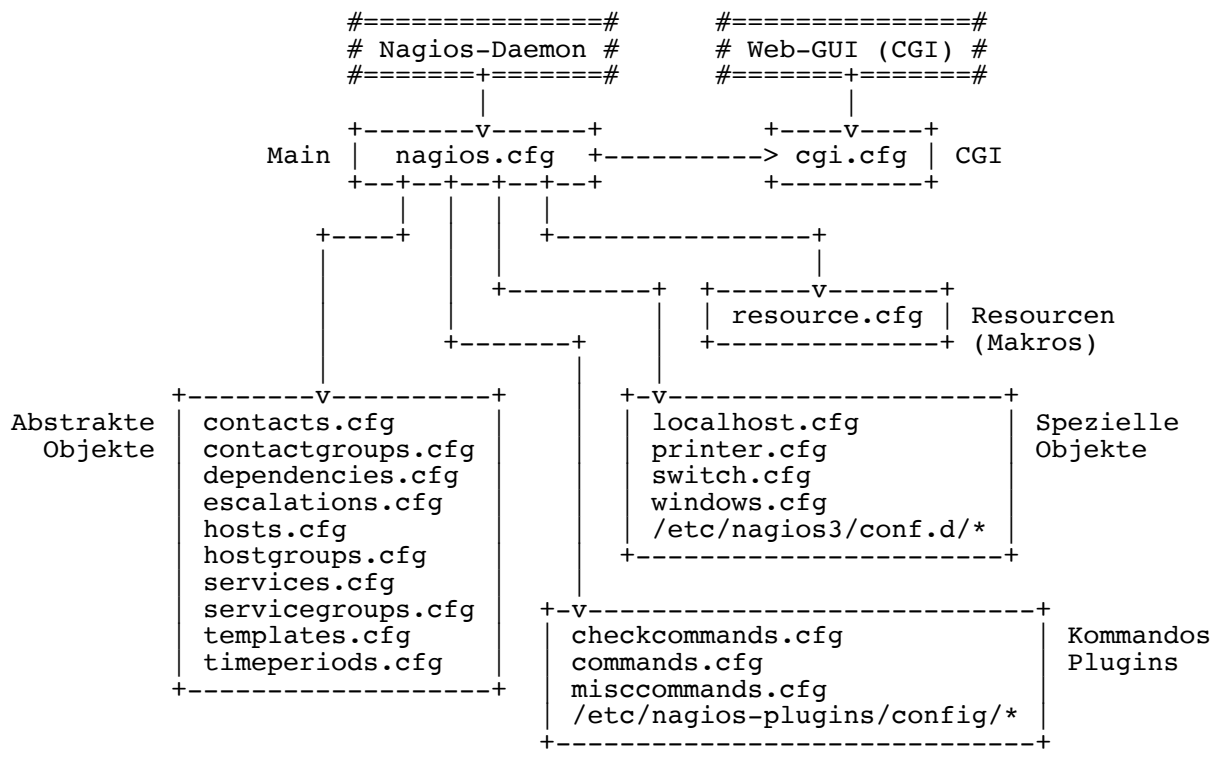


* Einschränkungen gegenüber OOP:
  + KEINE neuen Objektarten möglich
  + KEINE neuen Attribute in Objektarten möglich
    - In V3 doch möglich, aber für Nagios nicht relevant
  --> keine echte Objektorientierung!

22) Konfigurations-Dateien -- Struktur
----

* Die AUFTEILUNG der Konfiguration auf Dateien und Verz. ist frei wählbar,
  ebenso sind die NAMEN der Konfigurations-Dateien/Verz. beliebig wählbar
  (haben nichts mit den Namen von Objekten zu tun)
  + Extremfälle:
    - EINE einzige Konfigurations-Datei enthält ALLES
    - Pro Objekt eine getrennte Konfigurations-Datei
  + STARTPUNKT der Konfiguration ist "nagios.cfg"
    - Enthält globale Nagios-Konfiguration (nur dort erlaubt!)
    - Legt fest, welche weiteren Dateien + Verz. einzulesen sind (include)
        cfg_file = ...   # Einzelne Datei einlesen
        cfg_dir = ...   # Verz.baum einlesen (nur Dateien mit Extension "*.cfg")
  + Grundkonfiguration wird mitgeliefert (Struktur siehe unten)
    - nagios.cfg   # Globale Parameter + Auswahl weitere Konf.dateien+verz.
    - cgi.cfg   # GUI-Konfiguration
    - resource.cfg   # Fixe Makros für Pfade, Benutzernamen, Passworte, ...
    - ...   # Weitere Dateien mit Objekten

```
  + Mitgelieferte Beispielkonfiguration
    - Startet zunächst nicht
    - Mind. 1 Host + 1 Service + 1 Contact definieren
      . Die ersten Checks lokal durchführen (auf Nagios-Server)
        . "localhost"            erreichbar?
        . "lokale NW-Verbindung" erreichbar?
  + Gesunder Mittelweg:
    - Mitgelieferte Grundstruktur als Ausgangspunkt verwenden
    - Eigene Objekte in getrenntem neuen Verz. ablegen
    - Pro Objekttyp oder zusammengehörenden Objekten eine Konf.datei
    - Einheitliche Namensgebung für Dateien, Verz. und Objekte:
      . Dateien: Namens-Strukturierung per Unterstrich "_"
      . Objekte: Namens-Strukturierung per Bindestrich "-"
  + NUR Dateien mit Endung ".cfg" (configuration) sind für Nagios relevant!
    --> TIPP: Umbenennen nach ".cfg.old" deaktiviert sie!

* Startpunkt der Konfigurations-Dateien liegt unter:
  + /usr/local/nagios/etc   # Aus Quellcode installiert
  + /etc/nagios             # V2
  + /etc/nagios3            # V3
  + /etc/nagios4            # V4
  + /etc/pnp4nagios         # PNP4Nagios
  + /etc/nagios-plugins     # Plugins
  + /etc/nagiosgrapher      # NagiosGrapher

* Zusammenhang der Konfigurations-Dateien (Grundkonfiguration):
```



Datei "*.cfg"	Bedeutung
nagios	Haupt-Konfiguration (Server-Optionen + Includes)
resource	Benutzerdef. Makros (z.B. Pfade, Passworte)
cgi	Web-GUI Konfiguration
objects/...	Weitere Objekte (Kommandos, Server, ...)
conf.d/...	Weitere Objekte (Kommandos, Server, ...)
templates	Templates (Vorlagen) für Hosts + Services
hosts	Überwachte Systeme (IP!)
hostgroups	Zsfg. von Systemen zu Gruppen
services	Überwachte Dienste und Betriebsparameter
servicegroups	Zsfg. von Services zu Gruppen
dependencies	Logische Abhängigkeiten zw. Hosts/Services
commands	Kommandos für Tests + Benachrichtigungen
checkcommands	Checks mit Aufrufparametern der Plugins
misccommands	Kommandos (z.B. für E-Mail-Benachrichtigung)

timeperiods	Definition von Zeitfenstern (z.B. "workhours")
contacts	Kontaktpersonen
contactgroups	Zsfg. von Kontaktpersonen zu Gruppen
escalations	Eskalationen bei anhaltenden Fehlern
localhost	Checks für Localhost
printer	Checks für Drucker
switch	Checks für Switches/Hubs/Router
windows	Checks für Hosts+Services von Windows-Rechnern

23) Konfigurations-Dateien -- Inhalt

* Konfigurations-Dateien
  + Dateiname unwichtig, Extension ".cfg" wichtig (NUR diese werden eingelesen)
  + Pfadangaben verwenden "/" (Linux-Form), NICHT "\" (Windows-Form)
    (oder "\" verdoppeln --> "\\")
  + Wo Dateien liegen entscheidet Installation (Quellen vs. Distro-Paket)

* Nagios verzeiht keine Syntaxfehler --> IMMER PRÜFEN mit Option "-v" (verify)!
  + Prüft aber auch einige Dinge nicht ab
    - Externe Pfade existent?
    - Zugriffsrechte vorhanden?
    - Plugin-Optionen + Parameter sinnvoll?
    - Plugin-Aufruf möglich
    - Makro definiert
    - Macro mit $...$ umrahmt
    - ...

* ACHTUNG: Egal wie viele Dateien/Objekten/Templates vorhanden sind,
  bei Änderung eines einzigen Details (z.B. Name, Attributwert, Abhängigkeit)
  muss wieder die GESAMTE KONFIGURATION verifiziert und eingelesen werden!
  --> Idee bei Icinga 2: Auch einzelne Objekte einzulesen und zu aktivieren,
                         Rest bleibt aktiv
  Kann durchaus einige Sekunden/Minuten dauern (abh. von Anz. Hosts/Services)
  bis komplett eingelesen, geprüft und alle Abhängigkeiten ermittelt!

* Objekte
  + Fast alles dreht sich um "OBJEKTE" aus einer Liste von OBJEKT_TYPEN
    (Hosts, Services, Commands, ...)
  + Objekte müssen EINDEUTIGEN NAMEN = BEZEICHNER haben
  + Alle Namen sind CASE-SENSITIVE (d.h. GROSS/kleinschreibung wird beachtet)
    - Außer bei Objekt-Hostnamen (da DNS-Hostnamen das auch nicht machen)!
  + Irrelevant:
    - AUFTEILUNG der Objekte auf Definitionsdateien
    - REIHENFOLGE der Objekte in Definitionsdateien
  + Extremfälle: Alle Objekte in einer Datei <-> Pro Objekt eine Datei

* Format der Konfigurations-Dateien
  + Zentrale Konfigurations-Dateien: nagios.cfg + cgi.cfg
    - Liste von Variable-Wert-Paaren der Form
      VARIABLE = WERT
  + Globale Makros: resource.cfg
    - Liste von Makro-Wert-Paaren der Form (n = 1..32)
      $USERn$ = WERT
  + Objekt-Definition
    - Liste von Objekten bestehend aus

```
define OBJEKT_TYP {   # Objekt-Typ
  ..._name NAME       # Objekt-Name
  VARIABLE WERT       # Attribut + Leerraum + Wert
  ...                 # (KEIN "=" dazwischen!)
}
```

* Format der Konfigurations-Dateien nur geringfügig geändert zwischen Versionen
  + Umbenennungen oder Zusammenfassungen von Parametern
  + Überflüssige Parameter fallen weg
  + Zusätzliche Möglichkeiten kommen hinzu
  --> Nagios-Upgrade leicht möglich (z.B. von V2 auf V3)

* Sonstige Syntaxelemente
  + Leerzeilen und Leerraum beliebig verwendbar
    - Einrücken erlaubt (und sinnvoll)

```
    - Leerraum um "{" und "}" erlaubt
  + Service-Description und Kommando-Name darf beliebige Zeichen enthalten
    - Vorsicht mit Umlauten "äöüÄÖÜß", Sonderzeichen und Leerzeichen!)
      --> Codierungsproblem: GUI != Konfig != Kommandozeile (Plugins)
  + ";" leitet KOMMENTAR bis Zeilenende ein (extra Zeile oder nach Code)
  + "#" leitet KOMMENTAR bis Zeilenende ein (extra Zeile oder nach Code)
  + Listenelemente per "," OHNE Leerz. trennen (Host, Service, Contact, ...)
  + "*" steht für ALLE (Hosts, Services, ...)
  + "!" trennt ARGUMENTE beim Aufruf eines Plugins (Leerzeichen
  + "$...$" kennzeichnet MAKROS (fehlendes schließendes $ wird nicht erkannt!)
  + Nach "\" am Zeilenende ist Umbruch erlaubt --> wird ignoriert (line continuation)
  + "!" als Text übergeben --> "%3E" verwenden (oder "\!" V3)
  + "$" als Text übergeben --> "$$"  verwenden (oder "\$" V3)

* Beispiel-Konfiguration siehe:
    /usr/share/doc/nagios3-common/examples/
    /usr/local/nagios/etc/

  +---------------------------------+----------------------------+
  | Datei                           | Bedeutung                  |
  +---------------------------------+----------------------------+
  | README                          | Anleitung                  |
  | nagios.cfg                      | Start der Konfiguration    |
  | cgi.cfg                         | GUI-Konfiguration          |
  | resource.cfg                    | Wichtige Makros $USERi$    |
  | apache2.conf                    | V2/V3                      |
  | apache2.nagios1.conf            | V1                         |
  | commands.cfg                    | Kommando-Objekte           |
  | localhost.cfg                   | Nagios-Server selbst       |
  | mrtg.cfg                        |                            |
  +---------------------------------+----------------------------+
  | template-object/README          |                            |
  |         "       commands.cfg    | Templates: Kommandos       |
  |         "       contacts.cfg    |            Kontakte        |
  |         "       localhost.cfg   |            Nagios-Server   |
  |         "       printer.cfg     |            Drucker         |
  |         "       switch.cfg      |                            |
  |         "       templates.cfg   |                            |
  |         "       timeperiods.cfg |            Zeitperioden    |
  |         "       windows.cfg     |            Windows-Rechner |
  +---------------------------------+----------------------------+


23.1) Haupt-Konfigurations-Datei "nagios.cfg"
---------------------------------------------

* Wird als einzige direkt eingelesen (-d = Daemon-Modus)
    /usr/local/nagios/sbin/nagios -d /usr/local/nagios/etc/nagios.cfg
    /usr/sbin/nagios3             -d /etc/nagios3/nagios.cfg

* Enthält generelle Einstellungen

* Bindet weitere Dateien oder Verz. (rekursiv) ein
  (bei "cfg_dir" nur Dateien mit Extension *.cfg!):
    resource_file = /usr/local/nagios/etc/resource.cfg
    cfg_file      = /usr/local/nagios/etc/objects/commands.cfg
    cfg_dir       = /usr/local/nagios/etc/objects

* Beispiel
  # Weitere Konfigurations-Dateien + -verz. (automatisch eingelesen)
  resource_file                 = /etc/nagios3/resource.cfg
  cfg_dir                       = /etc/nagios-plugins/config
  cfg_dir                       = /etc/nagios3/conf.d
  cfg_file                      = /etc/nagios3/commands.cfg

  p1_file                       = /usr/lib/nagios3/p1.pl
  object_cache_file             = /var/cache/nagios3/objects.cache
  precached_object_file         = /var/lib/nagios3/objects.precache

  lock_file                     = /var/run/nagios3/nagios3.pid

  temp_file                     = /var/cache/nagios3/nagios.tmp
  temp_path                     = /tmp

  state_retention_file          = /var/lib/nagios3/retention.dat
  retention_update_interval     = 60
```

```
  status_file                    = /var/cache/nagios3/status.dat
  status_update_interval         = 10

  check_result_path              = /var/lib/nagios3/spool/checkresults
  check_result_reaper_frequency  = 10

  nagios_group                   = nagios
  nagios_user                    = nagios

  check_external_commands        = 0
  command_file                   = /var/lib/nagios3/rw/nagios.cmd
  command_check_interval         = -1
  external_command_buffer_slots  = 4096

  execute_host_checks            = 1
  execute_service_checks         = 1

  accept_passive_host_checks     = 1
  accept_passive_service_checks  = 1

  enable_notifications           = 1
  enable_event_handlers          = 1

  sleep_time                     = 0.25
  service_check_timeout          = 60
  host_check_timeout             = 30
  event_handler_timeout          = 30
  notification_timeout           = 30
  perfdata_timeout               =  5

  check_interval                 = 2.5   # (alt: "normal_check_interval")
  interval_length                = 60    # Sekunden

  process_performance_data       = 0                          # Perf.daten verarbeiten
  service_perfdata_command       = process-service-perfdata   # Kmdo zur Verarb. von
Perf.d.
  host_perfdata_command          = process-host-perfdata      # Kmdo zur Verarb. von
Perf.d.

  translate_passive_host_checks = 0
  passive_host_checks_are_soft  = 0
  check_for_orphaned_services   = 1
  check_for_orphaned_hosts      = 1

  enable_flap_detection          =    1
  low_service_flap_threshold     =  5.0
  high_service_flap_threshold    = 20.0
  low_host_flap_threshold        =  5.0
  high_host_flap_threshold       = 20.0

  date_format                    = iso8601

  # Embedded Perl Interpreter
  p1_file                        = /usr/lib/nagios3/p1.pl
  enable_embedded_perl           = 1
  use_embedded_perl_implicitly   = 1

  illegal_object_name_chars      = `~!$%^&*|'"<>?,()=
  illegal_macro_output_chars     = `~$&|'"<>

  admin_email                    = root@localhost
  admin_pager                    = pageroot@localhost

  max_check_result_file_age      = 3600
  max_check_result_reaper_time   = 30
  max_concurrent_checks          = 0
  max_host_check_spread          = 30
  max_service_check_spread       = 30

  enable_environment_macros                   = 1
  enable_predictive_host_dependency_checks    = 1
  enable_predictive_service_dependency_checks = 1

  # Kommandos definieren, die nach JEDEM Host Check durchgefuehrt werden
  # (nach einem Event-Handler oder Notification-Command)
```

```
  obsess_over_services          = 0        # Global für alle Services an/aus (Plural!)
  obsess_over_hosts             = 0        # Global für alle Hosts an/aus (Plural!)
  obsess_over_service / obsess  = 0        # Bei einzelnem Service an/aus (Singular!)
  obsess_over_host / obsess     = 0        # Bei einzelnem Host an/aus (Singular!)
  ochp_command                  = CMD      # Status-Meldungen weiter an Dest-Nagios
  ocsp_command                  = CMD      # Status-Meldungen weiter an Dest-Nagios
  ochp_timeout                  = 5
  ocsp_timeout                  = 5

  use_regexp_matching           = 0
  use_true_regexp_matching      = 0

  auto_reschedule_checks        = 0
  auto_rescheduling_interval    = 30
  auto_rescheduling_window      = 180

  retain_state_information      = 1
  use_retained_program_state    = 1
  use_retained_scheduling_info  = 1

  retained_contact_host_attribute_mask    = 0
  retained_contact_service_attribute_mask = 0
  retained_host_attribute_mask            = 0
  retained_process_host_attribute_mask    = 0
  retained_process_service_attribute_mask = 0
  retained_service_attribute_mask         = 0

  check_host_freshness             = 0
  check_service_freshness          = 1
  host_freshness_check_interval    = 60
  service_freshness_check_interval = 60
  additional_freshness_latency     = 15

  host_inter_check_delay_method    = s
  service_inter_check_delay_method = s

  cached_host_check_horizon     = 15
  cached_service_check_horizon  = 15

  daemon_dumps_core             = 0
  event_broker_options          = -1
  service_interleave_factor     = s
  soft_state_dependencies       = 0
  use_aggressive_host_checking  = 0
  use_large_installation_tweaks = 0
  enable_environment_macros     = 0
```

