



Icinga Version 1.6 Dokumentation

[Weiter](#)

Icinga Version 1.6 Dokumentation

Copyright 2009-2011 Icinga Development Team.

Teile Copyright © von Nagios/Icinga-Community-Mitgliedern - weitere Informationen finden Sie in der Datei THANKS in den Icinga-Core-Sourcen.

Kudos an Yoann LAMY für die Erstellung des Vautour Style, den wir für Icinga Classic UI nutzen.

Icinga ist lizenziert unter den Bedingungen der GNU General Public License Version 2 wie von der Free Software Foundation veröffentlicht. Das gibt Ihnen das Recht, Icinga unter bestimmten Bedingungen zu kopieren, zu verteilen und/oder zu modifizieren. Lesen Sie die 'LICENSE'-Datei in der Icinga-Distribution oder lesen Sie die Online-Version der Lizenz für weitere Einzelheiten.

Icinga wird zur Verfügung gestellt „SO WIE ES IST“ ohne „GARANTIE JEGLICHER ART, EINSCHLIESSLICH DER GARANTIE DES DESIGNS, DER VERMARKTBARKEIT UND DER TAUGLICHKEIT FÜR EINEN BESTIMMTEN ZWECK.“

Nagios ist lizenziert unter den Bedingungen der GNU General Public License Version 2 wie von der Free Software Foundation veröffentlicht. Das gibt Ihnen das Recht, Nagios unter bestimmten Bedingungen zu kopieren, zu verteilen und/oder zu modifizieren. Lesen Sie die 'LICENSE'-Datei in der Nagios-Distribution oder lesen Sie die Online-Version der Lizenz für weitere Einzelheiten.

Nagios und das Nagios-Logo sind registrierte Schutzmarken von Ethan Galstad. Alle andere Schutzmarken, registrierte Schutzmarken und Bezeichnungen, die in diesem Dokument genannt werden, können das Eigentum der jeweiligen Besitzer sein. Die hierin enthaltenen Informationen werden zur Verfügung gestellt „SO WIE SIE SIND“ ohne „GARANTIE JEGLICHER ART, EINSCHLIESSLICH DER GARANTIE DES DESIGNS, DER VERMARKTBARKEIT UND DER TAUGLICHKEIT FÜR EINEN BESTIMMTEN ZWECK.“

2011.11.30

Versionsgeschichte

Version 1.6	2011-11-30
1.6 Icinga Dokumentation	
Version 1.x	2010/2011
1.x Icinga Dokumentation	
Version 0.1	2009-09-27
Erste Ausgabe	

Inhaltsverzeichnis

1. Über

[Über Icinga](#)

[What's New in Icinga 1.6](#)

2. Los geht's

[Hinweise für Neulinge](#)

[Schnellstart-Installationsanleitungen](#)

[Icinga-Schnellstart auf Linux](#)

[Icinga-Schnellstart auf FreeBSD](#)

[Icinga-Schnellstart mit IDOUtils](#)

[Icinga-Schnellstart mit IDOUtils auf FreeBSD](#)

[Links zu weiteren Howtos](#)

[Icinga aktualisieren](#)

[IDOUtils-Datenbank aktualisieren](#)

[Windows-Maschinen überwachen](#)

[Linux/Unix-Rechner überwachen](#)

[Netware-Server überwachen](#)

[Netzwerk-Drucker überwachen](#)

[Router und Switches überwachen](#)

[Öffentlich zugängliche Dienste überwachen](#)

3. Icinga konfigurieren

[Konfigurationsüberblick](#)

[Optionen der Hauptkonfigurationsdatei](#)

[Überblick Objektkonfiguration](#)

[Objektdefinitionen](#)

[Host-Definition](#)

[Hostgruppen-Definition](#)

[Service-Definition](#)

[Servicegruppen-Definition](#)

[Kontakt-Definition](#)

[Kontaktgruppen-Definition](#)

[Zeitfenster-Definition \(timeperiod\)](#)

[Befehls-Definition \(command\)](#)

[Service-Abhängigkeits-Definition \(servicedependency\)](#)

[Serviceescalations-Definition](#)

[Host-Abhängigkeits-Definition \(hostdependency\)](#)

[Host-Escalations-Definition](#)

[erweiterte Hostinformations-Definition \(hostextinfo\)](#)

[erweiterte Serviceinformations-Definition \(serviceextinfo\)](#)

- Module-Definition
- Maßgeschneiderte Objektvariablen
- Optionen CGI-Konfigurationsdatei
- Authentifizierung und Autorisierung in den CGIs
- 4. Icinga starten/stoppen/prüfen
 - Überprüfen Ihrer Icinga-Konfiguration
 - Icinga starten und stoppen
- 5. Die Grundlagen
 - Icinga Plugins
 - Makros verstehen und wie sie arbeiten
 - Standard-Makros in Icinga
 - Host-Prüfungen (Host checks)
 - Service-Prüfungen (Service Checks)
 - Aktive Prüfungen (Active Checks)
 - Passive Prüfungen (Passive Checks)
 - Statustypen
 - Zeitfenster
 - Ermitteln des Zustands und der Erreichbarkeit von Netzwerk-Hosts
 - Benachrichtigungen
- 6. Die Benutzeroberflächen
 - Icinga Classic UI: Informationen über die CGIs
 - Informationen zu den CGI-Parametern
 - Ausführen von CGIs auf der Kommandozeile
 - Installation des Icinga-Web Frontend
 - Konfigurationsübersicht Icinga-Web
 - Aktualisierung von Icinga-Web und Icinga-Web Datenbank
 - Einführung in Icinga-Web
 - Einführung in Icinga-Web (>= 1.6)
 - Integration von PNP4Nagios in das Icinga-Web Frontend
- 7. Fortgeschrittene Themen
 - Externe Befehle
 - Eventhandler
 - sprunghafte Services
 - Service- und Host-Frische-Prüfungen
 - Verteilte Überwachung
 - Redundante und Failover-Netzwerk-Überwachung
 - Erkennung und Behandlung von Status-Flattern
 - Benachrichtigungseskalationen
 - Eskalations-Bedingung
 - Bereitschafts-Rotation
 - Service- und Host-Gruppen überwachen
 - Host- und Service-Abhängigkeiten
 - Status Stalking
 - Performance-Daten
 - Geplante Ausfallzeiten
 - Benutzen des Embedded Perl Interpreters
 - Adaptive Überwachung
 - Vorausschauende Abhängigkeitsprüfungen
 - Zwischengespeicherte Prüfungen
 - Passive Host-Zustandsübersetzung
 - Service- und Host-Prüfungsplanung
 - Angepasste CGI-Kopf- und Fußzeilen
 - Objektvererbung

- Zeitsparende Tricks für Objektdefinitionen
- 8. Sicherheit und Leistungsoptimierung
 - Sicherheitsüberlegungen
 - Verbesserte CGI-Sicherheit und Authentifizierung
 - Icinga für maximale Leistung optimieren
 - Schnellstart-Optionen
 - Large Installation Tweaks
 - Nutzung des Icingastats-Utilitys
 - grafische Darstellung von Performance-Informationen mit PNP4Nagios
 - Temporäre Daten
- 9. Integration mit anderer Software
 - Integrationsüberblick
 - SNMP-Trap-Integration
 - TCP-Wrapper-Integration
 - MKLiveStatus-Integration
 - Installation von Icinga-Reporting mit JasperServer
- 10. weitere Software
 - Icinga Addons
 - NRPE
 - NSCA
- 11. Entwicklung
 - Nagios Plugin API
 - Entwickeln von Plugins für die Nutzung mit Embedded Perl
 - Liste der externen Befehle
 - Installation und Benutzung der Icinga-API
 - Die Icinga-Web REST API
- 12. IDOUtils
 - Einleitung
 - Zweck
 - Design-Überblick
 - Instanzen
 - Installation
 - Komponenten
 - Überblick
 - IDOMOD
 - LOG2IDO
 - FILE2SOCK
 - IDO2DB, IDO2DB
 - Konfiguration der IDOUtils
 - IDOMOD Konfigurationsoptionen
 - IDO2DB Konfigurationsoptionen
 - Beispielkonfigurationen
 - Einzelner Server, einzelne Instanz
 - Einzelner Server, mehrere Instanzen
 - Einzelner Server, einzelne Instanz, Log-Datei-Import
 - IDOUtils Database Model
 - Central Tables
 - Debugging Tables
 - Historical Tables
 - Current Status Tables
 - Configuration Tables
 - Datenbank-Anpassungen/Änderungen
- Stichwortverzeichnis