23.2) Resource-Konfigurations-Datei "resource.cfg"

* Nicht für alle lesbar, nur für Gruppe "nagios" (wg. Passworten)

* Benutzerdefinierte Makros für
  + Pfadangaben   = Einmal definiert, häufig verwendet
  + Benutzernamen = Sicherheitrelevante Infos
  + Passworte   = Sicherheitrelevante Infos

* Max. 32 Variablen $USER1$ .. $USER32$

* Beispiel

```
  +-------------------------------------------------+
  | $USER1$ = /usr/lib/nagios/plugins               | # Pfad Check Plugins
  | $USER2$ = /usr/lib/nagios/plugins/eventhandlers | # Pfad Event-Handler Plugins
  | $USER3$ = user                                  | # Benutzername
  | $USER4$ = geheim                                | # Passwort
  | ...                                             | # ...
  | $USER32$ = ...                                  | # ...
  +-------------------------------------------------+
```

23.3) GUI-Konfigurations-Datei "cgi.cfg"

* Enthält Einstellungen für GUI

* Muss für Webserver "apache2" lesbar sein (Benutzer/Gruppe "www-data")

* Beispiel

```
  main_config_file                        = /etc/nagios3/nagios.cfg
  physical_html_path                      = /usr/share/nagios3/htdocs
  url_html_path                           = /nagios3

  show_context_help                       = 1
  use_pending_states                      = 1

  nagios_check_command                    = /usr/lib/nagios/plugins/check_nagios \
                                            /var/cache/nagios3/status.dat 5 \
                                            '/usr/sbin/nagios3'

  use_authentication                      = 1
  use_ssl_authentication                  = 0
  authorized_for_system_information       = nagiosadmin,root
  authorized_for_configuration_information = nagiosadmin,root
  authorized_for_system_commands          = nagiosadmin,root
  authorized_for_all_services             = nagiosadmin,root
  authorized_for_all_hosts                = nagiosadmin,root
  authorized_for_all_service_commands     = nagiosadmin,root
  authorized_for_all_host_commands        = nagiosadmin,root
  authorized_for_read_only                = user1,user2,...
  default_user_name                       = guest

  statusmap_background_image              = smbackground.gd2

  color_transparency_index_r              = 255
  color_transparency_index_g              = 255
  color_transparency_index_b              = 255

  default_statusmap_layout                = 5
  default_statuswrl_layout                = 4
  statuswrl_include                       = myworld.wrl

  ping_syntax                             = /bin/ping -n -U -c 5 $HOSTADDRESS$
  refresh_rate                            = 90
  escape_html_tags                        = 1

  host_unreachable_sound                  = hostdown.wav
  host_down_sound                         = hostdown.wav
  service_critical_sound                  = critical.wav
  service_warning_sound                   = warning.wav
  service_unknown_sound                   = warning.wav
  normal_sound                            = noproblem.wav

  action_url_target                       = _blank
  notes_url_target                        = _blank

  lock_author_names                       = 1

  enable_splunk_integration               = 1
  splunk_url                              = http://127.0.0.1:8000/
```

23.4) Objekt-Konfigurations-Dateien

* Objekt-Typen

Objekt-Typ	Schlüsselwort	Bedeutung
Host	host	Rechner (DNS-Name/IP-Adresse)
Host Group	hostgroup	Rechner-Gruppe (Hosts-Liste)
Service	service	Dienst (Port) oder Ressource
Service Group	servicegroup	Dienst-Gruppe (Services-Liste)
Contact	contact	Kontakt
Contact Group	contactgroup	Kontakt-Gruppe (Kontakte-Liste)
Timeperiod	timeperiod	Zeitraum
Command	command	Kommando (intern --> extern)
Host Dependency	hostdependency	Rechner-Abhängigkeit (logisch)
Service Dependency	servicedependency	Service-Abhängigkeit (logisch)
Host Escalation	hostescalation	Weiterreichen Rechner-Meldung

Service Escalation	serviceescalation	Weiterreichen Dienst-Meldung
Extended Host Info	hostextinfo	Zusätzliche Rechner-Info (V3: Teil von "host", V4: -)
Extended Service Info	serviceextinfo	Zusätzliche Dienst-Info (V3: Teil v. "service", V4: -)

* Objektdefinitions-Dateien "conf.d/*.cfg"
  + Alle zu überwachenden Dinge
    - Hosts + Hostgroups
    - Services + Servicegroups
  + Wie sie zu überwachen sind
    - Contacts + Contactgroups
    - Timeperiods
    - Commands (Check + Notification + Event-Handler)
    - Dependencies + Escalations
    - Host/Service Extended Information (V3: Teil von Host/Service)

* Mögliche Strukturierung der Aufteilung von Objekten auf Dateien
  *) Beliebig (kein inhaltlicher Bezug zw. Namen von Objekten und Dateien)
  A) Alle Objekte in einer Datei
  B) Pro Objekt eine Datei
  C) Objekttypen auf Konfigurations-Dateien verteilen
  D) Konfiguration nach logischen Gesichtspunkten in Unterverz. strukturiert
     + Globale Einstellungen
     + Standorte
     + Kunden
     + Abteilungen
     + Gerätetypen

* Wenn möglich von TEMPLATE (Vorlage) Objekt-Eigenschaften "erben"
  + Template = UNVOLLSTÄNDIG definiertes Objekt
    - Mit "name TPL_NAME" und Attribut "register 0"
  + Vererben bzw. ableiten von Objekt aus Template
    - Direktive "use TPL_NAME"
    - Parameter hinzufügen
    - Parameter überschreiben
    - Parameter erweitern
  + Mehrfachvererbung + Vererbungshierarchie erlaubt (V3)
    - Template aus Template ableitbar
    - Objekt aus mehreren Templates ableitbar

* Rechner-Objekt (vollständige Definition, *=MUSS, **=Objekt, */** nicht Syntax-Teil)
  + Kommando-Angabe ohne Parameter

```
    define host {
       *host_name              charlton         # evtl. identisch zum DNS-Namen
        hostgroups             linux            # --> OBJEKT!
        alias                  Sony Notebook    # Beschreibungstext
       *address                192.168.0.1      # oder DNS-Name (optional ab V4)
        check_command          check_host       # --> OBJECT! = Kmdo-Aufruf!
       *max_check_attempts     5                # SOFT --> HARD
       *check_period           24x7             # --> OBJEKT!
      **contact_groups         admins           # --> OBJEKT!
      **contacts               admin            # --> OBJEKT!
       *notification_interval  240              # alle 4h wiederholen falls unverändert
       *notification_period    24x7             #
       *notification_options   d,r              #
        parents                switch           # --> OBJEKT!
    }
```

* Rechner-Template (partielle Definition, *=MUSS)
  + Kommando-Angabe ohne Parameter

```
    define host {
       *name                   template-host    # Template-Name
       *register               0                # Template
        check_command          check_host       # --> OBJEKT! = Kmdo-Aufruf!
        max_check_attempts     5                #
        check_period           24x7             # --> OBJEKT!
        contact_groups         admins           # --> OBJEKT!
        contacts               admin            # --> OBJEKT!
        notification_interval  240              #
        notification_period    24x7             #
        notification_options   d,r              #
        hostgroups             xyz              # Std-Gruppe (ergänzt/ersetzt/behalten)
    }
```

```
* Rechner-Objekt (Template + Ergänzungen --> vollständige Definition, *=MUSS)
    define host {
        use         template-host     # Ableitung/Vererbung von Template
       *host_name   charlton          #
        hostgroups  +linux            # --> OBJEKT! (hinzufügen)
        alias       Sony Notebook     #
       *address     192.168.0.1       # Attributwert --> Platzhalter $HOSTADDRESS$
        parents     switch            # --> OBJEKT!
    }

    define host {
        use         template-host     # Ableitung/Vererbung von Template
       *host_name   exeter            #
        hostgroups  linux             # --> OBJEKT! (überschreiben)
        alias       HP Notebook       #
       *address     192.168.0.2       #
        parents     switch            # --> OBJEKT!
    }

    define host {                     # Hostgruppe "xyz" wird aus Template 1:1
übernommen
        use         template-host     # Ableitung/Vererbung von Template
       *host_name   plymouth          #
        alias       HP Notebook       #
       *address     192.168.0.2       #
        parents     switch            # --> OBJEKT!
    }

* Hostgroup-Objekt (*=MUSS)
    define hostgroup {
       *hostgroup_name  linux             #
       *alias           Linux-Rechner     #
        members         charlton          # --> OBJEKT!
    }
    define hostgroup {
       *hostgroup_name  routers           #
       *alias           Router/Switches   #
    }

* Service-Objekt (vollständige Definition, *=MUSS)
  + Kommando-Angabe mit Parametern
    define service {
       *host_name              charlton          # --> OBJEKT!
       *service_description    PING              #
       *check_command          check_icmp!100.0,20%!500.0,60%    #Objekt --> Kmdo-
Aufruf
       *max_check_attempts     3                 # SOFT-->HARD
       *check_interval         5                 #
       *retry_interval         1                 #
       *check_period           24x7              # --> OBJEKT!
       *notification_interval  240               #
       *notification_period    24x7              # --> OBJEKT!
       *notification_options   c,r               #
      **contact_groups         admins            # --> OBJEKT!
      **contacts               admin             # --> OBJEKT!
    }

* Service-Template (partielle Definition, *=MUSS)
    define service {
       *name                   template-service  #
       *register               0                 # Template
        max_check_attempts     3                 #
        check_interval         5                 #
        retry_interval         1                 #
        check_period           24x7              # --> OBJEKT!
        notification_interval  240               #
        notification_period    24x7              # --> OBJEKT!
        notification_options   c,r               #
        contact_groups         admins            # --> OBJEKT!
    }

* Service-Objekt (Template + Ergänzungen --> vollständige Definition, *=MUSS)
  + Kommandoangabe mit Parametern
    define service {
        use                 template-service                # Ableitung/Vererbung
```

```
       *host_name            charlton                          #
       *service_description  PING                              #
       *check_command        check_icmp!100.0,20%!500.0,60%    # Kmdo-Aufruf
    }

* Weitere Service-Objekte (*=MUSS)
    define service {
       *host_name            charlton,switch   # Service ex. auf allen Hosts dieser
Liste
       *service_description  PING              #
       ...                                     #
    }
    define service {
       *hostgroup_name       linux   # Service ex. auf allen Hosts dieser Gruppe
       *service_description  PING    #
       ...                           #
    }
    define service {
       *host_name            *       # Service ex. auf allen Hosts!
       *service_description  PING    #
       ...                           #
    }

* Servicegroup-Objekt (*=MUSS)                    # --> HOST-Beziehung
    define servicegroup {                         #
       *servicegroup_name      remote             #
       *alias                  Remote-Dienste     #
        members                SSH,HTTP,SMTP      # --> OBJEKT!
        servicegroup_members   RPC                # --> OBJEKT!
    }

* Timeperiod-Objekt (*=MUSS)
    define timeperiod {
       *timeperiod_name  24x7          #
        alias            7x24h         #
        sunday           00:00-24:00   #
        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   #
    }

    define timeperiod {
       *timeperiod_name  weekend       #
        alias            Wochende      #
        saturday         00:00-24:00   #
        sunday           00:00-24:00   #
    }

    define timeperiod {
       *timeperiod_name  worktime              #
        alias            Normale Arbeitszeit   #
        monday           09:00-16:30           #
        tuesday          09:00-16:30           #
        wednesday        09:00-16:30           #
        thursday         09:00-16:30           #
        friday           09:00-13:00           #
    }

    define timeperiod {
       *timeperiod_name  none   #
        alias            Nie!   #
    }

    define timeperiod {
        timeperiod_name      us-holidays    #
        alias                U.S. Holidays  #
        january 1            00:00-00:00    # New Years
        monday -1 may        00:00-00:00    # Memorial Day (last Mon in May)
        july 4               00:00-00:00    # Independence Day
        monday 1 september   00:00-00:00    # Labor Day (1st Mon in Sep)
        thursday 4 november  00:00-00:00    # Thanksgiving (4th Thu in Nov)
        december 25          00:00-00:00    # Christmas
    }
```

```
    define timeperiod {
       *timeperiod_name 24x7_sans_holidays  #
        alias           24x7 Sans Holidays  #
        use us-holidays                     # --> OBJEKT als TEMPLATE!
        sunday          00:00-24:00         #
        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         #
    }

* Command-Objekt (für Check, *=MUSS)
  + Plugin-Aufruf mit Parametern
    define command {
       *command_name  check_host                              # Interner Nagios-Name
       *command_line  $USER1$/check_host -H $HOSTADDRESS$     # PLUGIN-Aufruf (ext. Kmdo)
    }
    define command {
       *command_name  check_icmp                                        # Interner Nagios-
Name
       *command_line  $USER1$/check_icmp -H $HOSTADDRESS$\              # PLUGIN-A. (ext.
Kmdo)
                                         -w '$ARG1$' -c $ARG2$   # (mehrzeilig)
                                    #       ^           ^
                                    #       |           |
                                    # !100.0,20% !500.0,60%
    }

* Command-Objekt (für Benachrichtigung, *=MUSS)
  + Plugin-Aufruf mit Parametern
    define command {
       *command_name  notify-by-email                         #
       *command_line  printf "%b" "MELDUNG..." |\              # Plugin-Aufruf
                        mail -s "SUBJECT..." ADDRESS
    }
    define command {
       *command_name  notify-by-sms                           #
       *command_line  /usr/bin/printf "%b" "MELDUNG..." |\    # Plugin-Aufruf
                        smsclient NUMBER
    }

* Contact-Objekt (*=MUSS)
  + Kommandoangabe mit Parametern
    define contact {
       *contact_name                   nagadmin               #
        contactgroups                  admins                 # --> OBJEKT!
       *alias                          Nagios Admin           # --> OBJEKT!
       *host_notification_period       24x7                   # --> OBJEKT!
       *service_notification_period    24x7                   # --> OBJEKT!
       *host_notification_options      d,r                    #
       *service_notification_options   w,c,r                  #
        host_notification_commands     host-notify-...        # --> OBJEKT! --> KMDO
        service_notification_commands  notify-by-email        # --> OBJEKT! --> KMDO
        email                          nagadmin@localhost     #
        pager
        address
    }

* Contactgroup-Objekt (*=MUSS)
    define contactgroup {
       *contactgroup_name     nagadmin             #
       *alias                 Administratoren      #
        members               CONTACT_NAMEs        # --> OBJEKT!
        contactgroup_members  CONTACTGROUP_NAMEs   # --> OBJEKT!
    }

* Host Dependency-Objekt (*=MUSS)
    define hostdependency {
       *dependent_host_name            HOST_NAME         # --> OBJEKT!
        dependent_hostgroup_name       HOSTGROUP_NAME    # --> OBJEKT!
       *host_name                      HOST_NAME         # --> OBJEKT!
        hostgroup_name                 HOSTGROUP_NAME    # --> OBJEKT!
        inherits_parent                [0/1]             #
```

```
        execution_failure_criteria     [o,d,u,p,n]       #
        notification_failure_criteria  [o,d,u,p,n]       # p=PENDING
        dependency_period              TIMEPERIOD_NAME   # --> OBJEKT!
    }

* Service Dependency-Objekt (*=MUSS)
    define servicedependency {
       *dependent_host_name              HOST_NAME             # --> OBJEKT!
        dependent_hostgroup_name         HOSTGROUP_NAME        # --> OBJEKT!
       *dependent_service_description    SERVICE_DESCRIPTION   # --> OBJEKT!
       *host_name                        HOST_NAME             # --> OBJEKT!
        hostgroup_name                   HOSTGROUP_NAME        # --> OBJEKT!
       *service_description              SERVICE_DESCRIPTION   # --> OBJEKT!
        inherits_parent                  [0/1]                 #
        execution_failure_criteria       [o,w,u,c,p,n]         # p=PENDING
        notification_failure_criteria    [o,w,u,c,p,n]         # o=OK
        dependency_period                TIMEPERIOD_NAME       # --> OBJEKT!
    }

* Host Escalation-Objekt (*=MUSS)
    define hostescalation {
       *host_name              HOST_NAME           #
        hostgroup_name         HOSTGROUP_NAME      #
       *contacts               CONTACT_NAMEs       #
       *contact_groups         CONTACTGROUP_NAME   #
       *first_notification     #                   #
       *last_notification      #                   #
       *notification_interval  #                   #
        escalation_period      TIMEPERIOD_NAME     #
        escalation_options     [d,u,r]             #
    }                                              #

* Service Escalation-Objekt (*=MUSS)
    define serviceescalation {
       *host_name              HOST_NAME             #
        hostgroup_name         HOSTGROUP_NAME        #
       *service_description    SERVICE_DESCRIPTION   #
       *contacts               CONTACT_NAMEs         #
       *contact_groups         CONTACTGROUP_NAMEs    #
       *first_notification     #                     #
       *last_notification      #                     #
       *notification_interval  #                     #
        escalation_period      TIMEPERIOD_NAME       #
        escalation_options     [w,u,c,r]             #
    }

* Extended Host Information (*=MUSS, seit V3 direkt beim Host angebbar)
    define hostextinfo {
       *host_name        HOST_NAME                #
        notes            Note_String              #
        notes_url        Url                      #
        action_url       Url                      #
        icon_image       Image_File               #
        icon_image_alt   Alt_String               #
        vrml_image       Image_File               #
        statusmap_image  Image_File               #
        2d_coords        X_Coord,Y_Coord          #
        3d_coords        X_Coord,Y_Coord,Z_Coord  #
    }

* Extended Service Information (*=MUSS, seit V3 direkt beim Service angebbar)
    define serviceextinfo {
       *host_name            HOST_NAME             #
       *service_description  SERVICE_DESCRIPTION   #
        notes                Note_String           #
        notes_url            Url                   #
        action_url           Url                   #
        icon_image           Image_File            #
        icon_image_alt       Alt_String            #
    }


23.5) Vordefinierte Objekte
---------------------------

* Vordefinierte Templates:
```

```
    +-----------------+---------+
    | generic-host    | Host    |
    | linux-server    | Host    |
    | windows-server  | Host    |
    | generic-printer | Host    |
    | generic-switch  | Host    |
    +-----------------+---------+
    | generic-service | Service |
    | local-service   | Service |
    +-----------------+---------+
    | generic-contact | Kontakt |
    +-----------------+---------+
```