Abbildungsverzeichnis

- 3.1. Beispiel des neuen Headers
- 6.1. Icinga-Web Login-Bildschirm
- 6.2. Icinga-Web Überblick
- 6.3. Icinga-Web Top-Menü
- 6.4. Icinga-Web Live-Suche
- 6.5. Icinga-Web Status-Cronk
- 6.6. Icinga-Web Data-Cronks
- 6.7. Icinga-Web Tactical Overview-Cronks
- 6.8. Icinga-Web Reporting-Cronks
- 6.9. Icinga-Web "Misc"-Cronks
- 6.10. Icinga-Web Zentrale Übersicht
- 6.11. Icinga-Web Host-Status
- 6.12. Icinga-Web Service-Status
- 6.13. Icinga-Web Hostgruppenstatus
- 6.14. Icinga-Web Log
- 6.15. Icinga-Web "Tackle"-Cronk
- 6.16. Icinga-Web Cronk-Leiste
- 6.17. Icinga-Web Cronk-Leiste
- 6.18. Icinga-Web Host-Befehle
- 6.19. Icinga-Web Service-Befehle
- 6.20. Icinga-Web Filtereinschränkung
- 6.21. Icinga-Web Filterbedingung
- 6.22. Icinga-Web Filter aktiv
- 6.23. Icinga-Web Top-Menü Admin
- 6.24. Icinga-Web Benutzer-Administration
- 6.25. Icinga-Web Benutzer editieren
- 6.26. Icinga-Web Gruppen-Administration
- 6.27. Icinga-Web Gruppen
- 6.28. Icinga-Web Rollen
- 6.29. Icinga-Web logs
- 6.30. Icinga-Web Tasks
- 6.31. PNP4Nagios integriert in Icinga-Web
- 8.1. Durchschnittliche Host-/Service-Prüfungslatenz
- 8.2. Service-Statistiken
- 8.3. Host-Statistiken
- 8.4. Durchschnittliche Ausführungszeiten
- 8.5. Externe Befehle
- 8.6. Puffer für externe Befehle
- 8.7. Zwischengespeicherte Host- und Service-Prüfungen
- 8.8. Durchschnittliche Zustandswechsel
- 9.1. Icinga reporting
- 10.1. NRPE
- 10.2. NRPE remote
- 10.3. NSCA
- 12.1. Mögliche Anordnungen
- 12.2. zukünftige Entwicklung: Eine Instanz, mehrere Datenbanken
- 12.3. Instanznamen basierend auf dem geografischen Standorts
- 12.4. Instanznamen basierend auf dem Zweck
- 12.5. Geladenes IDOMOD-Event-Broker-Modul
- 12.6. IDOMOD-Möglichkeiten
- 12.7. LOG2IDO-Utility

- 12.8. [FILE2SOCK-Utility](#)
 - 12.9. [IDO2DB-Daemon](#)
 - 12.10. [IDO2DB mit mehreren Clients](#)
 - 12.11. [Einzelserver, Einzelinstanz](#)
 - 12.12. [Einzelner Server, mehrere Instanzen](#)
 - 12.13. [Einzelner Server, einzelne Instanz, Log-Datei-Import](#)
 - 12.14. [Relationship of Central Tables](#)
 - 12.15. [Relationship of Debugging Tables](#)
 - 12.16. [Relationship of Historical Tables](#)
 - 12.17. [Relationship of Current Status Tables](#)
 - 12.18. [Relationship of Configuration Tables](#)
-

[Weiter](#)

[Kapitel 1. Über](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Kapitel 1. Über

[Zurück](#)

[Weiter](#)

Kapitel 1. Über

Inhaltsverzeichnis

[Über Icinga](#)

[What's New in Icinga 1.6](#)

[Zurück](#)

[Weiter](#)

[Icinga Version 1.6 Dokumentation](#)

[Zum Anfang](#)

[Über Icinga](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Über Icinga

[Zurück](#)[Kapitel 1. Über](#)[Weiter](#)

Über Icinga

Was ist das?

Icinga ist eine System- und Netzwerküberwachungsapplikation. Sie überwacht Hosts und Services, die Sie angeben, und alarmiert Sie, wenn sich die Dinge verschlechtern und wenn sie wieder besser werden.

Icinga läuft unter Linux, allerdings sollte es unter den meisten anderen Unix-Derivaten ebenfalls funktionieren.

Einige der vielen Features von Icinga umfassen:

- Überwachen von Netzwerkdiensten (SMTP, POP3, HTTP, NNTP, PING, etc.)
- Überwachen von Host-Ressourcen (Prozessorauslastung, Diskbelegung, usw.)
- Einfaches Plugin-Design, das es Benutzern erlaubt, schnell eigene Service-Prüfungen zu entwickeln
- Parallel laufende Service-Prüfungen
- Die Möglichkeit, Netzwerk-Host-Hierarchien mit Hilfe von "Eltern"-Hosts zu definieren, um die Erkennung von Hosts zu erlauben, die "down" sind und die Unterscheidung zwischen Hosts, die "down" bzw. unerreichbar sind
- Benachrichtigung von Kontakten, wenn Service- oder Host-Probleme auftreten bzw. gelöst werden (über e-Mail, Pager oder benutzerdefinierte Methoden)
- Die Möglichkeit, Routinen zur Ereignisbehandlung (Eventhandler) zu definieren, die bei Host- oder Service-Ereignissen ablaufen, um proaktive Problemlösungen zu erlauben
- Automatische Rotation von Protokolldateien
- Unterstützung, um redundante Überwachungs-Hosts zu implementieren
- Optionales Classic-Web-Interface, um den aktuellen Netzwerkstatus, Benachrichtigungs- und Problemverläufe, Protokolldateien usw. anzusehen
- Optionales neues Icinga-Web-Interface basierend auf Icinga Core, IDOUtils, API, das eine moderne, rundum erneuerte Web 2.0 Oberfläche mit Multilanguage-Support bietet, um den aktuellen Status und historische Informationen zur Verfügung zu stellen. Es werden Cronks und Filter bereitgestellt und es können Reports erstellt werden

Systemvoraussetzungen

Die einzige Voraussetzung für den Betrieb von Icinga ist eine Maschine, auf der Linux (oder eine UNIX-Variante) läuft und ein C-Compiler. Voraussichtlich werden Sie auch noch TCP/IP konfigurieren wollen, weil die meisten Service-Prüfungen über das Netzwerk durchgeführt werden.

Sie müssen nicht eins der Web-Interfaces benutzen, die in Icinga enthalten sind. Wenn Sie sich allerdings entscheiden, sie zu benutzen, muss zusätzlich die folgende Software installiert sein...

1. Ein Web-Server (vorzugsweise [Apache](#))
2. Thomas Boutells [gd library](#) Version 1.6.3 oder höher (wird benötigt von den [statusmap](#)- und [trends](#)-CGIs)
3. PHP

Lizenzierung

Icinga ist unter den Bedingungen der [GNU General Public License](#) Version 2 lizenziert, wie sie von der [Free Software Foundation](#) veröffentlicht wird. (A.d.Ü.: eine deutsche Übersetzung finden Sie [hier](#).) Die Lizenz gibt Ihnen das Recht, Icinga unter bestimmten Bedingungen zu kopieren, zu verteilen und/oder zu modifizieren. Lesen Sie die 'LICENSE'-Datei in der Icinga-Distribution oder lesen Sie die (englische) [Online-Version der Lizenz](#) für nähere Details.

Icinga wird geliefert WIE ES IST OHNE GARANTIE IRGENDERART, EINSCHLIESSLICH DER GARANTIE FÜR DESIGN, VERMARKTBARKEIT ODER DER TAUGLICHKEIT FÜR EINEN BESTIMMTEN ZWECK.

Danksagungen

Verschiedene Leute haben zu Icinga beigetragen, z.B. durch Meldung von Fehlern, vorschlagen von Verbesserungen, schreiben von Plugins usw.

Beschaffen der neuesten Version

Sie können auf <http://www.icinga.org/> nach neuen Versionen von Icinga suchen.

Kompatibilität

Icinga ist ein "Fork" des wohlbekannten Überwachungssystems [Nagios](#). Eine 100%ige Kompatibilität mit den internen Strukturen des letzteren erlaubt es Ihnen, mit Icinga alle Plugins und Addons zu benutzen, die von verschiedenen Firmen und der großen Community entwickelt wurden bzw. werden.