* Vordefinierte Kommandos (siehe auch --> 20) Standard-Plugins)

```
    +-------------------------+
    | notify-host-by-email    |
    | notify-service-by-email |
    +-------------------------+
    | check-host-alive        |
    +-------------------------+
    | process-host-perfdata   |
    | process-service-perfdata|
    +-------------------------+
```

24) Benachrichtigungen

* Durch externe Programme durchgeführt (Notification Plugin)

* Einzustellen ist (vielfache Möglichkeiten)
  + WANN generieren?
  + WANN zustellen?
  + WEM zustellen?
  + WIE zustellen?

* Erst nach einem Status-Wechsel zum Typ "HARD" wird durch folgende
  FILTERKETTE bestimmt, ob eine Benachrichtigung verschickt wird:

```
                                                                BEZOGEN AUF
    #=============================================================#
    # Aktuelle Zeit in Zeitraum?                    check_period # Host/Service
    #           |JA               retry_interval                #
    # Prüfung durchzuführen?      check_interval + check_command #     "
    #           |JA                                             #
    # Prüfung ergibt Status-Wechsel?                            #
    #           |JA                                             #     "
    # Perm. HARD-Status-Wechsel?              max_check_attempts #     "
    #           |JA                                             #
    # Benachrichtigungen generell aktiv?    enable_notifications #     "
    #           |JA                                             #
    # Host/Service in geplanter Downtime?  notifications_enabled #     "
    #       NEIN|                                               #
    # Status in festgelegter Statusliste?   notification_options #     "
    #           |JA                                             #
    # Aktuelle Zeit in Benachr.zeitraum?     notification_period #     "
    #           |JA                                             #
    # Schon Nachricht über diesen Status-Wechsel?               #     "
    #           |               |JA                             #
    #       NEIN|  Seitdem abgelaufen ...?  notification_interval #     "
    #           |               |JA                             #
    # Zuständige Kontakte ermitteln!      contacts/contact_groups #     "
    #           |                                               #
    # Status in Statusliste?   host/service_notification_options # Contact
    #           |JA                                             #
    # Aktuelle Zeit in Zeitraum? host/service_notification_period #     "
    #           |JA                                             #
    # Benachrichtigung!        host/service_notification_commands # Command
    #=============================================================#
```

* Einstellungen der Benachrichtigungen:

```
  + Global:       enable_notifications          = 1     # <-+
                  enable_event_handlers         = 1     #   |
                  interval_length               = 60    #   | Achtung!
  + Host/Service: check_period                  = 24x7  #   |
```

```
                  check_interval                = 5     #   |
                  max_check_attempts            = 3     #   |
                  retry_interval                = 1     #   |
                  check_command                 = CMD   #   |
                  notifications_enabled         = 1     # <-+
                  notification_options          = c,w,u,r,f
                  notification_period           = 24x7
                  notification_interval         = 120
                  contacts                      = ...
                  contact_groups                = ...
  + Contact:      host_notification_options      = n,d,u,r,f,s,a
                  host_notification_period       = 24x7
                  host_notification_commands     = CMD, ...
                  service_notification_options   = n,w,c,u,r,f,s,a
                  service_notification_period    = 24x7
                  service_notification_commands  = CMD, ...
  + Command:      command_name                  = CMD
                  command_line                  = EXTERN_CMD...
```

* Bei welcher Host-Statusänderung Benachrichtigung verschicken?
  --> "host_notification_options"

Opt	Name	Bedeutung
-	UP	Erreichbar (direkt oder indirekt)
d	DOWN	Nicht erreichbar (direkt)
u	UNREACHABLE	Nicht erreichbar (indirekt)
r	RECOVERING	Wieder erreichbar (nach Nichterreichbarkeit)
f	FLAPPING	Oszillierend/schwankend (Heuristik)
s	SCHEDULED	Geplante Downtime
n	NONE	Nie
a	ALL	Immer (undokumentiert)

* Bei welcher Service-Statusänderung Benachrichtigung verschicken?
  --> "service_notification_options"

Opt	Name	Bedeutung
-	OK	In Ordnung
w	WARNING	Problematisch
c	CRITICAL	Kritisch
u	UNKNOWN	Unbekannt (z.B. weil nicht erreichbar)
r	RECOVERING	Wieder erreichbar (nach Nichterreichbarkeit)
f	FLAPPING	Oszillierend/schwankend (Heuristik)
s	SCHEDULED	Geplante Downtime
n	NONE	Nie
a	ALL	Immer (undokumentiert)

* Wiederholung
  + Eskalation
  + Bestätigung

25) Makros

* Makros sind das Salz in der Suppe von Nagios
  + Ohne sie müsste jeder Service für jeden Host einzeln UND
  + Jedes Kommando für jeden Host und jeden Service getrennt definiert werden

* Makros erkennbar an
  + $...$ aussenrum (ACHTUNG: bei Shell/Perl/PHP nur führendes $-Zeichen!)
  + GROSSSCHREIBUNG

* Makro-Typen
  + User Makro: $USERn$
    - In Datei "resource.cfg" (32 Stück, Konstanten-Charakter)
    - Für Pfade (zu Plugin)

```
    - Für GEHEIME Daten (Account, Passwort), da nicht lesbar für alle
  + Standardmakro:             $HOSTADDRESS$, $HOSTSTATE$, ...
    - SEHR VIELE!
    - Aus Objektdefinitionen dynmaisch abgeleitet (nur lesbar)
    - LOKALE Variablenzugriffe (abhängig vom Host/Service/...)
  + Kommandozeilen-Argumente:  $ARGn$
    - Zur Datenübergabe von Service-Definition --> Externes Kommando
    - Numeriert 1..n von links nach rechts (32 Stück $ARG1$ .. $ARG32$)
    - Service-Definition:  Werteliste für $ARGn$ mit "!" als Trennzeichen
    - Kommando-Definition: Platzhalter $ARGn$ im Aufruf des externen Kommandos
  + On-demand Makro:           $MAKRO[:HOST[:SERVICE]]$
    - Für GLOBALE Variablenzugriffe (über Objekte hinweg!)
    - Beispiele:
        $HOSTADDRESS:linux02$            # IP-Adresse von Host "linux02"
        $SERVICESTATE:switch05:PING$     # Status-Text Service "PING" Host "Switch05"
        $SERVICESTATEID:switch05:PING$   # Status-Nummer Service "PING" Host "Switch05"
        $SERVICESTATE::PING$             # Status-Text Service "PING" aktueller Host
        $CONTACTNAME:smith$              # Kontaktname von Rechner "smith"
    - Erweiterte Standardmakros für Zugriff auf Wert eines fremden Objekts
      (Standardmakros beziehen sich nur auf akt. Host/Service/Kontakt-Objekt)
    - Auch für Host/Service/Kontakt-Gruppen möglich
  + Custom Makro (ab V3!)
    - NUR für Host/Service/Contact (nicht für ...gruppen)!
      (d.h. TYP = HOST / SERVICE / CONTACT)
    - Benutzerdefinierte Variablen in Objekt TYP (beliebig viele)
      . Definition: _NAME         __NAME        # In Objekt TYP
      . Zugriff:    $_TYPNAME$    $_TYP_NAME$   # 2. Variante besser!
    - Erweitern Host/Service/Kontakt-Objekte um frei wählbare Daten
      (z.B. weitere IP-Adresse eines Hosts für Cluster-Partner, ILO, ...
            oder Anmeldedaten die nicht in der GUI erscheinen sollen)
    - Bei Definition GROSS/kleinschreibung egal,
      bei Zugriff immer GROSSschreibung
    - Beispiel:
      . Definition
          define host {
            host_name   linux01
            ...
            _SSHPORT  10022    # EIN "_" ALS PRÄFIX NOTWENDIG!
            __SSHPORT 10023    # Zugriff besser lesbar!
          }
      . Zugriff:
          $_HOSTSSHPORT$       # Aus Host-Objekt --> 10022
          $_HOST_SSHPORT$      # Aus Host-Objekt --> 10023 (besser lesbar!)
          $_SERVICE...$        # Aus Service-Objekt
          $_SERVICE_...$       # Aus Service-Objekt (besser lesbar!)
          $_CONTACT...$        # Aus Kontakt-Objekt
          $_CONTACT_...$       # Aus Kontakt-Objekt (besser lesbar!)
  + Acknowledgement-Daten aus Weboberfläche:
    - $HOSTACKAUTHOR$       # Ersteller einer Host-Bestätigung
    - $HOSTACKCOMMENT$      # Text einer Host-Bestätigung
    - $SERVICEACKAUTHOR$    # Ersteller  einer Service-Bestätigung
    - $SERVICEACKCOMMENT$   # Text einer Service-Bestätigung
    Entfernen gefährlicher Zeichen darin per:
      illegal_object_name_chars = `~!$%^&*|'"<>?,()=
      illegal_macro_output_chars    = `~$&|'"<>

* Einsetzbar bei
  + Host- und Service-Check
  + Host- und Service-Benachrichtigung
  + Aufruf von Event-Handlern
  + OCSP/OCHP-Kommandos
  + Verarbeitung von Performance-Daten

* Oft benutzte Standard-Makros (sehr viele --> Auswahl der wichtigsten!):
  --> http://nagios.sourceforge.net/docs/3_0/macrolist.html
  --> http://nagios.sourceforge.net/docs/3_0/macros.html
   +----------------------+-------------------------------------------------+
   | Name                 | Beschreibung                                    |
   +----------------------+-------------------------------------------------+
   | $HOSTNAME$           | Aus Host-Def: "host_name"                       |
   | $HOSTALIAS$          | Aus Host-Def: "alias"                           |
   | $HOSTADDRESS$        | Aus Host-Def: "address" (IP/FQHN)               |
   +----------------------+-------------------------------------------------+
   | $HOSTSTATE$          | Zustand UP/DOWN/UNREACHABLE (Text)              |
   | $HOSTSTATEID$        | Zustand 0/1/3 (numerisch)                       |
   | $HOSTSTATETYPE$      | Zustand-Typ HARD/SOFT (Text)                    |
```

$HOSTATTEMPT$	Anzahl Testversuche im SOFT-Zustand	
$HOSTOUTPUT$	Host-Check: Textausgabe 1.Zeile	
$HOSTLONGOUTPUT$	Host-Check: Textausgabe Langtext	
$HOSTPERFDATA$	Host-Check: Textausgabe Performance-Daten	
$HOSTGROUPNAME$	Name 1. Hostgruppe	Analog:
$HOSTGROUPNAMES$	Liste aller Hostgruppen (kommasepariert)	SERVICE...
$HOSTGROUPALIAS$	Alias der Hostgruppe (On-Demand-Makro)	CONTACT...
$HOSTGROUPMEMBERS$	Mitglieder der Hostgruppe (On-Demand-Makro)	
$SERVICEDESC$	Aus Service-Def: "service_description"	
$SERVICESTATE$	Zustand OK/WARNING/CRITICAL/UNKNOWN (Text)	
$SERVICESTATEID$	Zustand 0/1/2/3 (numerisch)	
$SERVICESTATETYPE$	Zustand-Typ HARD/SOFT (Text)	
$SERVICEATTEMPT$	Anzahl Testversuche im SOFT-Zustand	
$SERVICEOUTPUT$	Service-Check: Textausgabe 1.Zeile	
$SERVICELONGOUTPUT$	Service-Check: Textausgabe Langtext	
$SERVICEPERFDATA$	Service-Check: Textausgabe Performance-Daten	
$SERVICEGROUPNAME$	Name 1. Servicegruppe	Analog:
$SERVICEGROUPNAMES$	Liste aller Servicegruppen (kommasepariert)	SERVICE
$SERVICEGROUPALIAS$	Alias der Servicegruppe (On-Demand-Makro)	CONTACT
$SERVICEGROUPMEMBERS$	Mitglieder der Servicegruppe (On-Demand-Makro)	
$CONTACTNAME$	Aus Contact-Def: "contact_name"	
$CONTACTALIAS$	Aus Contact-Def: "alias"	
$CONTACTEMAIL$	Aus Contact-Def: "email"	
$CONTACTPAGER$	Aus Contact-Def: "pager"	
$CONTACTADDRESSn$	Aus Contact-Def: "address" (n=1..6)	
$NOTIFICATIONTYPE$	Benachrichtigungs-Typ (LISTE)	
$NOTIFICATIONRECIPIENTS$	Liste Empfänger (kommasepariert) V3	
$HOSTNOTIFICATIONNUMBER$	Benachrichtigungszähler V3	
$SERVICENOTIFICATIONNUMBER$	Benachrichtigungszähler V3	
$LONGDATETIME$	Sa 29. Dez 17:23:22 CET 2007 (analog "date")	
$SHORTDATETIME$	2007-12-29 17:23:22	
$DATE$	2007-12-29	
$TIME$	17:23:22	
$TIMET$	119845589 (Timestamp: Sek. seit 1.1.1970 00:00:00)	
$TOTALHOSTSUP$	Statistik: Host-Anz UP	
$TOTALHOSTSDOWN$	Statistik: Host-Anz DOWN	
$TOTALHOSTSUNREACHABLE$	Statistik: Host-Anz UNREACHABLE	
$TOTALSERVICESOK$	Statistik: Service-Anz OK	
$TOTALSERVICESWARNING$	Statistik: Service-Anz WARNING	
$TOTALSERVICESCRITICAL$	Statistik: Service-Anz CRITICAL	
$TOTALSERVICESUNKNOWN$	Statistik: Service-Anz UNKNOWN	
$TOTALSERVICEPROBLEMS$	Statistik: Service-Anz WARN+CRIT+UNKNOWN	
$HOSTACKAUTHOR$	Ersteller einer Host-Bestätigung	
$HOSTACKCOMMENT$	Text einer Host-Bestätigung	
$SERVICEACKAUTHOR$	Ersteller einer Service-Bestätigung	
$SERVICEACKCOMMENT$	Text einer Service-Bestätigung	