Icinga und das Icinga-Logo sind Markenzeichen von icinga.org. Alle anderen Markenzeichen, Dienstmarkenzeichen, registrierte Markenzeichen und registrierte Dienstmarkenzeichen können das Eigentum der jeweiligen Inhaber sein.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Kapitel 1. Über](#)

[Zum Anfang](#)

[What's New in Icinga 1.6](#)



What's New in Icinga 1.6

[Zurück](#)

[Kapitel 1. Über](#)

[Weiter](#)

What's New in Icinga 1.6

1.5.0

- Enhancements

Icinga-Core:

- do not update host/service status during scheduler initialization on startup (idoutils already know that) #1741

Classic UI:

- Added option to suppress colouring of hosts/services in maintenance #992
- Added option to show partial hostgroups #1013
- changing drop down menu in status.cgi to jquery-dd #1644
- default downtime duration is configurable in cgi.cfg #1642
- Added ability to combine host and service in one CGI parameter (Aaron Brady) #1633
- Changed tac.cgi to distinct between active, passive and disabled properly #1120
- Add log entries to json output of avail.cgi #1659
- add host urgency sort order in status.cgi (Jochen Bern) #1452
- add author to comment tooltips in status.cgi #1691
- reduce the header part of status.cgi #1507 (&nostatusheader used in frameset menu, info table is smaller)
- authorization for * in cgi.cfg via contactgroups (Stephen Gran) #1752
 (authorized_contactgroup_for_all_hosts, authorized_contactgroup_for_all_services,
 authorized_contactgroup_for_system_information,
 authorized_contactgroup_for_configuration_information,
 authorized_contactgroup_for_all_host_commands,
 authorized_contactgroup_for_all_service_commands,
 authorized_contactgroup_for_system_commands,
 authorized_contactgroup_for_read_only)

- add xml output to avail.cgi for direct source of jasper reports #1712
- add combined view of all unhandled hosts and services including the multiple command dropdowns for each (initial idea by Thruk) #1328
- add "link to this page" to all cgis #1783
- add export icons instead of "export to..." text on cgis supporting it (csv, json, xml, link) #1789
- add highlight_table_rows to status.cgi, can be disabled via cfg option #1777

IDOUtils:

- add update-version-schema for updating the idoutils db schema versions automagically, independant of update-version #1616
- dump multiline mbufs as bulk inserts #1110 (configfile/runtime variables, host contacts/parents, service contacts, timeranges, host/service/contactgroup members; oracle uses bulk arrays binded on prepared insert statements)
- remove host and service checks from default data_processing_options as they are not used by icinga-web #1757
- Oracle: add logging of bind variable value #1314
- Oracle: change perfdata columns to clob #1749
- Oracle: change logentry_data type to clob #1750

Install:

- add --with-p1-file-dir allowing custom location (default is now \$libdir instead of \$bindir) #1569 (only when embedded perl is enabled; affects (new_)mini_epn and icinga.cfg)
- add configure option to set log dir and cgi log dir and phpapi log dir #1715 (e.g. --with-phpapi-log-dir=/var/log/icinga/api
--with-log-dir=/var/log/icinga
--with-cgi-log-dir=/var/log/icinga/gui; apache user permissions)
- after daemon stop in init script, retry and then issue a kill -9 on the daemon #1747

Config:

- add easy-to-use config examples and scripts for notifications #1697 (advanced notify-by-email; notify-by-jabber, notify-by-twitter examples)
- add check_dummy command definitions as example to commands.cfg #1719
- tune icinga.cfg for performance, enable_environment_macros=0, enable_embedded_perl=0 by default #1635

Style:

- add new indent tools #1617

- Fixes

Icinga-Core

- remove duplicated unlinks for check result files with multiple results (Ton Voon) #1615
- fix retain status file over an init script reload #1579
- fix freshness_threshold problem in host checks by using check_interval in HARD or OK state, else retry_interval (like service checks) (Andreas Ericsson) #1331
- make command_check_interval=-1 equal to 5s instead of 60s, targetting as often as possible (Andreas Ericsson) #1651
- fix problem where acknowledgements were getting reset when a hard state change occurred (Ton Voon) #1618
- avoid using global macros when sending notifications (Andreas Ericsson) #1653
- avoid sending notifications to the wrong contact (Andreas Ericsson) #1654
- try to avoid a single check being reaped multiple times (Andreas Ericsson) #1652
- Fix typo in configure.in preventing compilation on FreeBSD (thanks sag47) #1732

Classic UI

- fix cross site scripting vulnerability in config.cgi on config expander arguments #1605
- better handling of writing to cgi.log in cmd.cgi #1161
- fixing tac.cgi header problems with counting and adding pending and descriptions #1505 #1506 #1508
- corrected behaviour of pending states in tac header #1508
- fix counting for services in tac header is wrong #1508
- remove sidebar.html inclusion in index.html causing troubles on reload #1632
- fixed: User can execute host/servicegroup commands even if not authorized for (Sven Nierlein) #1679
- fixed: plugin_output_short didn't get checked properly and caused segfault in status.cgi #1673
- fixed: submitting non existent host/service on avail.cgi causes segfault #1700
- fixed: Alert History State filter for HARD/SOFT state not working (Lennart Betz) #1722
- fixed: jquery 1.5.2 triggers memory leak in IE #1740
- fixed: ssi files don't get included if icinga path contains uppercase letters #1716

Install

- do not update start_time of already started downtimes #1658
- fix started downtime update for table scheduleddowntime in oracle #1658
- remove leftover socket file in init-script startup, e.g. from a system crash #1745
- simplify ido2db_get_object_id oracle function #1655
- fix oracle long_output column in eventhandlers missed #1748
- fix oracle ocilib 3.9.2+ fixes causing wrong 4byte->8Byte casts #1751
- fix ORA-00600 while executing merge #1362

Classic UI

- fix event handlers cmd file location in contrib #1501
- icinga.spec should not provide nagios version #1562
- don't include git meta data in upstream tarballs #1562
- fix make install-idoutils overwrites sample - adding idoutils.cfg-sample instead #1625
- fix icinga.cmd and log dir replacement in icinga-api submodule #1573 #1715
- fix DEBUG_LOG_PATH in p1.pl, will now be using configurable log-dir #1715
- do not overwrite config in make fullinstall #1754
- fix configure should tell if ocilib or libdbi if idoutils enabled in summary #1780

Config

- deprecate log_external_commands_user #1619

[More at Icinga Wiki](#)

Should you find any issues, please report them to one of the following links:

[Report Issue Icinga-Core](#)

[Report Issue Icinga-Web](#)

[Report Issue Icinga-API](#)

[Report Issue Icinga-Reporting](#)

[Report Issue Icinga-Docs](#)

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Über Icinga](#)

[Zum Anfang](#)

[Kapitel 2. Los geht's](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Kapitel 2. Los geht's

[Zurück](#)

[Weiter](#)

Kapitel 2. Los geht's

Inhaltsverzeichnis

- [Hinweise für Neulinge](#)
 - [Schnellstart-Installationsanleitungen](#)
 - [Icinga-Schnellstart auf Linux](#)
 - [Icinga-Schnellstart auf FreeBSD](#)
 - [Icinga-Schnellstart mit IDOUtils](#)
 - [Icinga-Schnellstart mit IDOUtils auf FreeBSD](#)
 - [Links zu weiteren Howtos](#)
 - [Icinga aktualisieren](#)
 - [IDOUtils-Datenbank aktualisieren](#)
 - [Windows-Maschinen überwachen](#)
 - [Linux/Unix-Rechner überwachen](#)
 - [Netware-Server überwachen](#)
 - [Netzwerk-Drucker überwachen](#)
 - [Router und Switches überwachen](#)
 - [Öffentlich zugängliche Dienste überwachen](#)
-

[Zurück](#)

[Weiter](#)

[What's New in Icinga 1.6](#)

[Zum Anfang](#)

[Hinweise für Neulinge](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Hinweise für Neulinge

[Zurück](#)[Kapitel 2. Los geht's](#)[Weiter](#)

Hinweise für Neulinge

Herzlichen Glückwunsch zur Wahl von Icinga! Icinga ist ziemlich mächtig und flexibel, aber es kann viel Arbeit bedeuten, es so zu konfigurieren, wie Sie es haben wollen. Sobald Sie damit vertraut sind, wie es funktioniert und was es für Sie tun kann, dann werden Sie nicht mehr ohne leben wollen :-). Hier sind einige wichtige Dinge, die zu beachten sind, wenn Sie zum ersten Mal Icinga benutzen:

1. **Entspannen Sie sich - es wird einige Zeit dauern.** Erwarten Sie nicht, dass alles gleich so funktioniert, wie Sie sich das vorstellen. Das Aufsetzen von Icinga kann ein bisschen an Arbeit erfordern - teilweise wegen der Optionen, die Icinga bietet, teilweise weil Sie wissen müssen, was Sie in Ihrem Netzwerk überwachen wollen (und wie das am Besten zu tun ist).
2. **Nutzen Sie die Schnellstartanleitungen.** Die [Schnellstart-Installationsanleitung](#) ist so ausgelegt, dass die meisten neuen Benutzer ziemlich schnell ein einfaches Icinga zum Laufen bekommen. Innerhalb von 20 Minuten ist Icinga installiert und überwacht Ihr lokales System. Sobald das erledigt ist, können Sie lernen, wie Icinga konfiguriert wird, um mehr zu tun.
3. **Lesen Sie die Dokumentation.** Icinga kann schwierig zu konfigurieren sein, wenn Sie ein Gespür dafür haben, was passiert, und ziemlich unmöglich, wenn Sie keins haben. Stellen Sie sicher, dass Sie die Dokumentation lesen (besonders die Abschnitte "Icinga konfigurieren" und "Die Grundlagen"). Heben Sie sich die fortgeschrittenen Themen auf, bis Sie ein gutes Verständnis der Grundlagen haben.
4. **Suchen Sie die Hilfe von anderen.** Wenn Sie die Dokumentation gelesen haben, sich die Beispiel-Konfigurationsdateien angesehen und immer noch Probleme haben, dann senden Sie eine e-Mail mit der Beschreibung Ihrer Probleme an die *Icinga-users*-Mailing-Liste. Aufgrund der Arbeit an diesem Projekt können wir die meisten der direkt an uns gesandten Fragen nicht beantworten, so dass die beste Quelle die Mailing-Liste sein dürfte. Wenn Sie bereits einiges gelesen haben und eine gute Problembeschreibung liefern, dann stehen die Chancen gut, dass jemand Ihnen Hinweise geben kann, um die Dinge zum Laufen zu bringen. Mehr (englischsprachige) Informationen, wie Sie sich den Mailing-Listen anschließen oder die Archive durchsuchen können, finden Sie unter <http://www.Icinga.org/support/>. Das deutsche Icinga-Portal finden Sie unter <http://www.Icinga-portal.de>

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Kapitel 2. Los geht's](#)

[Zum Anfang](#)

[Schnellstart-Installationsanleitungen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Schnellstart-Installationsanleitungen

[Zurück](#)[Kapitel 2. Los geht's](#)[Weiter](#)

Schnellstart-Installationsanleitungen

Einführung

Diese Schnellstartanleitungen sind dazu gedacht, Ihnen einfache Anweisungen zu liefern, wie Sie Icinga innerhalb von 20 bis 30 Minuten aus dem Quellcode installieren und Ihren lokalen Rechner damit überwachen. Hier werden keine fortgeschrittenen Installationsoptionen vorgestellt - lediglich die Grundlagen, die für 95% aller Benutzer funktionieren, die anfangen wollen.

Verweise zu Konfigurations-Tools und anderen Addons finden Sie [hier](#).



Anmerkung

Anstatt Icinga von Grund auf zu installieren möchten Sie vielleicht ein Paket benutzen, das es möglicherweise für Ihr Betriebssystem gibt. Bitte werfen Sie einen Blick auf die [Tabelle der Pakete](#).

Falls Sie aus den Sourcen installieren möchten, dann benutzen Sie bitte die offiziellen Release-Tarballs.



Wichtig

Bitte benutzen Sie keine GIT-Snapshots, solange Sie kein Problem haben, das in der aktuellen Entwicklerversion ggf. gelöst ist.

Außerdem gibt es eine neue Schnellstartanleitung für die Installation von Icinga zusammen mit den IDOUtils für die Anbindung einer MySQL-, Oracle- oder PostgreSQL-Datenbank.

Anleitungen

In den Schnellstart-Anleitungen wird auf verschiedene Linux-Distributionen eingegangen, u.a. Fedora, Ubuntu und openSuSE. Einige Distributionen sind ähnlich, so dass Sie diese Anleitungen voraussichtlich auch für RedHat, Debian und SLES verwenden können. Von Marcel Hecko stammt eine Anleitung für Icinga auf FreeBSD



Anmerkung

Bitte beachten Sie, dass sich teilweise die Namen der Pakete zwischen verschiedenen Versionen des gleichen Betriebssystems verändern. Wenn Sie also die genannten Pakete nicht finden, dann sollten Sie Ihre bevorzugte Suchmaschine nutzen, um den korrekten Namen zu ermitteln.

Die [Schnellstartanleitung für Linux](#) und die [Schnellstartanleitung für FreeBSD](#) geben Ihnen Anweisungen für die Installation von Icinga, grundlegenden Plugins, um verschiedene Dinge zu prüfen, und der klassischen GUI, auf die sie mit Ihrem Web-Browser zugreifen können. Das äußere Erscheinungsbild ist Nagios-ähnlich, obwohl die GUI verschiedene Verbesserungen bietet (z.B. erweiterter CSV-Export, Befehle gleichzeitig für mehrere Objekte, usw.). Es gibt keine Datenbank und alle Informationen werden in normalen Dateien gespeichert. Icinga-Web ist in diesem Setup *nicht* verfügbar.

Mit Hilfe der [Schnellstartanleitung für Icinga mit IDOUtils](#) und der [Schnellstartanleitung für Icinga mit IDOUtils für FreeBSD](#) bekommen Sie die o.g. Dinge sowie eine Datenbank zur Speicherung von aktuellen und historischen Informationen. Die Datenbank wird *nicht* benutzt, um die Konfiguration über ein Web-Interface zu ermöglichen. Es gibt verschiedene [Addons](#), die auf diesen Zweck spezialisiert sind und es gibt keine Pläne für ein eingebautes Tool. Das neue Icinga-Webinterface kann installiert werden, wenn Sie dieses Setup benutzen. Separate Anweisungen finden Sie [hier](#).

- [Schnellstartanleitung für Linux](#)
- [Schnellstartanleitung für FreeBSD](#)
- [Schnellstartanleitung für Icinga mit IDOUtils](#)
- [Schnellstartanleitung für Icinga mit IDOUtils für FreeBSD](#)

Wenn Sie Icinga auf einem Betriebssystem oder einer Linux-Distribution installieren, die oben nicht aufgeführt ist, lesen Sie die [Schnellstartanleitung für Linux](#), um einen Überblick zu bekommen, was zu tun ist. Befehlsnamen, Pfade und anderes variiert stark zwischen verschiedenen Betriebssystemen/Distributionen, so dass Sie wahrscheinlich die Installationsanleitung ein wenig anpassen müssen, damit sie für Ihren besonderen Fall funktioniert.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Hinweise für Neulinge](#)

[Zum Anfang](#)

[Icinga-Schnellstart auf Linux](#)



Icinga-Schnellstart auf Linux

[Zurück](#)

[Kapitel 2. Los geht's](#)

[Weiter](#)

Icinga-Schnellstart auf Linux

Einführung



Anmerkung

Anstatt Icinga von Grund auf zu installieren möchten Sie vielleicht ein Paket benutzen, das es möglicherweise für Ihr Betriebssystem gibt. Bitte werfen Sie einen Blick auf die [Tabelle der Pakete](#).

Falls Sie aus den Sourcen installieren möchten, dann benutzen Sie bitte die offiziellen Release-Tarballs.



Wichtig

Bitte benutzen Sie keine GIT-Snapshots, solange Sie kein Problem haben, das in der aktuellen Entwicklerversion ggf. gelöst ist.

Diese Schnellstartanleitung ist dazu gedacht, Ihnen einfache Anweisungen zu liefern, wie Sie Icinga innerhalb von 20 Minuten aus dem Quellcode installieren und Ihren lokalen Rechner damit überwachen.

Hier werden keine fortgeschrittenen Installationsoptionen vorgestellt - lediglich die Grundlagen, die für 95% aller Benutzer funktionieren, die anfangen wollen.

Diese Anleitung enthält momentan Anweisungen für drei verschiedene Linux-Distributionen: [Fedora](#), [Ubuntu](#) und [openSuSE](#). Ähnliche Distributionen werden wahrscheinlich auch funktionieren, darunter [RedHat](#), [CentOS](#), [Debian](#) und [SLES](#).

Wenn Sie planen, eine Datenbank zusammen mit IDOUtils zu nutzen oder wenn Sie das neue Web-Interface einsetzen möchten, dann lesen Sie statt dessen die [Schnellstartanleitung mit IDOUtils!](#)

Was dabei herauskommt

Wenn Sie diesen Anweisungen folgen, werden Sie am Ende folgendes haben:

- Icinga und die Plugins werden unterhalb von /usr/local/icinga installiert sein

- Icinga wird so konfiguriert sein, dass es einige Dinge auf Ihrem lokalen System überwacht (CPU-Auslastung, Plattenbelegung, usw.)
- das klassische Icinga-Web-Interface ist erreichbar unter `http://localhost/icinga/` oder `http://yourdomain.com/icinga/`

Voraussetzungen

Während einiger Teile der Installation benötigen Sie **root**-Zugang zu Ihrer Maschine.

Stellen Sie sicher, dass die folgenden Pakete installiert sind, bevor Sie fortfahren.

- Apache
- GCC-Compiler
- C/C++ development libraries
- [GD](#)-Development-Libraries

Optional

Zu irgendeiner Zeit möchten Sie wahrscheinlich SNMP-basierte Prüfungen verwenden, so dass es eine gute Idee ist, die benötigten Pakete gleich zu installieren. Andernfalls werden die Plugins nicht kompiliert und sind nicht verfügbar, wenn Sie diese brauchen.

Installation der Pakete

Sie können diese Pakete mit Hilfe der folgenden Befehle installieren (als root oder mit sudo).



Anmerkung

Unglücklicherweise ändern sich manchmal die Paketnamen zwischen verschiedenen Ausgaben der gleichen Distribution, so dass Sie die Suchoption Ihres Paket-Managers nutzen sollten, falls Sie die Fehlermeldung bekommen, dass eins der Pakete nicht gefunden wurde.

- `yum search <package name>` (*Fedora/RHEL/CentOS*)
- `apt-cache search <package name>` (*Debian/Ubuntu*)
- `zypper search <package name>` (*openSuSE/SLES*)

Fedora / RedHat / CentOS

```
#> yum install httpd gcc glibc glibc-common gd gd-devel
#> yum install libjpeg libjpeg-devel libpng libpng-devel
#> yum install net-snmp net-snmp-devel net-snmp-utils
```



Anmerkung

ggf. sind `libjpeg-turbo` bzw. `libjpeg-turbo-devel` zu installieren

Debian / Ubuntu

```
#> apt-get install apache2 build-essential libgd2-xpm-dev
#> apt-get install libjpeg62 libjpeg62-dev libpng12 libpng12-dev
#> apt-get install snmp libsnmp5-dev
```



Anmerkung

Die Zahlen <62/12> können je nach Distribution abweichen.



Anmerkung

Ab Ubuntu 10.10 heißt das Paket libpng12-0, der Name des dev-Pakets ändert sich nicht.

openSuSE / SLES

Bitte nutzen Sie YaST für die Installation der Pakete gd, gd-devel, libjpeg, libjpeg-devel, libpng, libpng-devel und -optional- net-snmp, net-snmp-devel und perl-Net-SNMP.

Die Nutzung von zypper sollte ebenfalls funktionieren:

```
#> zypper install gd gd-devel libjpeg libjpeg-devel libpng libpng-devel  
#> zypper install net-snmp net-snmp-devel perl-Net-SNMP
```



Anmerkung

Abhängig von der Softwareauswahl bei der Installation des Betriebssystems benötigen Sie evtl. weitere Pakete (z.B. apache2, gcc). Die devel-Pakete sind ggf. auf den SDK-DVDs zu finden.

Benutzerinformationen erstellen

Werden Sie zum root-Benutzer.

```
$> su -l
```

Erstellen Sie ein neues Benutzerkonto *icinga* und vergeben Sie ein Passwort:

```
#> /usr/sbin/useradd -m icinga  
#> passwd icinga
```

Bei einigen Distributionen müssen Sie die Gruppe in einem gesonderten Schritt anlegen:

```
#> /usr/sbin/groupadd icinga
```

Damit Sie über das klassische Web-Interface Befehle an Icinga senden können, legen Sie noch eine neue Gruppe icinga-cmd an und fügen Sie den Webbenutzer und den Icingabenutzer dieser Gruppe hinzu.

```
#> /usr/sbin/groupadd icinga-cmd  
#> /usr/sbin/usermod -a -G icinga-cmd icinga  
#> /usr/sbin/usermod -a -G icinga-cmd www-data
```

(oder www, wwwrun, apache je nach Distribution)



Anmerkung

Bei einigen usermod-Versionen (z.B. bei OpenSuSE 11 bzw. SLES 11) fehlt die Option -a. In diesen Fällen kann sie entfallen.



Anmerkung

Solaris unterstützt nur Gruppennamen bis max. 8 Zeichen, verwenden Sie icingcmd anstelle von icinga-cmd.

Icinga und die Plugins herunterladen

Wechseln Sie in Ihr lokales Source-Verzeichnis, z.B. /usr/src

```
#> cd /usr/src
```

Laden Sie die Sourcen von der [Icinga Website](#).

Vergessen Sie nicht die [Nagios-Plugins](#).

Icinga kompilieren und installieren

Entpacken Sie das Icinga-Archiv (oder wechseln Sie in den GIT Snapshot)

```
#> cd /usr/src/
#> tar xvzf icinga-1.6.tar.gz
#> cd icinga-1.6
```

Führen Sie das Icinga-configure-Script aus. Durch die Nutzung des --help-Flags erhalten Sie Hilfe zu den Optionen.

```
#> ./configure --with-command-group=icinga-cmd
```

Kompilieren Sie den Icinga-Source-Code. Um mögliche Optionen zu sehen, rufen Sie lediglich "make" auf.

```
#> make all
```

Installieren Sie die Binaries, das Init-Script, Beispiel-Konfigurationsdateien und setzen Sie die Berechtigungen für das External-Command-Verzeichnis.

```
#> make install
#> make install-init
#> make install-config
#> make install-commandmode
```

oder kürzer

```
#> make fullinstall
#> make install-config
```



Anmerkung

Ab Icinga 1.5.0 ist make install-config NICHT mehr in make fullinstall enthalten, um ein versehentliches Überschreiben der Konfigurationsdateien zu verhindern.

Die Icinga-API wird beim Aufruf von "make install" installiert, wenn Sie nur die Icinga-Api (nach)installieren möchten, nutzen Sie:

```
# make install-api
```

Die Icinga-API ist Voraussetzung für das Icinga-Web-Interface (nicht für die klassische Ansicht!).

Bitte starten Sie Icinga noch nicht - es gibt noch ein paar Dinge zu tun...

Anpassen der Konfiguration

Beispiel-Konfigurationsdateien werden durch

```
#> make install-config
```

in /usr/local/icinga/etc/ installiert. Nun fehlt nur noch eine Änderung, bevor Sie fortfahren können...

Ändern Sie die /usr/local/icinga/etc/objects/contacts.cfg-Konfigurationsdatei mit Ihrem bevorzugten Editor und passen die e-Mail-Adresse in der icingaadmin-Kontaktdefinition an, so dass sie die Adresse enthält, die im Falle von Alarmen benachrichtigt werden soll.

```
#> vi /usr/local/icinga/etc/objects/contacts.cfg
```

Installieren und konfigurieren des klassischen Web-Interface

Icinga stellt das klassische Webinterface zur Verfügung ("Classic Web", "die CGIs"). Sie können dieses wie folgt installieren:

```
#> make cgis
#> make install-cgis
#> make install-html
```

Wenn Sie (zusätzlich) das neue Icinga Web installieren wollen, lesen Sie bitte [Installation des Web-Interface](#).

Installieren Sie die Icinga-Web-Konfigurationsdatei im Apache conf.d-Verzeichnis.

```
#> make install-webconf
```

Legen Sie ein icingaadmin-Konto an, um sich am klassischen Web-Interface anmelden zu können. Merken Sie sich das Passwort, das Sie diesem Konto geben - Sie brauchen es später.

```
#> htpasswd -c /usr/local/icinga/etc/htpasswd.users icingaadmin
```



Anmerkung

Abhängig von der Apache-Version müssen Sie ggf. *htpasswd2* verwenden.