* Standardmakros werden auch als Umgebungsvariablen zur Verfügung gestellt
  + $HOSTADDRESS$ --> $NAGIOS_HOSTADDRESS
  + On-Demand: Keine Host/Service-Makros aus Sicherheitsgründen
  + $USERn$: Nicht aus Sicherheitsgründen
  + Sehr zeitaufwendig --> Abschalten (nur in V3 möglich)
    - use_large_installation_tweaks = 1
    - enable_environment_macros   = 0

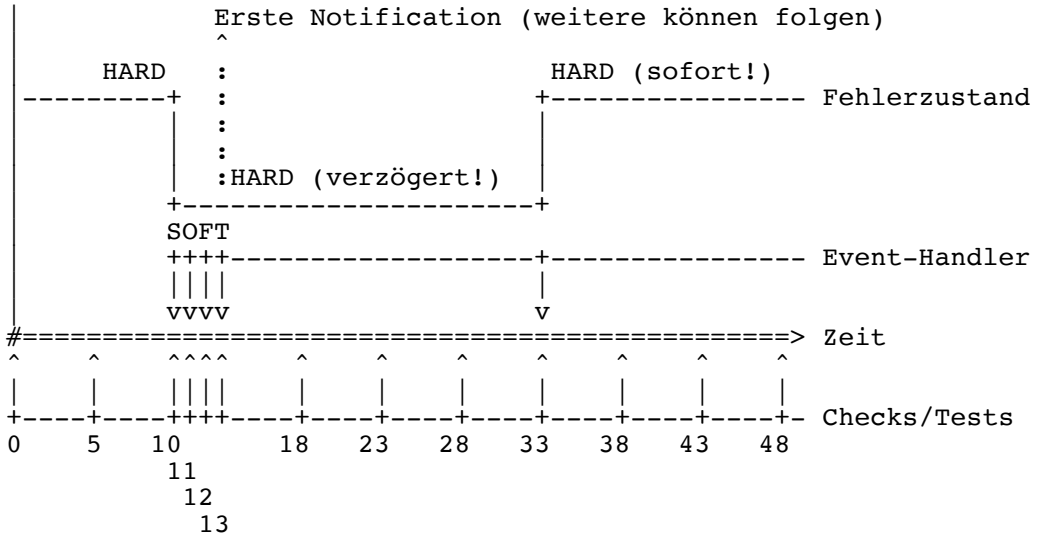
* $NOTIFICATIONTYPE$
  + PROBLEM
  + RECOVERY
  + ACKNOWNLEDGEMENT
  + FLAPPING START
  + FLAPPING STOP

  + FLAPPING
  + DOWNTIME START
  + DOWNTIME STOP
  + DOWNTIME CANCELLEd

26) Event-Handler

* Behebung eines Problems ist BESTE, autom. Restart nur ZWEITBESTE Lösung
  + Nicht jeder Dienst eignet sich für automatischen Restart
    - Evtl. ständiger Neustart
    - Evtl. Datenverlust
    - Evtl. nur in Logdatei vermerkt --> Problem wird gar nicht bemerkt!

* Ausgeführt für
  + Alle SOFT-Fehlerzustände
  + Erstes Eintreten eines HARD-Fehlerzustands
  + Wiederherstellung des OK-Zustands nach einem Fehler



* Aufruf bei JEDEM Event, es muss aber nicht jedesmal eine Aktion erfolgen

```
define host {
  ...
  event_handler  restart-FOO
  ...
}

define service {
  ...
  event_handler  restart-FOO
  ...
}

define command {
  command_name  restart-FOO
  command_line  $USER1$/eventhandler/restart-FOO \
                $SERVICESTATE$ $SERVICESTATETYPE$ $SERVICEATTEMPT$
}
```

* Liegen in /usr/local/nagios/libexec/eventhandler/
  /usr/share/nagios3/plugins/eventhandler/

* Service: Drei Parameter mit folgenden Makro-Werten
  - $SERVICESTATE$ # Zustand OK, WARNING, CRITICAL, UNKNOWN
  - $SERVICESTATETYPE$ # Zustand-Typ SOFT, HARD
  - $SERVICEATTEMPT$ # Versuch 1, 2, ...

* Host: Drei Parameter mit folgenden Makro-Werten
  - $HOSTSTATE$ # Zustand UP, DOWN, UNREACHABLE
  - $HOSTSTATETYPE$ # Zustand-Typ SOFT, HARD
  - $HOSTATTEMPT$ # Versuch 1, 2, ...

* Beispiel: Nur beim 3. Soft- und beim ersten Hard-Fehlerzustand etwas tun

(Shell-Skript, mit Rechten von Benutzer "nagios" ausgeführt)

```
# $1=$SERVICESTATE$ $2=$SERVICESTATETYPE$ $3=$SERVICEATTEMPT$
case "$1" in
  OK) ;;
  WARNING) ;;
  CRITICAL)  if [ "$2" = "HARD" -o "$2" = "SOFT" -a "$3" -eq 3 ]
             then
               echo "Restarting service FOO"
               /usr/bin/sudo /etc/init.d/FOO restart
             fi ;;
  UNKNOWN) ;;
esac
exit 0
```

Notwendiger Inhalt der "/etc/sudoers"-Datei für obiges Skript:

```
nagios ALL=(root) NOPASSWD: /etc/init.d/FOO
```

27) Flap Detection

* Unzuverlässige Checks (oszillierend) temporär deaktivieren

* Letzte 21 Zustände (Testergebnisse) jedes Services gemerkt (Historyspeicher)
  + Nur Hard-Zustände und Soft-Recovery
  + Zeitliche Abstände nicht konstant
    - interval_length   = 60 # Sekunden
    - check_interval   = 5   # Minuten (alt: "normal_check_interval")
    - max_check_attempts   = 5   # Check-Wiederholungen bei Status-Änderung
    - retry_interval   = 2   # Minuten (alt: "retry_check_interval")

* Max. 20 Status-Wechsel in diesen 21 Zuständen möglich
  prozentuale Anzahl an Wechseln (gewichtet von alt 0.8 --> 1.2 neu):
  +   0% = 0 Wechsel (nie)
  + 25% = 5 Wechsel
  + 50% = 10 Wechsel
  + 75% = 15 Wechsel
  + 100% = 20 Wechsel (jedesmal)

* 2 Schwellwerte für Host und für Service
  + Oberer überschritten   --> Keine Meldung mehr, nichtpermanenter Kommentar
  + Unterer unterschritten --> Wieder Meldungen

* Globale Einstellungen (nagios.cfg)
  + enable_flap_detection   =   1   # Flag
  + low_host_flap_threshold   = 5.0   # Prozent
  + high_host_flap_threshold   = 20.0   # Prozent
  + low_service_flap_threshold = 5.0   # Prozent
  + high_service_flap_threshold = 20.0   # Prozent

* Konfiguration Hosts (nach "flap" suchen):
  + flap_detection_enabled   =   1   # Flag
  + low_flap_threshold   = 5.0   # Prozent
  + low_flap_threshold   = 20.0   # Prozent
  + flap_detection_options   = o,d,u   # Nur OK,DOWN,UNREACHABLE

* Konfiguration Services (nach "flap" suchen):
  + flap_detection_enabled   =   1   # Flag
  + low_flap_threshold   = 5.0   # Prozent
  + low_flap_threshold   = 20.0   # Prozent
  + flap_detection_options   = o,w,c,u   # Nur OK,WARNING,CRITICAL,UNKNOWN

28) External Commands (externe Kommandos)

* Für Kommunikation externer Programme mit Nagios (write-only) oder
  von Event-Handlern mit Nagios
  (ob Kommando akzeptiert wurde kann aus Logfile ermittelt werden, read-only)
  + GUI   --> Daemon steuern
  + Nagios Events   --> Daemon steuern
  + Passive Check Plugins --> Daemon informieren

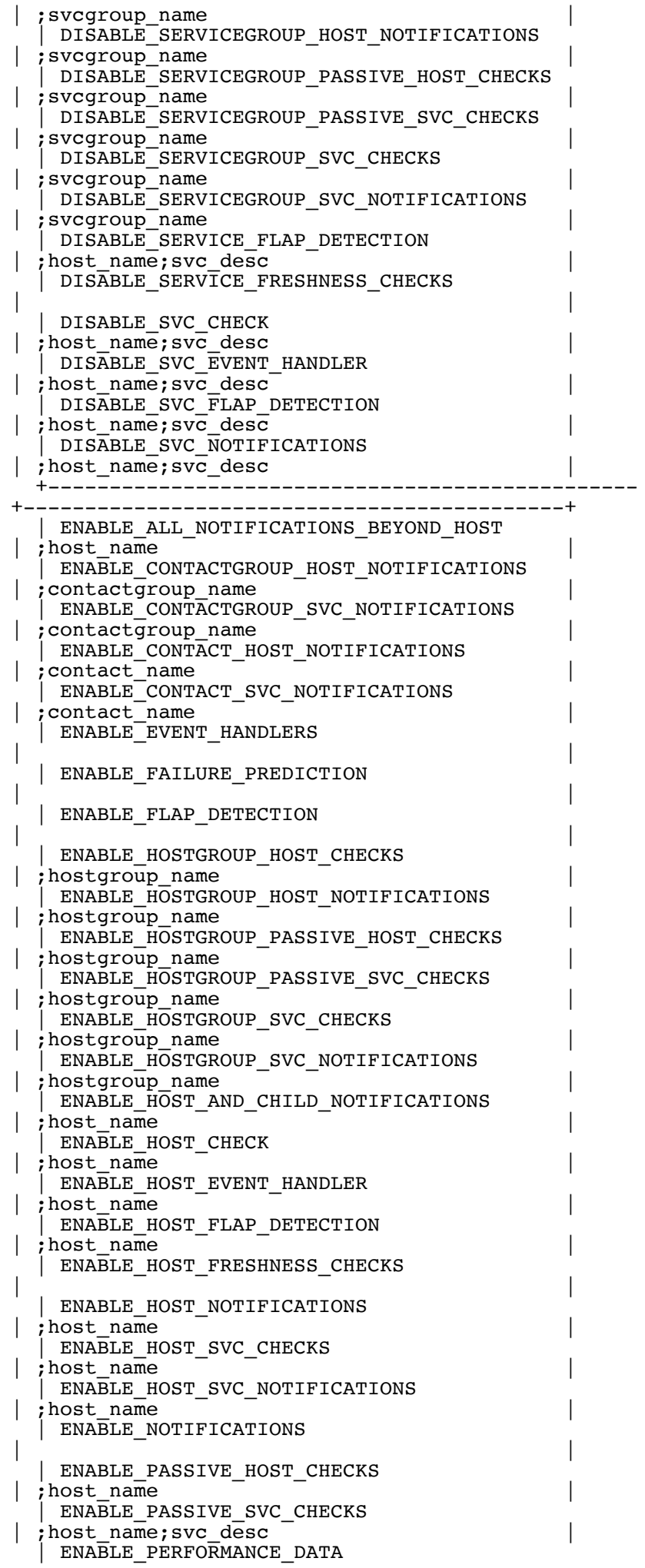
```
  + Benutzer(kommandos)   --> Daemon steuern

* Beispiel
  + Host/Service SOFORT checken
  + Host/Service/Contact/... Custom Variable ändern

* Event-Handler
  - Bei jedem Statuswechsel ausgeführt
  - Bei SOFT und HARD-Statuswechsel ausgeführt
    --> Reaktion möglich bevor Nagios Notification verschickt
    $SERVICESTATE$
    $SERVICESTATETYPE$
    $SERVICEATTEMPT$
  --> 26) Event-Handler

* Kommando-Beschreibung siehe:
  --> http://www.nagios.org/developerinfo/externalcommands/commandlist.php
  --> http://old.nagios.org/developerinfo/externalcommands/commandinfo.php
  --> http://www.nagios.org/developerinfo/externalcommands/commandlist.php?
command_id=NNN

* 14 Kategorien
  + Adaptive Monitoring
  + Comments
  + Contacts
  + Event Handlers
  + Flap Detection
  + Hosts
  + Misc
  + Nagios Process
  + Notifications
  + OCHP/OCSP = Obsessive Compulsive Host/Service Processor Command
  + Performance Data
  + Scheduled Downtime
  + Service and Host Checks
  + Services

* 157 Kommandos
  + Nummer 1-158 (133 nicht belegt)
  + Einzeilig
  + GROSSschreibung wichtig
  + Parameter durch ";" zu trennen
  + Korrekte Namen von Hosts, Services, ... sind Voraussetzung
  + Kein Output
  + Keine Rückmeldung

* Absetzen durch Schreiben in "External Command File"
    --> /usr/local/nagios/var/rw/nagios.cmd
  + "Named Pipe/FIFO" zur Kommunikation mit Nagios von außen (Aufruf)
  + Aktivieren in "nagios.cfg":
    --> check_external_commands = 1
  + Kein Schutz (keine Authentifizierung) außer Zugriffsrechte
    - Besitzer "nagios"
    - Gruppe "nagioscmd"
  + Keine Antwort --> check_logfile
  + Format (TS=Timestamp=date +%s, CMD=Uppercase, ARG1-n=Argumente)
      [TS] CMD;ARG1;ARG2;...;ARGn\n
  + Beispiel (Retention-Datei sichern/lesen)
      CMDFILE=/usr/local/nagios/var/rw/nagios.cmd
      echo "[$(date +%s)] SAVE_STATE_INFORMATION" > $CMDFILE
      echo "[$(date +%s)] READ_STATE_INFORMATION" > $CMDFILE
  + Beispiel (Kommentar zu Host/Service eintragen)
      echo "[$(date +%s)] ADD_HOST_COMMENT;host;1;tsbirn;Kommentar..." > $CMDFILE
      echo "[$(date +%s)] ADD_SVC_COMMENT;host;svcdesc;1;tsbirn;Kommentar..." >
$CMDFILE
  + Beispiel (Nagios passiv schalten):
      echo "[$(date +%s)] DISABLE_EVENT_HANDLERS"             > $CMDFILE
      echo "[$(date +%s)] DISABLE_FAILURE_PREDICTION"         > $CMDFILE
      echo "[$(date +%s)] DISABLE_FLAP_DETECTION"             > $CMDFILE
      echo "[$(date +%s)] DISABLE_HOST_FRESHNESS_CHECKS"      > $CMDFILE
      echo "[$(date +%s)] DISABLE_HOST_NOTIFICATIONS"         > $CMDFILE
      echo "[$(date +%s)] DISABLE_NOTIFICATIONS"              > $CMDFILE
      echo "[$(date +%s)] DISABLE_PASSIVE_HOST_CHECKS"        > $CMDFILE
      echo "[$(date +%s)] DISABLE_PERFORMANCE_DATA"           > $CMDFILE
      echo "[$(date +%s)] DISABLE_SERVICEGROUP_HOST_CHECKS"   > $CMDFILE
      echo "[$(date +%s)] DISABLE_SERVICE_FRESHNESS_CHECKS"   > $CMDFILE
```

```
      echo "[$(date +%s)] DISABLE_SVC_CHECK"                  > $CMDFILE
      echo "[$(date +%s)] STOP_ACCEPTING_PASSIVE_HOST_CHECKS" > $CMDFILE
      echo "[$(date +%s)] STOP_ACCEPTING_PASSIVE_SVC_CHECKS"  > $CMDFILE
      echo "[$(date +%s)] STOP_EXECUTING_HOST_CHECKS"         > $CMDFILE
      echo "[$(date +%s)] STOP_EXECUTING_SVC_CHECKS"          > $CMDFILE
      echo "[$(date +%s)] STOP_OBSESSING_OVER_HOST_CHECKS"    > $CMDFILE
      echo "[$(date +%s)] STOP_OBSESSING_OVER_SVC_CHECKS"     > $CMDFILE
```