Wenn Sie das Passwort später ändern oder einen weiteren Benutzer hinzufügen möchten, verwenden Sie den folgenden Befehl:

```
#> htpasswd /usr/local/icinga/etc/htpasswd.users <USERNAME>
```

Starten Sie Apache neu, damit die Änderungen wirksam werden.

Fedora/RedHat/CentOS

```
#> service httpd restart
```

Debian / Ubuntu / openSuSE

```
#> /etc/init.d/apache2 reload
```



Anmerkung

Prüfen Sie die Implementierung der verbesserten CGI-Sicherheitsmaßnahmen wie [hier](#) beschrieben, um sicherzustellen, dass Ihre Web-Authentifizierungsinformationen nicht kompromittiert werden.

Kompilieren und installieren der Nagios-Plugins

Entpacken Sie die Nagios-Plugins-Quellcode-Archivdatei.

```
#> cd /usr/src
#> tar xzf nagios-plugins-1.4.15.tar.gz
#> cd nagios-plugins-1.4.15
```

Kompilieren und installieren Sie die Plugins.

```
#> ./configure --prefix=/usr/local/icinga \
--with-cgiurl=/icinga/cgi-bin --with-htmurl=/icinga \
--with-nagios-user=icinga --with-nagios-group=icinga
#> make
#> make install
```

Anpassen der SELinux-Einstellungen

RHEL und ähnliche Distributionen wie Fedora oder CentOS werden mit installiertem SELinux (Security Enhanced Linux) ausgeliefert und laufen im "Enforcing"-Modus. Dies kann zu "Internal Server Error"-Fehlern führen, wenn Sie versuchen, die Icinga-CGIs aufzurufen.

Schauen Sie, ob SELinux im Enforcing-Modus läuft.

```
#> getenforce
```

Setzen Sie SELinux in den "Permissive"-Modus.

```
#> setenforce 0
```

Damit diese Änderung dauerhaft wird, müssen Sie diese Einstellung in `/etc/selinux/config` anpassen und das System neustarten.

Statt SELinux zu deaktivieren oder es in den Permissive-Modus zu versetzen, können Sie den folgenden Befehl benutzen, um die CGIs im Enforcing/Targeted-Modus laufen zu lassen:

```
#> chcon -R -t httpd_sys_script_exec_t /usr/local/icinga/sbin/
#> chcon -R -t httpd_sys_content_t /usr/local/icinga/share/
#> chcon -t httpd_sys_script_rw_t /usr/local/icinga/var/rw/icinga.cmd
```

Besuchen Sie das NagiosCommunity.org-Wiki unter <http://www.nagioscommunity.org/wiki>, um Informationen darüber zu erhalten, wie die Icinga-CGIs im Enforcing-Modus mit einer Targeted-Richtlinie ausgeführt werden.

Icinga starten

Fügen Sie Icinga zu der Liste der System-Services hinzu und sorgen Sie für einen automatischen Start, wenn das System hochfährt (stellen Sie sicher, dass Sie vorher das Init-Script installiert haben).

Fedora / RedHat / CentOS / openSuSE

```
#> chkconfig --add icinga
#> chkconfig icinga on
```

Debian / Ubuntu

```
#> update-rc.d icinga defaults
```

Überprüfen Sie die Icinga-Beispielkonfigurationsdateien.

```
#> /usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
```

Anstatt die Pfade für das Binary und die Konfigurationsdatei anzugeben können Sie auch den folgenden Befehl eingeben:

```
#> /etc/init.d/icinga show-errors
```

Die Ausführung ergibt einen OK-Meldung, wenn alles in Ordnung ist, oder eine Reihe von Zeilen, die zeigen, wo der/die Fehler zu finden sind.

Wenn es dabei keine Fehler gibt, starten Sie Icinga.

Fedora / openSuSE

```
#> service icinga start
```

Debian / Ubuntu

```
#> /etc/init.d/icinga start
```

Anmelden am klassischen Web-Interface

Sie sollten nun auf das klassische Icinga-Web-Interface zugreifen können. Sie werden nach dem Benutzernamen (*icingaadmin*) und Passwort gefragt, das Sie vorhin angegeben haben.

<http://localhost/icinga/>

oder

<http://yourdomain.com/icinga/>

Klicken Sie auf den "Service Detail"-Verweis in der Navigationsleiste, um Details darüber zu erhalten, was auf Ihrer lokalen Maschine überwacht wird. Es wird ein paar Minuten dauern, bis Icinga alle mit Ihrer Maschine verbundenen Services geprüft hat, weil die Prüfungen über eine gewisse Zeit verteilt werden.

Andere Anpassungen

Stellen Sie sicher, dass die Firewall-Einstellungen Ihrer Maschine einen Zugriff auf das klassische Web-Interface ermöglichen, wenn Sie von anderen Rechnern darauf zugreifen wollen.

```
#> iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Die Konfiguration von e-Mail-Benachrichtigungen ist nicht Gegenstand dieser Anleitung. Icinga ist konfiguriert, um e-Mail-Benachrichtigungen zu versenden, aber möglicherweise ist auf Ihrem System noch kein Mail-Programm installiert bzw. konfiguriert. Schauen Sie in Ihre Systemdokumentation, suchen Sie im Web oder gucken Sie im [IcingaCommunity.org-Wiki](#) nach genauen Anweisungen, wie Ihr System konfiguriert werden muss, damit es

e-Mail-Mitteilungen an externe Adressen versendet. Mehr Informationen zu Benachrichtigungen finden Sie [hier](#).

Fertig

Glückwunsch! Sie haben erfolgreich Icinga installiert. Ihre Reise in die Überwachung hat gerade begonnen. Sie werden ohne Zweifel mehr als nur Ihre lokale Maschine überwachen wollen, so dass Sie u.a. das folgende [Kapitel](#) lesen sollten...

[Zurück](#)[Nach oben](#)[Weiter](#)[Schnellstart-Installationsanleitungen](#)[Zum Anfang](#)[Icinga-Schnellstart auf FreeBSD](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Icinga-Schnellstart auf FreeBSD

[Zurück](#)

[Kapitel 2. Los geht's](#)

[Weiter](#)

Icinga-Schnellstart auf FreeBSD

Einführung



Anmerkung

Anstatt Icinga von Grund auf zu installieren möchten Sie vielleicht ein Paket benutzen, das es möglicherweise für Ihr Betriebssystem gibt. Bitte werfen Sie einen Blick auf die [Tabelle der Pakete](#).

Falls Sie aus den Sourcen installieren möchten, dann benutzen Sie bitte die offiziellen Release-Tarballs.



Wichtig

Bitte benutzen Sie keine GIT-Snapshots, solange Sie kein Problem haben, das in der aktuellen Entwicklerversion ggf. gelöst ist.

Diese Schnellstartanleitung ist dazu gedacht, Ihnen einfache Anweisungen zu liefern, wie Sie Icinga innerhalb von 20 Minuten aus dem Quellcode installieren und Ihren lokalen Rechner damit überwachen.

Hier werden keine fortgeschrittenen Installationsoptionen vorgestellt - lediglich die Grundlagen, die für 95% aller Benutzer funktionieren, die anfangen wollen.

Diese Anleitung enthält Anweisungen für [FreeBSD 7.2](#).

Spätere Distributionen von FreeBSD werden wahrscheinlich auch mit diesen Anweisungen funktionieren.

Was dabei herauskommt

Wenn Sie diesen Anweisungen folgen, werden Sie am Ende folgendes haben:

- Icinga und die Plugins werden unterhalb von /usr/local/icinga
- Icinga wird so konfiguriert sein, dass es einige Dinge auf Ihrem lokalen System überwacht (CPU-Auslastung, Plattenbelegung, usw.)

- das klassische Icinga-Web-Interface ist erreichbar unter `http://localhost/nagios/` oder `http://yourdomain.com/icinga/`

Voraussetzungen

Während einiger Teile der Installation benötigen Sie **root**-Zugang zu Ihrer Maschine.

Stellen Sie sicher, dass die folgenden Pakete installiert sind, bevor Sie fortfahren.

- [Apache](#)
- GCC compiler
- C/C++ development libraries
- [GD](#) development libraries

Installieren Sie die Ports

Sie können diese Ports mit den folgenden Befehlen installieren (als root):

Bitte aktualisieren Sie Ihre Ports bevor Sie beginnen.

```
# cd /usr/ports/devel/libtool22/ && make deinstall && make clean && make install
# cd /usr/ports/graphics/jpeg && make deinstall && make clean && make && make install
# cd /usr/ports/graphics/png && make deinstall && make clean && make && make install
# cd /usr/ports/graphics/gd && make deinstall && make clean && make && make install
```



Anmerkung

Bitte stellen Sie sicher, dass Apache installiert ist - das Vorgehen wird hier nicht beschrieben, aber ein Hinweis ist `#> cd /usr/ports/www/apache22 && make clean && make`.

Benutzerinformationen erstellen

Werden Sie zum root-Benutzer.

```
# su -l
```

Erstellen Sie ein neues Benutzerkonto *icinga* ohne Passwort und ohne die Möglichkeit, sich anzumelden (setzen Sie kein Passwort, wenn Sie danach gefragt werden):

```
# adduser -D -w no -s nologin
```

Damit Sie über das klassische Webinterface Befehle an Icinga senden können, legen Sie noch eine neue Gruppe *icinga-cmd* an und fügen Sie den Web-Server-Benutzer (*www*) und den Icinga-Benutzer dieser Gruppe hinzu:

```
# pw groupadd -n icinga-cmd -M icinga,www
```

Icinga und die Plugins herunterladen

Wechseln Sie in Ihr lokales Source-Verzeichnis, z:b. `~/src`

```
# mkdir ~src
# cd ~src
```

Laden Sie die Sourcen von der [Icinga Website](#).

Vergessen Sie nicht die [Nagios-Plugins](#).

Icinga kompilieren und installieren

Entpacken Sie das Icinga-Archiv (oder wechseln Sie in den GIT-Snapshot)

```
# cd ~/src/
# tar xvzf icinga-1.6.tar.gz
# cd icinga-1.6
```

Führen Sie das Icinga-configure-Script aus. Durch die Nutzung des --help-Flags erhalten Sie Hilfe zu den Optionen.

```
# ./configure --with-httdp-conf=/usr/local/etc/apache22/Includes/ \
--with-gd-lib=/usr/local/lib/ \
--with-gd-inc=/usr/local/include/ \
--with-command-group=icinga-cmd
```

Kompilieren Sie den Icinga-Source-Code. Um mögliche Optionen zu sehen, rufen Sie lediglich "make" auf.

```
# make all
```

Installieren Sie die Binaries, das Init-Script, Beispiel-Konfigurationsdateien und setzen Sie die Berechtigungen für das External-Command-Verzeichnis.

```
# make install
# make install-init
# make install-config
# make install-commandmode
```

oder kürzer

```
# make fullinstall
# make install-config
```



Anmerkung

Ab Icinga 1.5.0 ist `make install-config` NICHT mehr in `make fullinstall` enthalten, um ein versehentliches Überschreiben der Konfigurationsdateien zu verhindern.

Die Icinga-API wird beim Aufruf von "make install" installiert, wenn Sie nur die Icinga-API (nach)installieren möchten, nutzen Sie:

```
# make install-api
```

Die Icinga-API ist Voraussetzung für das Icinga Web-Interface (nicht für die klassische Ansicht!).

Starten Sie Icinga noch nicht - es gibt noch ein paar Dinge zu tun...

Anpassen der Konfiguration

Beispiel-Konfigurationsdateien werden durch

```
# make install-config
```

in /usr/local/icinga/etc/ installiert. Nun fehlt nur noch eine Änderung, bevor Sie fortfahren können...

Ändern Sie die /usr/local/icinga/etc/objects/contacts.cfg-Konfigurationsdatei mit Ihrem bevorzugten Editor und passen die e-Mail-Adresse in der icingaadmin-Kontaktdefinition an, so dass sie die Adresse enthält, die im Falle von Alarmen benachrichtigt werden soll.

```
# vi /usr/local/icinga/etc/objects/contacts.cfg
```

Installieren und konfigurieren des klassischen Web-Interface

Icinga stellt das klassische Web-Interface zur Verfügung ("Classic Web", "die CGIs"). Sie können dieses wie folgt installieren:

```
#> make cgis
#> make install-cgis
#> make install-html
```

Wenn Sie (zuätzlich) das neue Icinga Web installieren wollen, lesen Sie bitte [Installation des Web-Interface](#).

Installieren Sie die Icinga-Web-Konfigurationsdatei im Apache-Konfigurationsverzeichnis.



Anmerkung

Es gibt momentan einen Bug im Icinga-Makefile, der die Ausführung dieses *make*-Befehls unter FreeBSD verhindert, daher editieren Sie die Makefile-Datei im Icinga-Source-Verzeichnis und ändern Sie die Zeile

```
$(INSTALL) -D -m 644 sample-config/httpd.conf $(DESTDIR)$(HTTPD_CONF)/icinga.conf
```

in

```
$(INSTALL) -m 644 sample-config/httpd.conf $(DESTDIR)$(HTTPD_CONF)/icinga.conf
```

```
# make install-webconf
```

Legen Sie ein *icingaadmin*-Konto an, um sich am klassischen Web-Interface anmelden zu können. Merken Sie sich das Passwort, das Sie diesem Konto geben - Sie brauchen es später.

```
# htpasswd -c /usr/local/icinga/etc/htpasswd.users icingaadmin
```

Wenn Sie das Passwort später ändern oder einen weiteren Benutzer hinzufügen möchten, verwenden Sie den folgenden Befehl:

```
# htpasswd /usr/local/icinga/etc/htpasswd.users <USERNAME>
```

Starten Sie Apache neu, damit die Änderungen wirksam werden.

```
# /usr/local/etc/rc.d/apache2 reload
```

Kompilieren und installieren der Nagios-Plugins

Entpacken Sie die Nagios-Plugins-Quellcode-Archivdatei.

```
# cd ~/src
# tar xvzf nagios-plugins-1.4.15.tar.gz
# cd nagios-plugins-1.4.15
```

Kompilieren und installieren Sie die Plugins, indem Sie das Installationverzeichnis auf `/usr/local/icinga`

```
# ./configure --prefix=/usr/local/icinga \
--with-cgiurl=/icinga/cgi-bin --with-htmurl=/icinga \
--with-nagios-user=icinga --with-nagios-group=icinga
# make
# make install
```

Icinga starten

Fügen Sie Icinga zur Liste der System-Services hinzu, damit es automatisch beim Start des Systems gestartet wird (stellen Sie sicher, dass Sie das Init-Script vorher installiert haben).

```
# echo icinga_enable=\"YES\" >> /etc/rc.conf
```

Überprüfen Sie die Icinga-Konfigurationsdateien.

```
# /usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
```

Wenn es dabei keine Fehler gibt, starten Sie Icinga.

```
# /usr/local/etc/rc.d/icinga start
```

Anmelden am klassischen Web-Interface

Sie sollten nun auf das klassische Icinga-Web-Interface zugreifen können. Sie werden nach dem Benutzernamen (*icingaadmin*) und Passwort gefragt, das Sie vorhin angegeben haben.

<http://localhost/icinga/>

oder

<http://yourdomain.com/icinga/>

Klicken Sie auf den "Service Detail"-Verweis in der Navigationsleiste, um Details darüber zu erhalten, was auf Ihrer lokalen Maschine überwacht wird. Es wird ein paar Minuten dauern, bis Icinga alle mit Ihrer Maschine verbundenen Services geprüft hat, weil die Prüfungen über eine gewisse Zeit verteilt werden.

Andere Modifikationen

Stellen Sie sicher, dass die Firewall-Einstellungen Ihrer Maschine einen Zugriff auf das klassische Web-Interface ermöglichen, wenn Sie von anderen Rechnern darauf zugreifen wollen.