* Liste der Kommandos mit ihren Parametern:

Kommando-Name	Kommando-Parameter (durch ";" getrennt)
ACKNOWLEDGE_HOST_PROBLEM	;host_name;sticky;notify;persistent;author
ACKNOWLEDGE_SVC_PROBLEM	;host_name;svc_desc;sticky;notify
ADD_HOST_COMMENT	;host_name;persistent;author;comment
ADD_SVC_COMMENT	;host_name;svc_desc;persistent;author
CHANGE_CONTACT_HOST_NOTIFICATION_TIMEPERIOD	;contact_name
CHANGE_CONTACT_MODATTR	;contact_name;value
CHANGE_CONTACT_MODHATTR	;contact_name;value
CHANGE_CONTACT_MODSATTR	;contact_name;value
CHANGE_CONTACT_SVC_NOTIFICATION_TIMEPERIOD	;contact_name
CHANGE_CUSTOM_CONTACT_VAR	;contact_name;var_name;var_value
CHANGE_CUSTOM_HOST_VAR	;host_name;var_name;var_value
CHANGE_CUSTOM_SVC_VAR	;host_name;svc_desc;var_name;var_value
CHANGE_GLOBAL_HOST_EVENT_HANDLER	;event_handler_cmd
CHANGE_GLOBAL_SVC_EVENT_HANDLER	;event_handler_cmd
CHANGE_HOST_CHECK_COMMAND	;host_name;check_cmd
CHANGE_HOST_CHECK_TIMEPERIOD	;host_name;check_timeperod
CHANGE_HOST_CHECK_TIMEPERIOD	;host_name;timeperiod
CHANGE_HOST_EVENT_HANDLER	;host_name;event_handler_cmd
CHANGE_HOST_MODATTR	;host_name;value
CHANGE_MAX_HOST_CHECK_ATTEMPTS	;host_name;check_attempts
CHANGE_MAX_SVC_CHECK_ATTEMPTS	;host_name;svc_desc
CHANGE_NORMAL_HOST_CHECK_INTERVAL	;host_name;check_interval
CHANGE_NORMAL_SVC_CHECK_INTERVAL	;host_name;svc_desc
CHANGE_RETRY_HOST_CHECK_INTERVAL	;host_name;svc_desc
CHANGE_RETRY_SVC_CHECK_INTERVAL	;host_name;svc_desc
CHANGE_SVC_CHECK_COMMAND	;host_name;svc_desc;check_cmd
CHANGE_SVC_CHECK_TIMEPERIOD	;host_name;svc_desc
CHANGE_SVC_EVENT_HANDLER	;host_name;svc_desc
CHANGE_SVC_MODATTR	;host_name;svc_desc;value
CHANGE_SVC_NOTIFICATION_TIMEPERIOD	

```
| ;host_name;svc_desc                          |
  +----------------------------------------------
+-----------------------------------------+
  | DELAY_HOST_NOTIFICATION
| ;host_name;notification_time              |
  | DELAY_SVC_NOTIFICATION
| ;host_name;svc_desc;notification_time     |
  +----------------------------------------------
+-----------------------------------------+
  | DEL_ALL_HOST_COMMENTS
| ;host_name                                |
  | DEL_ALL_SVC_COMMENTS
| ;host_name;svc_desc                       |
  | DEL_HOST_COMMENT
| ;comment_id                               |
  | DEL_HOST_DOWNTIME
| ;downtime_id                              |
  | DEL_SVC_COMMENT
| ;comment_id                               |
  | DEL_SVC_DOWNTIME
| ;downtime_id                              |
  +----------------------------------------------
+-----------------------------------------+
  | DISABLE_ALL_NOTIFICATIONS_BEYOND_HOST
| ;host_name                                |
  | DISABLE_CONTACTGROUP_HOST_NOTIFICATIONS
| ;contactgroup_name                        |
  | DISABLE_CONTACTGROUP_SVC_NOTIFICATIONS
| ;contactgroup_name                        |
  | DISABLE_CONTACT_HOST_NOTIFICATIONS
| ;contact_name                             |
  | DISABLE_CONTACT_SVC_NOTIFICATIONS
| ;contact_name                             |
  | DISABLE_EVENT_HANDLERS
|                                           |
  | DISABLE_FAILURE_PREDICTION
|                                           |
  | DISABLE_FLAP_DETECTION
|                                           |
  | DISABLE_HOSTGROUP_HOST_CHECKS
| ;hostgroup_name                           |
  | DISABLE_HOSTGROUP_HOST_NOTIFICATIONS
| ;hostgroup_name                           |
  | DISABLE_HOSTGROUP_PASSIVE_HOST_CHECKS
| ;hostgroup_name                           |
  | DISABLE_HOSTGROUP_PASSIVE_SVC_CHECKS
| ;hostgroup_name                           |
  | DISABLE_HOSTGROUP_SVC_CHECKS
| ;hostgroup_name                           |
  | DISABLE_HOSTGROUP_SVC_NOTIFICATIONS
| ;hostgroup_name                           |
  | DISABLE_HOST_AND_CHILD_NOTIFICATIONS
| ;host_name                                |
  | DISABLE_HOST_CHECK
| ;host_name                                |
  | DISABLE_HOST_EVENT_HANDLER
| ;host_name                                |
  | DISABLE_HOST_FLAP_DETECTION
| ;host_name                                |
  | DISABLE_HOST_FRESHNESS_CHECKS
|                                           |
  | DISABLE_HOST_NOTIFICATIONS
| ;host_name                                |
  | DISABLE_HOST_SVC_CHECKS
| ;host_name                                |
  | DISABLE_HOST_SVC_NOTIFICATIONS
| ;host_name                                |
  | DISABLE_NOTIFICATIONS
|                                           |
  | DISABLE_PASSIVE_HOST_CHECKS
| ;host_name                                |
  | DISABLE_PASSIVE_SVC_CHECKS
| ;host_name;svc_desc                       |
  | DISABLE_PERFORMANCE_DATA
|                                           |
  | DISABLE_SERVICEGROUP_HOST_CHECKS
```


;svcgroup_name
DISABLE_SERVICEGROUP_HOST_NOTIFICATIONS
;svcgroup_name
DISABLE_SERVICEGROUP_PASSIVE_HOST_CHECKS
;svcgroup_name
DISABLE_SERVICEGROUP_PASSIVE_SVC_CHECKS
;svcgroup_name
DISABLE_SERVICEGROUP_SVC_CHECKS
;svcgroup_name
DISABLE_SERVICEGROUP_SVC_NOTIFICATIONS
;svcgroup_name
DISABLE_SERVICE_FLAP_DETECTION
;host_name;svc_desc
DISABLE_SERVICE_FRESHNESS_CHECKS
DISABLE_SVC_CHECK
;host_name;svc_desc
DISABLE_SVC_EVENT_HANDLER
;host_name;svc_desc
DISABLE_SVC_FLAP_DETECTION
;host_name;svc_desc
DISABLE_SVC_NOTIFICATIONS
;host_name;svc_desc
ENABLE_ALL_NOTIFICATIONS_BEYOND_HOST
;host_name
ENABLE_CONTACTGROUP_HOST_NOTIFICATIONS
;contactgroup_name
ENABLE_CONTACTGROUP_SVC_NOTIFICATIONS
;contactgroup_name
ENABLE_CONTACT_HOST_NOTIFICATIONS
;contact_name
ENABLE_CONTACT_SVC_NOTIFICATIONS
;contact_name
ENABLE_EVENT_HANDLERS
ENABLE_FAILURE_PREDICTION
ENABLE_FLAP_DETECTION
ENABLE_HOSTGROUP_HOST_CHECKS
;hostgroup_name
ENABLE_HOSTGROUP_HOST_NOTIFICATIONS
;hostgroup_name
ENABLE_HOSTGROUP_PASSIVE_HOST_CHECKS
;hostgroup_name
ENABLE_HOSTGROUP_PASSIVE_SVC_CHECKS
;hostgroup_name
ENABLE_HOSTGROUP_SVC_CHECKS
;hostgroup_name
ENABLE_HOSTGROUP_SVC_NOTIFICATIONS
;hostgroup_name
ENABLE_HOST_AND_CHILD_NOTIFICATIONS
;host_name
ENABLE_HOST_CHECK
;host_name
ENABLE_HOST_EVENT_HANDLER
;host_name
ENABLE_HOST_FLAP_DETECTION
;host_name
ENABLE_HOST_FRESHNESS_CHECKS
ENABLE_HOST_NOTIFICATIONS
;host_name
ENABLE_HOST_SVC_CHECKS
;host_name
ENABLE_HOST_SVC_NOTIFICATIONS
;host_name
ENABLE_NOTIFICATIONS
ENABLE_PASSIVE_HOST_CHECKS
;host_name
ENABLE_PASSIVE_SVC_CHECKS
;host_name;svc_desc
ENABLE_PERFORMANCE_DATA

```
|                                                |
  | ENABLE_SERVICEGROUP_HOST_CHECKS
| ;svcgroup_name                                 |
  | ENABLE_SERVICEGROUP_HOST_NOTIFICATIONS
| ;svcgroup_name                                 |
  | ENABLE_SERVICEGROUP_PASSIVE_HOST_CHECKS
| ;svcgroup_name                                 |
  | ENABLE_SERVICEGROUP_PASSIVE_SVC_CHECKS
| ;svcgroup_name                                 |
  | ENABLE_SERVICEGROUP_SVC_CHECKS
| ;svcgroup_name                                 |
  | ENABLE_SERVICEGROUP_SVC_NOTIFICATIONS
| ;svcgroup_name                                 |
  | ENABLE_SERVICE_FRESHNESS_CHECKS
|                                                |
  | ENABLE_SVC_CHECK
| ;host_name;svc_desc                            |
  | ENABLE_SVC_EVENT_HANDLER
| ;host_name;svc_desc                            |
  | ENABLE_SVC_FLAP_DETECTION
| ;host_name;svc_desc                            |
  | ENABLE_SVC_NOTIFICATIONS
| ;host_name;svc_desc                            |
  +------------------------------------------------
+-----------------------------------------+
  | PROCESS_FILE
| ;file_name;delete                              |
  | PROCESS_HOST_CHECK_RESULT
| ;host_name;status_code;plugin_output           |
  | PROCESS_SERVICE_CHECK_RESULT
| ;host_name;svc_desc;status_code;plugin_outp|
  +------------------------------------------------
+-----------------------------------------+
  | READ_STATE_INFORMATION
|                                                |
  +------------------------------------------------
+-----------------------------------------+
  | REMOVE_HOST_ACKNOWLEDGEMENT
| ;host_name                                     |
  | REMOVE_SVC_ACKNOWLEDGEMENT
| ;host_name;svc_desc                            |
  +------------------------------------------------
+-----------------------------------------+
  | RESTART_PROGRAM
|                                                |
  | SAVE_STATE_INFORMATION
|                                                |
  +------------------------------------------------
+-----------------------------------------+
  | SCHEDULE_AND_PROPAGATE_HOST_DOWNTIME
| ;host_name;start_time;end_time                 |
  | SCHEDULE_AND_PROPAGATE_TRIGGERED_HOST_DOWNTIME
| ;host_name;start_time                          |
  | SCHEDULE_FORCED_HOST_CHECK
| ;host_name;check_time                          |
  | SCHEDULE_FORCED_HOST_SVC_CHECKS
| ;host_name;check_time                          |
  | SCHEDULE_FORCED_SVC_CHECK
| ;host_name;svc_desc;check_time                 |
  | SCHEDULE_HOSTGROUP_HOST_DOWNTIME
| ;hostgroup_name;start_time;end_time            |
  | SCHEDULE_HOSTGROUP_SVC_DOWNTIME
| ;hostgroup_name;start_time;end_time            |
  | SCHEDULE_HOST_CHECK
| ;host_name;check_time                          |
  | SCHEDULE_HOST_DOWNTIME
| ;host_name;start_time;end_time;fixed           |
  | SCHEDULE_HOST_SVC_CHECKS
| ;host_name;check_time                          |
  | SCHEDULE_HOST_SVC_DOWNTIME
| ;host_name;start_time;end_time;fixed           |
  | SCHEDULE_SERVICEGROUP_HOST_DOWNTIME
| ;svcgroup_name;start_time                      |
  | SCHEDULE_SERVICEGROUP_SVC_DOWNTIME
| ;svcgroup_name;start_time;end_time             |
  | SCHEDULE_SVC_CHECK
```

```
| ;host_name;svc_desc;check_time              |
  | SCHEDULE_SVC_DOWNTIME
| ;host_name;svc_desc;start_time;end_time     |
  +-------------------------------------------
+-----------------------------------------+
  | SEND_CUSTOM_HOST_NOTIFICATION
| ;host_name;options;author;comment           |
  | SEND_CUSTOM_SVC_NOTIFICATION
| ;host_name;svc_desc;options                 |
  +-------------------------------------------
+-----------------------------------------+
  | SET_HOST_NOTIFICATION_NUMBER
| ;host_name;notification_number              |
  | SET_SVC_NOTIFICATION_NUMBER
| ;host_name;svc_desc                         |
  +-------------------------------------------
+-----------------------------------------+
  | SHUTDOWN_PROGRAM
|                                             |
  +-------------------------------------------
+-----------------------------------------+
  | START_ACCEPTING_PASSIVE_HOST_CHECKS
|                                             |
  | START_ACCEPTING_PASSIVE_SVC_CHECKS
|                                             |
  | START_EXECUTING_HOST_CHECKS
|                                             |
  | START_EXECUTING_SVC_CHECKS
|                                             |
  | START_OBSESSING_OVER_HOST
| ;host_name                                  |
  | START_OBSESSING_OVER_HOST_CHECKS
|                                             |
  | START_OBSESSING_OVER_SVC
| ;host_name;svc_desc                         |
  | START_OBSESSING_OVER_SVC_CHECKS
|                                             |
  +-------------------------------------------
+-----------------------------------------+
  | STOP_ACCEPTING_PASSIVE_HOST_CHECKS
|                                             |
  | STOP_ACCEPTING_PASSIVE_SVC_CHECKS
|                                             |
  | STOP_EXECUTING_HOST_CHECKS
|                                             |
  | STOP_EXECUTING_SVC_CHECKS
|                                             |
  | STOP_OBSESSING_OVER_HOST
| ;host_name                                  |
  | STOP_OBSESSING_OVER_HOST_CHECKS
|                                             |
  | STOP_OBSESSING_OVER_SVC
| ;host_name;svc_desc                         |
  | STOP_OBSESSING_OVER_SVC_CHECKS
|                                             |
  +-------------------------------------------
+-----------------------------------------+
* Growl notification System für OS X und Windows:
  --> http://www.growl.info/
  --> http://www.growlforwindows.com/


29) Performance-Daten mit PNP4Nagios
------------------------------------

* RRD-Datenbanken
  + Round Robin Database
  + Feste Größe --> Speicherbedarf begrenzt
  + Beschränkt auf vorgegebenes Zeitfenster
  + Für Trends

* Performance-Daten (nach dem "|" in der Plugin-Nachricht)
  + --> http://nagios.frank4dd.com/pnp4nagios/graph?host=.pnp-internal&srv=runtime
  + --> http://localhost/pnp4nagios/graph?host=.pnp-internal&srv=runtime
  + Format:   LABEL=WERT*;WARN;CRIT;MIN;MAX <BLANK> LABEL=... <BLANK> ...
    - Auf Komma "," und ";" achten!
```

```
    - Bei Wert evtl. Einheit mitgeliefert
    - Mehrere Werte per Leerraum getrennt möglich
    - MAX darf fehlen = unendlich
    - MIN darf fehlen = 0
  + Beispiel: Ergebnis von "check_ping" (MAX fehlt jeweils)
    LABEL=WERT   ;WARN   ;CRIT   ;MIN BLANK LABEL=WERT;WARN;CRIT;MIN
      rta=0.067ms;200.000;500.000;0;           pl=0%;40;80;0;
      (rta=round trip, pl=packet loss)
   /usr/lib/nagios/plugins/check_ping -H glasgow -w 200,40% -c 500,80%

     PING OK - Packet loss = 0%, RTA = 1.75 ms|rta=1.747000ms;
200.000000;500.000000;0.000000 pl=0%;40;80;0
     <---------------------------------------->|
<-------------------------------------------------------->
         Status-Nachricht für GUI-Anzeige     |  Performancedaten (hier 2 Werte: rta,
pl)

  + Manche Plugins geben KEINE Performancedaten zurück:
     /usr/lib/nagios/plugins/check_ssh -H glasgow
     --> SSH OK - OpenSSH_5.9p1 Debian-5ubuntu1.1 (protocol 2.0)
  + Weiteres Beispiel:
     /usr/lib/nagios/plugins/check_load -w 8,4,4 -c 10,5,5

     OK - load average: 0.18, 0.17, 0.21|load1=0.180;8.000;10.000;0; \
     load5=0.170;4.000;5.000;0; load15=0.210;4.000;5.000;0;

* Performance-Daten-Verarbeitung in Nagios konfigurieren
    perf_data_timeout        = 5
    process_performance_data = 1                          # Performanzdaten verarbeiten
    service_perfdata_command = process-service-perfdata   # Kmdo zur Verarb. von
Perf.daten
    host_perfdata_command    = process-host-perfdata      # Kmdo zur Verarb. von
Perf.daten
    host_perfdata_file                        = PATH
    service_perfdata_file                     = PATH
    host_perfdata_file_template               = TEXT (mit Makros)
    service_perfdata_file_template            = TEXT (mit Makros)
    host_perfdata_file_mode                   = a|w|p (append,write,pipe)
    service_perfdata_file_mode                = a|w|p (append,write,pipe)
    host_perfdata_file_processing_interval    = 0
    service_perfdata_file_processing_interval = 0
    host_perfdata_file_processing_command     = CMD
    service_perfdata_file_processing_command  = CMD
  PRO Service ist Perf.datenverarb. zu aktivieren:
    process_perf_data  1                  # Z.B. generisches Service-Template
  PRO Host ist Perf.datenverarb. zu aktivieren:
    process_perf_data  1                  # Z.B. generisches Host-Template

* PNP4Nagios in Nagios integrieren --> Zusätzliche Grafik in GUI angezeigt
  PRO Host (z.B. generisches Host-Template)
    action_url         /nagios/pnp/index.php?host=$HOSTNAME$
    define command {
      command_name process-host-perfdata
      command_line $USER1$/process_perfdata.pl -d HOSTPERFDATA
    }
  PRO Service (z.B. generisches Service-Template)
    action_url         /nagios/pnp/index.php?host=$HOSTNAME$&srv=$SERVICEDESC$
    define command {
      command_name process-service-perfdata
      command_line $USER1$/process_perfdata.pl
    }

* Verz.+Dateien für Performance-Daten:
    /opt/nagios/share/perfdata/...          # Bei Quell-Installation
    /var/nagios/perfdata/HOST/SERVICE.rrd   # RRD-Daten
    /var/nagios/perfdata/HOST/SERVICE.xml   # Info zu graf. Aufbereitung

* PNP4Nagios Konfiguration und Integration in Nagios verifizieren:
    -m default --> -m standard --> sync bulk bulk+npcd npcdmod
    perl verify_php_config -m bulk \
                           -c /etc/nagios3/nagios.cfg \
                           -p /etc/pnp4nagios

* Notwendige Pakete für PNP4Nagios:
    RRDTool
    zlib
```

```
    GD
    php5-gd
    php5-zlib

* PNP kennt drei Modi:
  + Default/Standard (siehe oben):
    - Pro Prüfergebnis die Performance-Daten verarbeiten (Skript aufrufen)
    - Hohe Last bei vielen Hosts/Services
  + Bulk:
    - Performance-Daten erst sammeln und zwischenspeichern
    - Periodisch verarbeiten (festes Intervall)
      process_performance_data = 1
      service_perfdata_command = # KEIN Kmdo zur Verarbeitung von Perf.daten
      host_perfdata_command    = # KEIN Kmdo zur Verarbeitung von Perf.daten
      service_perfdata_file          = /var/nagios/service-perfdata
      service_perfdata_file_template = DATATYPE::SERVICEPERFDATA\t\
                                       TIMET::$TIMET$\t\
                                       HOSTNAME::$HOSTNAME$\t\
                                       SERVICEDESC::$SERVICEDESC$\t\
                                       SERVICEPERFDATA::$SERVICEPERFDATA$\t\
                                       SERVICECHECKCOMMAND::$SERVICECHECKCOMMAND$\t\
                                       HOSTSTATE::$HOSTSTATE$\t\
                                       HOSTSTATETYPE:$HOSTSTATETYPE$\t\
                                       SERVICESTATE::$SERVICESTATE$\t\
                                       SERVICESTATETYPE::$SERVICESTATETYPE$
      service_perfdata_file_mode                  = a
      service_perfdata_file_processing_interval = 15
      service_perfdata_file_processing_command  = process-service-perfdata-file
      service_perfdata_file_processing_command  = pnp-bulk-service
      host_perfdata_file          = /var/nagios/host-perfdata
      host_perfdata_file_template = DATATYPE::HOSTPERFDATA\t\
                                    TIMET::$TIMET$\t\
                                    HOSTNAME::$HOSTNAME$\t\
                                    HOSTPERFDATA::$HOSTPERFDATA$\t\
                                    HOSTCHECKCOMMAND::$HOSTCHECKCOMMAND$\t\
                                    HOSTSTATE::$HOSTSTATE$\t\
                                    HOSTSTATETYPE:$HOSTSTATETYPE$
      host_perfdata_file_mode                 = a
      host_perfdata_file_processing_interval = 15
      host_perfdata_file_processing_command  = process-host-perfdata-file
      host_perfdata_file_processing_command  = pnp-bulk-host
      define command {
        command_name process-service-perfdata
        command_line $USER1$/process_perfdata.pl \
                     --bulk=/var/lib/nagios/service-perfdata
      }
      define command {
        command_name process-host-perfdata
        command_line $USER1$/process_perfdata.pl -d HOSTPERFDATA \
                     --bulk=/var/lib/nagios/host-perfdata
      }
  + NPC-Daemon (Nagios Performance C):
    - Periodisch verarbeiten (kein festes Intervall, abhängig von Bedarf)
    - NPC-Daemon überwacht Verz. wohin Performance-Daten verschoben werden
      --> Vollständige Entkopplung der Perf.-Daten-Verarb. vom Nagios-Daemon
          (verteilter Storage --> Verarbeitung auf anderem Host)
    - NPC verarbeitet Perf.daten nur wenn Leistungsreserve auf Nagios-Host vorh.
    - Von Nagios aufgerufene Kommandos verschieben Dateien in vom NPC-Daemon
      ueberwachte Verz. und versehen sie dabei mit einem Zeitstempel
      define command {
        command_name process-service-perfdata-file
        command_line /bin/mv /var/lib/nagios/service-perfdata \
                     /var/lib/nagios/spool/perfdata/service-perfdata.$TIMET$
      }
      define command {
        command_name process-host-perfdata-file
        command_line /bin/mv /var/lib/nagios/host-perfdata \
                     /var/lib/nagios/spool/perfdata/host-perfdata.$TIMET$
      }
    - Konfiguration in /etc/pnp4nagios/npcd.cfg:
       user                  = nagios
       group                 = nagios
       perfdata_spool_dir    = /var/lib/nagios/spool/perfdata/
       perfdata_file_run_cmd = $USER1$/process_perfdata.pl
    - Starten des NPC-Daemons:
       npcd -d -f /etc/init.d/npcd    # -d=Daemon -f=Config file
```

```
      /etc/init.d/npcd start
  + Event-Broker-Modul
    - Direkt in Nagios-Dienst integriert
    - Perf.daten in Spool-Verz. von NPC schreiben

* Templates
  + Zur Erzeugung der Graphen
  + Für alle gängigen Plugins bereits dabei
  + Zuordnung von Templates zu Services
    - 1:1
      Kommando "check_ping" --> Template "check_ping.php"
  + Realisiert per Verlinkung in Verz.:
      share/templates         # Individuelle Tempates (mitgebracht)
      share/templates.dist    # Default-Tempates (Zuordnung)
      cd share/templates
      ln -s ../templates.dist/check_ping.php check_icmp.php
  + Default-Tempate
      share/templates.dist/default.php
```

30) Logging

Nagios führt ein eigenes Logging seiner wichtigsten Aktionen (Start, Stop, ...) auf Datei durch. Ebenso das Rotieren Komprimieren der Logdatei (h=hourly, d=daily, w=weekly, m=monthly, y=yearly) sowie das Verschieben in ein Archivverz.

```
log_file            = /var/log/nagios3/nagios.log
log_archive_path    = /var/log/nagios3/archives
log_rotation_method = d
```

Die Logmeldungen können parallel auch an den Syslog-Daemon (syslog, syslog-ng, rsyslog) geschickt werden:

```
use_syslog = 1
```

Auswahl der in das Logging aufzunehmenden Elemente:

```
log_notifications     = 1   # Benachrichtigungen
log_service_retries   = 1   # Wiederholung von Servic-Checks (bei Fehler)
log_host_retries      = 1   # Wiederholung von Servic-Checks (bei Fehler)
log_event_handlers    = 1   # Aufruf von Event-Handler
log_initial_states    = 0   # Startzustand aller Checks
log_external_commands = 1   # Aufruf externer Kommandos
log_passive_checks    = 1   # Passive Prüfungen (eingehende)
```

31) Debugging

Nagios kann zusätzlich Debug-Meldungen über seine internen Abläufe erzeugen, die Konfiguration erfolgt über folgende Parameter:

```
debug_level         = 0                                  # -1 = alles, Bitmuster
debug_verbosity     = 1                                  # 0/1/2
debug_file          = /var/log/nagios3/nagios.debug      # Debugdatei
max_debug_file_size = 1000000                            # Byte
```

Ist die max. Dateigröße erreicht, wird die Debugdatei mit Endung ".old" versehen und neu begonnen (rollierend, d.h. mehr als 2x max_debug_file_size wird nicht belegt):

```
/var/log/nagios3/nagios.debug       # Aktuelle Debug-Datei
/var/log/nagios3/nagios.debug.old   # Vorherige Debug-Datei
```

Die Auswahl der Loginhalte erfolgt per "debug_level = BITMUSTER" (Summe/Veroderung der Einzelbits)

Wert	Bitbedeutung	Vers
-1	Log everything	
0	Log nothing (default)	
1	Function enter/exit information	
2	Config information	
4	Process information	

8	Scheduled event information	
16	Host/service check information	
32	Notification information	
64	Event broker information	
128	External commands	V4
256	Commands	V4
512	Scheduled downtime	V4
1024	Comments	V4
2048	Macros	V4

Die Auswahl des Logtextumfangs erfolgt über "debug_verbosity = 0/1/2" :

Wert	Bedeutung
0	Grundlegende Infos
1	Ausführliche Infos (Default)
2	Detaillierte Infos

32.1) Erweiterungen von Nagios 3 (26.2.2008)

* Performance-Verbesserung in großen Umgebungen
  + Deutlich verbesserte Hostcheck-Logik (Hostcheck und Servicecheck parallel)
    (bisher alle Service-Checks ausgesetzt bis Host-Checks durchgeführt)
  + Zwischenspeicherung + Nutzung von Check-Ergebnissen (Cache) einstellbar

* Plugin-Ausgabe mehrzeilig und max. 8 KByte
  (bis V2 einzeilig und max. 300 Byte)

* Timeperiods flexibilisiert (bisher nur Wochentag + Zeit möglich)
  + Datumangaben (z.B. "Heilig Drei König")
  + Terminwiederholungen
  + Datumbereiche (z.B. gesetzliche Feiertage)

* Benachrichtigung bei Beginn UND Ende von Downtime-Ereignissen

* Verzögerung von Benachrichtigung einstellbar

* Vererbung (Inheritance)
  + Services erben Benachrichtigung-Einstellungen vom Host (falls fehlend)
  + Mehrfachvererbung von Templates
    + "+"...   # Fügt Attributwert hinzu --> Liste
    + "null"...   # Leert Attributwert

* Erweiterte (hierarchische) Gruppierung erlaubt
  + Host   <-- Hostgroup   <-- Hostgroup   <-- ...
    Service <-- Servicegroup <-- Servicegroup <-- ...
    Contact <-- Contactgroup <-- Contactgroup <-- ...
  + Ausnahmen wegnehmen per "!..."

* Vollständig überarbeitete und neu gestaltete Dokumentation

* Readonly-Zugriff in GUI (Kontakte dürfen keine Kommandos absetzen)
  (sowie Kontakte mit GUI-Zugang ohne Benachrichtigungen)

* Embedded Perl Interpreter (ePN = embedded Perl Nagios)
  + Besser konfigurierbar
  + Ein/Ausschalten pro Plugin

* Zusätzliche Optimierungsparameter

* Viele kleine, unscheinbare Dinge --> leichter, charmanter
  + Neue Makros $...$
  + Benutzerdefinierte Variablen "_XXX" (Custom Variable Makros, ab V3!)
    - Möglich in Objekt-Definition von Host, Service, Contact:
        _XXX   WERT   # OK
        __XXX WERT   # Besser
    - Ansprechen per Präfix _HOST/_SERVICE/_CONTACT + Name!
        Definition in Host: $_HOSTXXX$ bzw. $_HOST_XXX$
        Definition in Service: $_SERVICEXXX$ bzw. $_SERVICE_XXX$
        Definition in Contact: $_CONTACTXXX$ bzw. $_CONTACT_XXX$

```
    - Vererbung auf Hostgroup, Servicegroup, Contactgroup funktioniert
    - In Command dann wirklich eingesetzt

* Neue Plugins
  + check_logfile(s)
  + ...

* Nagios user/group:   nagios + nagios
  Command user/group:  nagios + nagcmd
  GUI-Benutzer:        nagiosadmin
  Lock file:           ${prefix}/var/nagios.lock

* Event Broker

* Gleitkomma-Intervalle (bisher nur ganzzahlig)
    check_interval 2.5
    2.5 * interval_length    # 2.5 x meist 60 Sek --> 150 Sek

* Zusammenfassung von Konfig.-Parametern
       Host V2           Service V2                  V3
    check_interval  normal_check_interval  -->  check_interval
    retry_interval  normal_retry_interval  -->  retry_interval

* Einige Objekt-Attribute optional statt zwingend:
    host:    hostextinfo, alias, ...    (Default: host_name)
    service: serviceextinfo, alias, ... (Default: service_description)
             Optional (bei Bedarf vom Host geerbt):
               contact_groups, notification_interval, notification_period

* Check Plugins
    parallelize_checks fällt weg (immer parallel in V3)

* Verarbeitung von Performance-Daten für Hosts
  --> PNP4Nagios, usw.


32.2) Erweiterungen von Nagios 4 (20.9.2013)
---------------------------------------------
* Service abhängig von Host --> Nicht mehr geprüft wenn Host nicht erreichbar

* Performance-Verbesserungen
  + Core Workers (Lightweight Process für Checks)
  + Verifikation der Konfiguration (O(n) statt O(n^2))
  + Event Queue (insert in O(log n) statt O(n))
  + Makro Resolution (alph. sort. --> binäre Suche in O(log n) statt O(n))

* Objekt-Definitionen
  + Host Attribut "address" optional (Std: host_name, meist aus DNS)
  + Service Attribut "parent" neu (statt Service Dependency verwendbar)
  + Host/Service Attribut "hourly_value" --> Contact Attribut "minium_value"

* Fehlerkorrekturen
  + Contact Inheritance
  + Timeperiods

* Query-Handler zur Kommunikation mit externen Entitäten
  + Konfig-Variable: query_socket
  + Eingebaute Handler: core, wproc, nerd, help, echo
  + Nagios Event Radio Dispatcher (NERD)

* Bibliothek "libnagios" für Entwickler von Core Workers + Query Handlers

* HTML/XML-Ausgabe von Plugins verarbeiten
  + Formatierung in GUI transferierbar

* Entfernt
  + Failure Prediction (was ist das?)
  + Option -o/--dont-verify-objects
  + Embedded Perl

* Veraltet (in nächster Version entfernt)
  + Objekte "hostextinfo" und "serviceextinfo"
  + Option -x/--dont-verify-paths
  + Konfig-Variable: check_result_reaper_frequency
  + Konfig-Variable: max_check_result_reaper_time
  + Konfig-Variable: sleep_time
```

  + Konfig-Variable: external_command_buffer_slots
  + Konfig-Variable: command_check_interval

* Neue GUI auf PHP-Basis (statt C)

* GUI-Themen

33) Glossar

Englisch	Deutsch
acknowledgement	Bestätigung, Quittung
alarm	Benachrichtigung
(event) broker	(Ereignis) Vermittler, Makler
check	Prüfung
compulsive	zwingend, obligatorisch, verpflichtend
dependency	Abhängigkeit
dispatcher	Verteiler, Zuteiler
downtime	Ausfall/Wartungszeitraum
escalation	Eskalation, Verstärkung, Steigerung
event	Ereignis (Status/Zustand-Wechsel!)
event handler	Ereignisverarbeitung
flapping	oszillierend, schwankend, sprunghaft, wankelmütig
inheritance	Vererbung
nagios	Netzwerk-Heiliger
notification	Benachrichtigung, Alarmierung
obsessive	obligatorisch, zwingend, zwanghaft, verpflichtend
outage	Ausfall
parent host	Vorgeschalteter Rechner
query	Abfrage
range	Bereich
recheck / retry check	Prüfungswiederholung
retention	Aufbewahrung
status / state	Zustand
template	Vorlage, Schablone
threshold	Schwellwert, Grenzwert
workflow	Arbeitsablauf

34) Links

34.0) Nagios Dokumentation

  http://nagios.sourceforge.net/docs/nagioscore/3/en/toc.html Nagios V3 Core Dokumentation
  http://nagios.sourceforge.net/docs/nagioscore/4/en/toc.html Nagios V4 Core Dokumentation
  http://nagios.sourceforge.net/docs/3_0/toc.html Nagios V3 Core Dokumentation
  http://nagios.sourceforge.net/docs/nrpe/NRPE.pdf NRPE Dokumentation
  http://nagios.sourceforge.net/docs/nsca/NSCA.pdf NSCA Dokumentation
  http://nagios.sourceforge.net/docs/nsca/NSCA_Setup.pdf NSCA Dokumentation
  http://nagios.sourceforge.net/download/contrib/documentation/german/ Weitere Dokumentation
  http://nagios.sourceforge.net/download/contrib/documentation/english/ Weitere Dokumentation
  http://nagios.sourceforge.net/download/contrib/documentation/misc/ Weitere Dokumentation

34.1) Deutsche Nagios Webseiten

  http://www.monitoring-portal.de/ Monitoring-Portal (Community)
  http://www.monitoring-portal.org/ Monitoring-Portal (Community)
  http://monitoringblog.ch/ Monitoringblog (CH)
  http://www.nagios-portal.de/ Nagios-Portal (alter Name)
  http://www.nagios-portal.org/ Nagios-Portal (alter Name)
  http://www.nagios-wiki.de/ Nagios-Wiki (Community)
  http://listi.jpberlin.de/mailman/listinfo/nagios Nagios-Mailingliste

34.2) Englische Nagios Webseiten

http://www.nagios.org/	Nagios Project (Community)
http://ideas.nagios.org/	Nagios Project (neue Ideen)
http://www.nagioscommunity.org/	Nagios Community
http://www.nagioswiki.org/	Nagios Wiki (Howtos, Best Practises, Code Snippets)
http://support.nagios.com/forum/	Nagios Support Forum
http://labs.nagios.com/	Nagios Development Blog
http://library.nagios.com/	Nagios Library (Tutorials, Videos, Tipps)
http://www.nagios.org/support/mailinglists.php	Nagios Mailinglisten

34.3) Nagios Bücher

http://www.swobspace.net/books/nagios/	Nagios: System- und Netzwerk-Monitoring (Wolfgang Barth)
http://www.nagios-das-praxisbuch.de/	Nagios: Das Praxisbuch (Gerhard Laußer)
http://www.praxisbuchnagios.org/	Praxisbuch Nagios (Tobias Scherbaum)
http://www.nagiosbook.org/	The Nagios Book (Chris Burgess)
http://www.monilog.info/	Das Nagios/Icinga Kochbuch (Timo Kucza, Ralf Staudemeyer)

34.4) Firmen mit Nagios-Expertise

http://www.braintower.de/	Braintower
http://www.cimt.de/	Cimt AG
http://www.consol.de/	ConSol (Plugins)
http://www.dass-it.de/	dass IT
http://www.mathias-kettner.de/	Mathias Kettner (Check_MK, Livestatus, Multisite, WATO)
http://www.heinlein-support.de/	Heinlein Support
http://www.ipodion.at/	Ipidion
http://www.itsolution2.de/	ITsolution2
http://www.it-novum.com/	it-novum (OpenITCockpit, SAP, VMWare, NetApp, AS400/iSeries)
http://www.nagios.com/	Nagios Enterprises
http://www.netways.de/de/produkte/nagios/	Netways (Nürnberg, Icinga)
http://it.oetiker.ch/	Oetiker+Partner
http://www.raxnet.net/	Raxnet (Cacti)
http://www.realstuff.ch/	RealStuff (CH)
http://www.systemmonitoring.de/	SM-Box (SHD, Dresden)
http://www.teamix.de/	Teamix (Nürnberg)
http://www.thomas-krenn.de/	Thomas-Krenn (HW, Server)
http://www.thomas-krenn.de/en/oss/	Thomas-Krenn (OSS-SW, Plugins, TKmon)
http://archive.thomas-krenn.de/	Thomas-Krenn (Archiv)
http://www.tuxad.de/	Tuxad (Herford, nagios-fwb + sac-tools --> ngtx)
http://www.xentity.de/	Xentity

34.5) Quellen für Nagios-Plugins

http://mathias-kettner.de/checkmk_livestatus.html	MK Livestatus
http://www.teamix.org/	Teamix (Nag(ix)SC, LConf, Kohana2, check_sensors)
http://www.mathias-kettner.de/	Mathias Kettner (Check_MK)
http://labs.consol.de/	ConSol (Plugins + AddOns)
http://labs.consol.de/lang/de/nagios/check_logfiles/	check_logfiles ("check_log" verbessert)
http://www.monitoringforge.org/	Monitoring Forge (Plugins + AddOns)
http://www.monitoringexchange.org/	Monitoring EXCHANGE (Groundwork)
http://www.netways.de/de/produkte/nagios_plugins/	Monitoring Plugins von Netways
http://www.netways.org/projects/plugins	Monitoring Plugins von Netways
http://my-plugin.de/wiki/index.php	MY-Plugin (check_multi/ check_generic/\

check_nagios_performance)
  http://packages.python.org/NagAconda/plugin.html   NagAcond (Python Nagios Integration)
  http://www.nagios-plugins.org/   Nagios-Plugins (nun Monitoring-Plugins)
  http://www.nagiosplugins.org/   Nagios-Plugins (nun Monitoring-Plugins)
  http://www.monitoring-plugins.org/   Monitoring-Plugins (ehemals Nagios-Plugins)
  http://nagiosplug.sourceforge.net/   Nagios-Plugins Sourceforge
  http://exchange.nagios.org/   Nagios-Exchange (Plugins, AddOns, Doku)
  http://www.nagiosforge.org/   Nagios-Forge (Plugin-Sammlung)
  http://code.google.com/p/pynag/   PyNag (Python Nagios Extension)
  http://blog.gmane.org/gmane.network.nagios.plugins Nagios plugins help list
  http://www.outsideit.net   OutsideIT (check_win_disk_load u.a.)

34.6) GUI/Web-Interfaces für Nagios

  http://www.vanheusden.com/java/CoffeeSaint/   CoffeeSaint (Java)
  http://fruity.sourceforge.net/   Fruity (PHP-Web-Konfig. --> Fork Lilac)
  http://www.fullyautomatednagios.org/   FAN (Fully Automated Nagios)
  http://sourceforge.net/projects/fannagioscd/   FAN (Fully Automated Nagios)
  http://ignoramus-anagi.sourceforge.net/   IGNORAMUS (Nagios Config Generator)
  http://www.netways.org/projects/lconf   LConf (LDAP-bas. Konfig. Manag. Tool)
  http://lilac--reloaded.sourceforge.net/   Lilac Reloaded
  http://www.lilacplatform.com/   Lilac (Nagios Configurator)
  http://sourceforge.net/projects/monarch/   Monarch (Groundwork Monitor Agent, Web)
  http://www.nagiosadmin.de/   NagiosAdmin (Nagios Administrator, deu)
  http://www.nag2web.de/   Nag2Web
  http://www.nag2web.eu/   Nag2Web
  http://nagat.sourceforge.net/   Nagat (Nagios Administration Tool)
  http://nagedit-nagview.sourceforge.net/   NagEdit + NagView (PHP-Web-Konfig.)
  http://sourceforge.net/projects/nagiosconfig/   Nagios Configurator
  http://sourceforge.net/projects/nagiosweb/   Nagios Web Config
  http://www.nagiosql.org/   NagiosQL (PHP-Web-Konfiguration)
  http://sourceforge.net/projects/nagiosql/   NagiosQL (PHP-Web-Konfiguration)
  http://www.nconf.org/   NConf (Enterprise Nagios Configurator)
  http://sourceforge.net/projects/ncpl/   NCPL (Nagios Control Panel)
  http://www.op5.org/community/plugin-inventory/op5-projects/ninja   Ninja
  http://www.opmon.org/project/opcfg   OpCfg
  http://www.onecmdb.org/   OneCMDB (Configuration Management DB)
  http://www.alienvault.com/community.php?section=Home OSSIM (Open Source Security \
                                                        Information Management)
  http://www.thomas-krenn.de/de/wiki/TKmon   TKmon (Web GUI Nagios Administration)
  http://www.thomas-krenn.de/tk_mon   TKmon (Web GUI Nagios Administration)
  http://www.thruk.org/   Thruk (Web-GUI für Nagios/Icinga/\
                                                        Shinken mit MKLivestatus)
  http://labs.consol.de/lang/en/nagios/thruk/   Thruk (Web-GUI für Nagios/Icinga/\
                                                        Shinken mit MKLivestatus)
  http://mathias-kettner.de/checkmk_multisite.html   MK-Multisite (via MK Livestatus)
  http://mathias-kettner.de/checkmk_livestatus.html MK-Livestatus (Nagios Event Broker \
                                                                  Modul, NEB)
  http://www.mathias-kettner.de/checkmk_wato.html   WATO (Web Admin Tool: Icinga/ Check_MK)

34.7) AddOns für Nagios

  http://apan.sourceforge.net/   APAN (Advanced Performance AddOn for N)
  http://www.netways.de/de/produkte/nagios_addons/business_process/ BP (Impact Analysis)
  http://bp-addon.monitoringexchange.org/   Business Process AddOns (View+Analysis)
  http://www.cacti.net/   Cacti (Graphing solution von raxnet)

```
  http://www.cactiusers.de/                          Cacti Users (Forum, deu)
  http://www.cactiusers.org/                         Cacti Users (Forum, eng)
  http://www.collectd.org/                           collectd (UNIX/Linux System
Statistics \
                                                              Collector)
  http://labs.consol.de/                             ConSol Labs (Plugins + AddOns)
  http://www.coshsh.org/                             coshsh (config, service, host,
shinken)
  http://web.taranis.org/drraw/                      drraw (Web-Frontend für rrdtool)
  http://dnx.sourceforge.net/                        DNX (Distributed Nagios eXecutor, \
                                                         verteiltes Monitoring)
  http://www.netways.de/de/produkte/nagios_addons/eventdb/  EventDB
  http://www.netways.org/projects/eventdb/wiki               EventDB
  http://www.gnokii.org/                             Gnokii
  http://labs.consol.de/lang/de/jolokia/             Jolokia (JMX on Capsaicin)
  http://www.op5.org/community/plugin-inventory/op5-projects/merlin   Merlin (Module
for \
                                                       Effortless Redundancy/
Loadbalancing)
  http://www.sorkmos.com/mntos/                      MNTOS (Multi-Nagios Tactical
Overview \
                                                             System)
  http://labs.consol.de/lang/de/nagios/mod-gearman/ mod_gearman (Lastverteilung)
  http://monitoringforge.org/                        Monitoring Forge (Plugins + AddOns)
  http://www.nagiosfs.de/                            MonitoringFS (alt: NagiosFS, tuxad)
  http://www.teamix.org/projects/nagixsc/wiki        Nag(ix)SC (Alternative zu NRPE/
NSCA)
  http://n2rrd.diglinks.com/cgi-bin/trac.fcgi        N2RRD (Nagios to RRD)
  http://www.steveshipway.org/software/f_nagios.html  NagEventLog \
                                                       (Windows Events --> Nagios NSCA)
  http://www.skeptech.org/doku.php?id=code           Nagfs (Nagios Event Broker als
Filesys)
  http://assets.nagios.com/downloads/exchange/nagiosvshell/vshell.tar.gz  Nagios V-
Shell
  http://code.google.com/p/nagios2cacti/             nagios2cacti (Gateway)
  http://www.nagiosexchange.org/                     Nagios Exchange
  http://exchange.nagios.org/directory/Addons/Passive-Checks/NSCA--2D-Nagios-Service-
Check-Acceptor/details  NSCA (Nagios Service Check Acceptor)
  http://www.nagiosexchange.org/cgi-bin/page.cgi?g=Detailed%2F2274.html;d=1  Send_NSCA
  http://nagiosgraph.sourceforge.net/                NagiosGraph
  http://www.netways.de/de/produkte/nagios_addons/nagiosgrapher/     NagiosGrapher
  http://www.netways.org/projects/grapher                            NagiosGrapher
  http://nagstamon.sourceforge.net/                  NagStaMon \
                                                     (Nagios Status Monitor für Desktop)
  http://www.nagvis.org/                             NagVis (Nagios visualization)
  http://sourceforge.net/projects/nagvis/            NagVis (Nagios visualization)
  http://itefix.no/i2/nagwin                         Nagwin (Nagios(Plugins) unter
Windows)
  http://www.nareto.org/                             NaReTo (Nagios Reporting Tool)
  http://sourceforge.net/projects/nc-net             NC_Net (Windows, .NET 2.0)
  http://exchange.nagios.org/directory/AddOns/Database-Backends/NDOUtils/details
NDOUtils
  http://www.pnp4nagios.org/ndo2fs/de/start          ndo2fs (NDO auf Filesystem
schreiben)
  http://www.nedi.ch/doku.php                        NeDi (Network Discovery)
  http://www.netways.de/de/produkte/nagios_addons/netwaysgrapherv2/  NETWAYSGrapherV2
  http://www.netways.org/projects/grapher-v2                         NETWAYSGrapherV2
  http://www.netways.de/de/produkte/nagios_addons/portal/            NETWAYS Portal
  http://www.tuxad.de/download-ngtx.html             ngtx (MonitoringFS, Tuxad)
  http://www.netways.org/projects/noma               NoMa (Notification Manager)
  http://www.miwi-dv.com/nrpent/                     Nrpe_nt (Windows NRPE-Port)
  http://nsclient.ready2run.nl/                      NSClient (Windows)
  http://nsclient.org/nscp/                          NSClient++ (NSCP, Win)
  http://www.opmon.org/                              OPmon (Verbesserungen von OpCfg,
OpCp, \
                                                        OpDb, OpMonAgent=Windows)
  http://www.opservices.com.br/                      OpMon (OpServices)
  http://perfparse.sf.net/                           PerfParse (perm. Speich. Perf.daten
in SQL-DB)
  http://www.ederdrom.de/pnp/de/start                PNP (PNP is not PerfParse)
  http://www.pnp4nagios.org/pnp/start                PNP4Nagios (Perf.daten + RRD-DB,
Perl)
  http://sourceforge.net/projects/pnp4nagios/        PNP4Nagios (Perf.daten + RRD-DB,
Perl)
  http://oss.oetiker.ch/rrdtool/                     RRDtool (Round Robin Database: \
                                                          MRTG-Verbesserung)
```

  http://www.hendrik-sattler.de/scmxx   SCMxx (Siemens S35i)
  http://simple-evcorr.sourceforge.net/   SEC (Simple Event Correlator)
  http://kodu.neti.ee/~risto/sec/   SEC (Simple Event Correlator)
  http://ssc-serv.com/   SSC Serv (Windows System Statistics
\
                                                             Collection Service)
  http://www.sta.to/ftp/yaps/   Yaps (Yet Another Paging Software,
\
                                                           SMS-Versand)

34.8) SNMP (Simple Network Management Protocol)

  http://www.iana.org/assignments/enterprise-numbers IANA Enterprise Numbers
  http://nagios.manubulon.com/index_commands.html   Manubulon (Nagios SNMP Plugins in
Perl: \
                                                       check_snmp)
  http://support.weblink.ch/kb/a24/anleitung-snmp-fuer-linux-centos-installieren.aspx
Anleitung: SNMP für Linux CentOS installieren
  http://www.mibdepot.com/   MIBdepot
  http://sourceforge.net/projects/nagios-snmp/   Nagios SNMP-Plugins
  http://nagtrap.org/   NagTrap \
                                                     (SNMP-Traps Frontend für Nagios/
Icinga)
  http://www.net-snmp.org/   Net-SNMP
  http://www.oidview.com/mibs/0/Printer-MIB.html   Printer-MIB
  http://www.opencsw.org/packages/scli/   Scli (SNMP cmd line interface,
CSWscli)
  http://www.ibr.cs.tu-bs.de/projects/scli/   Scli (SNMP cmd line interface,
CSWscli)
  http://wwwhome.cs.utwente.nl/~schoenw/scotty/   Scotty (grafisches SNMP-System)
  http://de.wikipedia.org/wiki/SNMP   SNMP (Wikipedia)
  http://www.msxfaq.net/konzepte/snmpbasics.htm   SNMP Basics
  http://de.wikipedia.org/wiki/SNMP   SNMP (Wikipedia)
  http://www.msxfaq.net/konzepte/snmpbasics.htm   SNMP Basics
  http://www.snmptt.org/   SNMPTT (SNMP Trap Translator)

34.9) Mobile Device Interfaces für Nagios (iOS/Android Apps)

  http://exchange.nagios.org/directory/Addons/Frontends-%28GUIs-and-CLIs%29/Mobile-
Device-Interfaces   Übersicht: Mobile Device Interfaces
  http://damien.degois.info/android/aNag/   aNag (Nagios
Client für Android)
  http://www.inag2.com/   iNag2 Nagios
Viewer (Nagios Client für iOS)
  http://itunes.apple.com/de/app/inag-nagios-viewer/id312840671?mt=8   iNag Nagios
Viewer (Nagios Client für iOS)
  http://itunes.apple.com/de/app/snmpmon/id327344119?mt=8   SNMPmon (SNMP
Client für iOS)

34.10) Nagios Distributionen

  http://www.omdistro.org/   OMD (Open Monitoring Distribution)
  http://www.fullyautomatednagios.org/   FAN (Fully Automated Nagios)
  http://sourceforge.net/projects/fannagioscd/   FAN (Fully Automated Nagios)

34.11) Nagios-Foren/Blogs

  http://users.telenet.be/mydotcom/howto/nagios/   Nagios
  http://www.steveshipway.org/   Steve Shipway (Nagios Utils)
  http://exchange.check-mk.org/   Check MK Exchange
  http://velt.de/   Velt
  http://nagios.larsmichelsen.com/   Lars Michelsen (Nagios-Blog)
  http://blog.koehntopp.de/archives/2660-Ein-paar-Gedanken-zu-Zeitreihendaten.html
Koehntopp: Ein paar Gedanken zu Zeitreihendaten
  http://bergs.biz/blog/2007/05/08/monitor-number-of-active-connections-to-mysql-using-
nagios/ MySQL-Connections
  http://blog.barfoo.org/2009/02/13/nagios-integrating-cisco-switches/   Cisco-
Plugins
  http://blog.ciphron.de/2008/06/checkeplus-nagios-checkt-handykosten.html/ Handy-
Check
  http://blog.netways.de/   Netways
  http://nagios.frank4dd.com/   Frank4dd (Plugins, Howtos, Graphs,
Doc)

34.12) Nagios-basierte Alternativen ("Nagios Powered")

```
  http://www.nagios.com/products/                 Übersicht Nagios-Zusatzprodukte
  http://www.groundworkopensource.com/            GroundWork Monitor (integrierte
Lösung)
  http://www.groundworkopensource.org/            GroundWork Monitor (integrierte
Lösung)
  http://www.icinga.org/                          Icinga / Icinga 2 (Nagios-Fork von
Netways)
  http://exchange.icinga.org/                     Icinga Exchange (Erweiterungen zu
Icinga)
  http://www.naemon.org/                          Naemon (Nagios 4.0.1 Fork, Andreas
Ericsson)
  http://naemon.github.io/                        Naemon (Nagios 4.0.1 Fork, Andreas
Ericsson)
  http://www.nagios.com/products/nagiosxi/        Nagios XI (extended Interface,
kommerziell)
  http://www.wuerth-phoenix.com/de/               NetEye (Nagios basiertes IT-System,
\
                                                         Management von Würth
Phoenix)
  http://www.op5.se/                              Op5 Monitor (integrierte Lösung)
  http://www.op5.org/                             Op5 Monitor (integrierte Lösung)
  http://www.open-itcockpit.com/                  OpenITCockpit (integrierte Lösung)
  http://www.openitcockpit.org/                   OpenITCockpit (integrierte Lösung)
  http://www.opsview.org/                         Opsview (Nagios + Net-SNMP + RRDtool
+ \
                                                          Catalyst Web Framework)
  http://www.shinken-monitoring.org/              Shinken (Nagios-Ersatz in Python)
```

34.13) Nagios-Alternativen

```
  http://www.agtech.com.br/                       AG-Monitor (AGTech)
  http://argus.tcp4me.com/download.html           Argus (System und Netzwerk
Monitoring)
  http://www.azeti.net/                           Azeti SONARPLEX
  http://bbwin.sourceforge.net/                   BBWin (Windows-Client für
BigBrother/Xymon/Hobbit)
  http://www.bb4.com/                             Big Brother (1984)
  http://www.bb4.org/                             Big Brother (1984)
  http://www.bigsister.ch/                        Big Sister (Network Monitor)
  http://www.centreon.com/                        Centreon (Oreon, integrierte Lösung)
  http://cricket.sourceforge.net/                 Cricket (Monitoring System)
  http://www.ganglia.info/                        Ganglia (Monitoring System)
  http://hobbitmon.sourceforge.net/               Hobbit Monitor (-> XyMon 8.11.2008)
  http://www.hobbitmon.com/                       Hobbit Monitor (-> XyMon 8.11.2008)
  http://sourceforge.net/projects/hyperic-hqic    Hyperic HQ
  http://www.hyperic.com/products/                Hyperic HQ
  http://lisa-home.sourceforge.net/               LISA (LAN Information Server, NW-
Browser)
  http://www.mmonit.com/                          M/Monit
  http://www.mmonit.com/monit/                    Monit
  http://www.monitorix.org/                       Monitorix (Perl)
  http://www.monito.us/                           Monitor.us (Cloud basiert,
kommerziell)
  http://www.monitis.com/                         Monitis (Cloud basiert, kommerziell)
  http://oss.oetiker.ch/mrtg/                     MRTG (Multi Router Traffic Grapher)
  http://www.mrtg.org/                            MRTG (Multi Router Traffic Grapher)
  http://www.munin-monitoring.org/                Munin (Gr. Netzwerk/System-
Langzeitmon)
  http://www.netmon.org/                          Network Monitoring (Tools,
Resources)
  http://www.netmrg.net/                          NetMRG (Netzwerk Monitor)
  http://nino.sourceforge.net/nino/index.html     Nino (Netzwerk Managagement Tool: \
                                                        "Nino is not Openview")
  http://snad.ncsl.nist.gov/itg/nistnet/          NIST Net (WLAN-Netzwerk-Simulator
für \
                                                          IP-Perf.-Messung)
  http://www.opennms.org/                         OpenNMS (Network Management
Software)
  http://www.openqrm.org/                         OpenQRM
  http://www.openqrm-portal.org/                  OpenQRM + OpenQRM-ng
  http://www.opensaf.org/                         OpenSAF (Service Availability
Framewrk)
  http://opensmart.sourceforge.net                OpenSMART (OSS System Monitoring \
                                                           and Reporting Tool)
  http://www.paessler.com/                        PRTG (MRTG für Windows auf SNMP-
Basis)
```

  http://www.rhq-project.org/display/RHQ/Home   RQH (Network Monitoring, Java)
  http://oss.oetiker.ch/rrdtool/   RRDtool (Logging + Graphing)
  http://www.sensuapp.org/   Sensu (OSS Monitoring Framework)
  http://www.sentinet3.com/   Sentinet3
  http://oss.oetiker.ch/smokeping/   SmokePing (Netzwerk-Latenzen messen)
  http://spong.sourceforge.net/   Spong (System- und Netzwerk-
Monitoring)
  http://www.splunk.com/   Splunk
  http://perso.orange.fr/sebastien.godard/   Sysstat (schlanke Systemmonitor-
Suite)
  http://www.markwell.btinternet.co.uk/webmon/   Webmon (Windows)
  http://www.xymon.com/   XyMon Demo (Bbgen-Addon von Hobbit)
  http://tools.rebel-it.com.au/xymon-rclient/   XyMon-rclient
  http://www.xymonton.org/   XyMon Extensions
  http://www.zabbix.com/   Zabbix (Netzwerk Monitoring Tool)
  http://www.zenoss.org/   Zenoss

34.14) Kommerzielle Monitoring-Produkte

  * CA-Monitoring (CA Wily Introscope)
  * Guard
  * HL Mon
  * HP Openview NNM/ITO / HP Service Activator (HPSA)
  * IBM Director
  * IBM Tivoli
  * InfoVista
  * OpenManage
  * ServerView
  * Siemens @active friend
  * Whatsup

34.15) Wikipedia Artikel zu Nagios

  + http://de.wikipedia.org/wiki/Nagios   Nagios-Artikel (eng)
  + http://de.wikipedia.org/wiki/Icinga   Icinga-Artikel (eng)
  + http://en.wikipedia.org/wiki/Nagios   Nagios-Artikel (deu)
  + http://en.wikipedia.org/wiki/Icinga   Icinga-Artikel (deu)
  + http://de.wikipedia.org/wiki/Check_MK   Check_MK-Artikel (deu)
  + http://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems   Vergleich
Netzwerk-Monitoring Systeme (eng)
  + http://en.wikipedia.org/wiki/Category:Network-related_software   Netzwerk-
Software
  + http://en.wikipedia.org/wiki/Category:Network_management   Netzwerk-
Management
  + http://en.wikipedia.org/wiki/Category:Open_source_network_management_software
Open Source Netzwerk-Management Software

35) Hinweise und Tipps
----

* JEDER Host ist einzeln zu definieren (Mapping auf IP)!

* JEDEM Host MUSS mindestens EIN Service zugeordnet sein
  + Service OK --> kein Host-Check
  --> Solange mindestens ein Service-Check OK liefert, leitet Nagios daraus ab,
      dass auch der Host selbst erreichbar ist und sich im Zustand OK befindet.

* Konfiguration von Objekten getrennt von Kommandos halten.

* Für Hostgroups, Servicegroups und Contactgroups gibt es 2 Definitionsarten
  + Group --> Member1,Member2,Member3 ODER Member1 --> Group
                                       ODER Member2 --> Group
                                       ODER Member3 --> Group
  + TIPP: Nur EINE davon benutzen!

* Die Standard-Mails von Nagios sind zu lang für das Verschicken als SMS
  (vorne steht uninteressanter Teil)
  --> Kommandos "notify-by-email" und "notify-by-sms" anpassen

* "/etc/hosts" für Test mit Mapping Host-Name <-> Host-IP füllen

* Trick zum Erstellen von Nachrichten:
  --> printf "%b" interpretiert "\"-Escapes (Oktale-Form: \0 ... \0NNN)

* Mehrere Adressen zu Host erlaubt --> Plugins müssen sie korrekt behandeln

(address 192.168.0.1,192.168.0.2 --> $HOSTADDRESS$ = 192.168.0.1,192.168.0.2)

* Status UNKNOWN (orange=3) nur für Services,
  UNREACHABLE (grau=2) nur für Hosts.

* Die Schnittstelle der Plugins ändert sich nicht
  --> Nagios2-Plugin mit Nagios3,Nagios4,Icinga,Shinken,Naemon,... einsetzbar

* alias = Name für Anzeige
  + Weglassen wenn identisch zu host_name/service_desc/..._name

* Wechsel WARNING/CRITICAL/UNKNOWN --> OK: Keine Benachrichtigung verschickt!
  Wechsel DOWN/UNREACHABLE   --> OK: Keine Benachrichtigung verschickt!

* State Retention File = Statusinfo Host+Service beim Nagios-Stop
                         für Nagios-Start in Datei aufheben.
  + Siehe "Retention Notes" (--> objectdefinitions.html)
    retain_state_information = 1
    state_retention_file   = /usr/local/nagios/var/retention.dat
    retention_update_interval = 60

* Neue Tricks in Nagios V3 (vererbte Attributwerte löschen/erweitern):
  + Geerbten Host/Service/... ausschließen: !HOST
  + Geerbten Attribut-Werte löschen: null
  + Attribut-Wert zu geerbten Werten hinzufügen: +...
  + Custom Variable Makro: _MAKRO

* "Multiple inheritance" seit V3 möglich --> Vorrang von links nach rechts aufsteigend

* "Multiple parents" seit V3 möglich --> Erreichbar wenn EIN Parent erreichbar

* Host/Service Dependency hat nichts mit parents-Eintrag in Host zu tun!
  + parents (in Host)
    - Physikalische Abhängigkeit (Netzwerk-Topologie)
    - Spiegelt sich in automatisch erzeugter Grafik in GUI wieder!
    - Zum Vermeiden überflüssiger Host/Service-Checks ausgenutzt
  + dependency (in Host/Service Dependency-Objekt)
    - Logische Abhängigkeit (unabhängig von Netzwerk-Topologie)
      (z.B. Datenbank für Anwendung notwendig)
    - Spiegelt sich NICHT in automatisch erzeugter Grafik in GUI wieder!
    - Zum Vermeiden überflüssiger Service-Checks ausgenutzt

* Was hat alles Namen und wo ist GROSS/kleinschreibung relevant?

Objekttyp	Intern	Extern	Alias
Host	ja	DNS,egal	ja
Hostgroup	ja	--	ja
Service	desc/ja	--	ja
Servicegroup	ja	--	ja
Contact	ja	MAIL	MAIL
Contactgroup	ja	--	ja
Command	ja	ja	ja
Timeperiod	ja	--	ja
Hostdependency	--	--	--
Servicedependency	--	--	--
Hostescalation	--	--	--
Serviceescalation	--	--	--
Hostextinfo	--	--	--
Serviceextinfo	--	--	--

* Probleme beim klassischen Monitoring
  + Hosts, Services, Protokolle, Prozesse, Ports
    --> für Anwender zählt nur ANWENDUNG!
  + Monitoring von Prozessen und Ports prüft nur die Serverseite
    und eventuell nur einen Bruchteil der Serverfunktionalität
  + Proprietäre Protokolle lassen sich schwer prüfen
  + Auch bei Standard-Protokollen wie HTTP lässt sich die
    Applikationslogik nur schwer testen
  + Clientseite fehlt völlig

* Debian: Web-GUI Absetzen von Kommandos erlauben (z.B. Test-Rescheduling):

```
  + Siehe "/usr/share/doc/nagios3-common/Debian-README"
    Nagios 3 is not configured to look for external commands in the default
    configuration as a security feature. To enable external commands,
    you need to allow the web server write access to the nagios command pipe
      /var/lib/nagios3/rw/nagios.cmd
    Set "check_external_commands=1" in your nagios configuration,
    and then change the permissions in a way which will be maintained
    across package upgrades (otherwise dpkg will overwrite your
    permission changes). The following is the recommended approach:
    + Activate external command checks in the nagios configuration:
        check_external_commands = 1
      in file
        /etc/nagios3/nagios.cfg
    + Perform the following commands to change directory permissions and
      to make the changes permanent:
        /etc/init.d/nagios3 stop
        dpkg-statoverride --update --add nagios www-data 2750 /var/lib/nagios3/rw
        dpkg-statoverride --update --add nagios nagios    751 /var/lib/nagios3
        /etc/init.d/nagios3 start


================================================================================
```