```
# TCP port 80
```

Die Installation eines Mail Transfer Agent (MTA) wie exim, sendmail oder postfix ist nicht Gegenstand dieser Anleitung. Icinga ist konfiguriert, um e-Mail-Benachrichtigungen zu versenden, aber möglicherweise ist auf Ihrem System noch kein Mail-Programm installiert bzw. konfiguriert. Schauen Sie in Ihre Systemdokumentation oder suchen Sie im Web nach weiteren Informationen.

Fertig

Glückwunsch! Sie haben erfolgreich Icinga installiert. Ihre Reise in die Überwachung hat gerade begonnen.

Sie werden ohne Zweifel mehr als nur Ihre lokale Maschine überwachen wollen, so dass Sie u.a. das folgende [Kapitel](#) lesen sollten...

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Icinga-Schnellstart auf Linux](#)

[Zum Anfang](#)

[Icinga-Schnellstart mit IDOUtils](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Icinga-Schnellstart mit IDOUtils

[Zurück](#)

[Kapitel 2. Los geht's](#)

[Weiter](#)

Icinga-Schnellstart mit IDOUtils

Einführung



Anmerkung

Anstatt Icinga von Grund auf zu installieren möchten Sie vielleicht ein Paket benutzen, das es möglicherweise für Ihr Betriebssystem gibt. Bitte werfen Sie einen Blick auf die [Tabelle der Pakete](#).

Falls Sie aus den Sourcen installieren möchten, dann benutzen Sie bitte die offiziellen Release-Tarballs.



Wichtig

Bitte benutzen Sie keine GIT-Snapshots, solange Sie kein Problem haben, das in der aktuellen Entwicklerversion ggf. gelöst ist.

Diese Schnellstartanleitung ist dazu gedacht, Ihnen einfache Anweisungen zu liefern, wie Sie Icinga innerhalb von 30 Minuten aus dem Quellcode installieren und Ihren lokalen Rechner damit überwachen.

Hier werden keine fortgeschrittenen Installationsoptionen vorgestellt - lediglich die Grundlagen, die für 95% aller Benutzer funktionieren, die anfangen wollen.

Diese Anleitung enthält Anweisungen für drei verschiedene Linux-Distributionen: [Fedora](#), [Ubuntu](#) und [openSuSE](#). Ähnliche Distributionen werden wahrscheinlich auch funktionieren, darunter [RedHat](#), [CentOS](#), [Debian](#) und [SLES](#).

Wenn Sie planen, Icinga OHNE IDOUtils zu nutzen, dann lesen Sie statt dessen „[Icinga-Schnellstart auf Linux](#)“!

Was dabei herauskommt

Wenn Sie diesen Anweisungen folgen, werden Sie am Ende folgendes haben:

- Icinga und die Plugins werden unterhalb von /usr/local/icinga installiert sein
- Icinga wird so konfiguriert sein, dass es einige Dinge auf Ihrem lokalen System überwacht (CPU-Auslastung, Plattenbelegung, usw.)

- Das klassische Icinga-Web-Interface ist erreichbar unter <http://localhost/icinga/>
- Eine Datenbank, die von Icinga mit Hilfe von IDOUtils gefüllt wird

Voraussetzungen

Während einiger Teile der Installation benötigen Sie **root**-Zugang zu Ihrer Maschine.

IDOUtils benutzt die [libdbi](#) und die libdbi-Treiber für verschiedene Datenbanken. Die Development-Libraries werden ebenfalls benötigt. Die folgenden Beispiele zeigen, wie die IDOUtils mit libdbi für MySQL oder PostgreSQL installiert werden.

Stellen Sie sicher, dass die folgenden Pakete installiert sind, bevor Sie fortfahren.

- Apache
- GCC-Compiler
- C/C++ development libraries
- [GD](#)-Development-Libraries
- libdbi/libdbi-Treiber, eine Datenbank wie z.B. MySQL oder PostgreSQL



Anmerkung

PostgreSQL: Aufgrund von Änderungen an Insert-Statements funktioniert PostgreSQL 8.1 nicht länger (und ist auch bereits "End-of-life" seit November 2010), also benutzen Sie bitte 8.2 oder besser noch 8.4. String escaping mit PostgreSQL 9.x befindet sich noch in einem experimentellen Stadium.

Optional

Zu irgendeiner Zeit möchten Sie wahrscheinlich SNMP-basierte Prüfungen verwenden, so dass es eine gute Idee ist, die benötigten Pakete gleich zu installieren. Andernfalls werden die Plugins nicht kompiliert und sind nicht verfügbar, wenn Sie diese brauchen.

Neue Features für die IDOUtils:

SSL-Verschlüsselung zwischen idomod und ido2db

Wenn Sie **SSL-Verschlüsselung** verwenden möchten, werden zusätzlich die Pakete openssl und openssl-devel/libssl-dev benötigt!



Anmerkung

SSL muss auch bei allen idomod-Clients aktiviert werden, sonst gehen Daten verloren!!!

Oracle-Datenbankunterstützung

Wenn Sie Oracle als RDBMS einsetzen möchten, müssen Sie installieren:

- die Oracle Libraries und SDK (zum Beispiel den Oracle Instantclient)

- installieren Sie statt des libdbi-Treibers den ocilib-Treiber



Anmerkung

Zum Einsatz von IDOUtils 1.5.0 mit Oracle benötigen Sie OCILIB 3.9.2 - installieren Sie nicht 3.9.0 oder 3.9.1, denn diese Version sind fehlerbehaftet.



Anmerkung

Detailliertere Informationen zu IDOUtils mit Oracle finden Sie in unserer Wiki-HowTo-Serie zu [Oracle](#).

Stellen Sie sicher, dass die Libraries in der Path-Variablen enthalten sind. Oder setzen Sie die den Oracle Library Path mit --with-oracle-lib=/path/to/instantclient

Download des ocilib-Treiber von <http://orclib.sourceforge.net/> und verweisen Sie beim configure auf Ihre Oracle-Libraries und die Header-Files, z.B. die des Oracle Instant-Client:

```
#> ./configure --with-oracle-headers-path=/path/to/instantclient/sdk/include \
  --with-oracle-lib-path=/path/to/instantclient/
#> make
#> make install
```

Icinga 1.4

Seit Icinga 1.4 wird mindestens Oracle 10gR2 benötigt. Ältere Versionen funktionieren möglicherweise, werden aber nicht unterstützt. Die Oracle-Skripte wurden geändert, um für Daten, Indexe und LOBs nun unterschiedliche Tablespace zu nutzen. Aus diesem Grund gibt es die Notwendigkeit, den Tablespace-Namen anzugeben, den Sie benutzen möchten. Wenn Sie eine kleine Umgebung haben, dann können Sie für alle "defines" den gleichen Tablespace angeben. Sie müssen das neue Skript `icinga_defines.sql` auf Ihre Bedürfnisse anpassen, bevor Sie das Skript `oracle.sql` ausführen. Um Ihnen ein wenig Arbeit abzunehmen, gibt es ein neues Skript `create_oracle_sys.sql`, das Ihnen helfen soll, die benötigten Tablespace und einen Icinga-Applikations-Benutzer anzulegen, das als SYS ausgeführt werden muss. Es benutzt außerdem `icinga_defines.sql`. Die Erzeugung von Objekten wurde von `oracle.sql` in das Skript `create_icinga_objects_oracle.sql` verlagert. Das alte `oracle.sql` wurde in ein "Master"-Skript umgewandelt und enthält per "include" Verweise auf alle anderen Skripte, die im aktuellen Verzeichnis erwartet werden. Aus diesem Grund sollten Sie `sqlplus` in diesem Verzeichnis starten, um `oracle.sql` auszuführen. Auf diese Weise erfolgt die Erzeugung von Benutzer und Tablespace sowie die Erzeugung der Icinga-Tabellen in einem Schritt. Als ein All-in-one-Beispiel gibt es das neue Skript `db/scripts/create_oracledb.sh`. Passen Sie die Variablen auf Ihre Bedürfnisse an und freuen Sie sich. Wenn Sie es vorziehen, die Schritte als SYS selbst zu erledigen, dann entfernen Sie den Kommentar vor `create_oracle_sys.sql` und stellen Sie sicher, dass Ihr Icinga-Datenbankbenutzer und die Tabellen existieren und (mindestens) mit den gleichen Rechten definiert sind und dass die korrekten Einstellungen in `icinga_defines.sql` vorhanden sind.

Zeitzonenunterstützung

Ab Icinga 1.6 werden alle Zeit-/Datumsinformationen als lokale Zeitstempel in der Datenbank gespeichert. Vorher gab es für jedes Datenbanksystem unterschiedliche Vorgehensweisen beim Speichern und Abrufen der Daten, was schwierig für Frontend-Applikationen war. Deshalb wurden die Datentypen geändert: für MySQL von DATETIME in TIMESTAMP und für Oracle von DATE in LOCAL TIMESTAMP (0). PostgreSQL benutzt bereits TIMESTAMP, es wurde aber in TIMESTAMP WITH TIME ZONE geändert. IDO2DB wird die Sitzungszeitzone auf UTC

setzen und alle Unix-Timestamps (die per Definition UTC sind) als UTC-basierte Werte speichern. Bitte stellen Sie sicher, dass Ihr System Unix-Timestamps als UTC-basierte Werte ausgibt (wie "date -u '+%s'").



Anmerkung

Stellen Sie sicher, dass Ihre Datenbanksitzung in der gleichen Zeitzone läuft, in der die existierenden Zeit-/Datumsinformationen gespeichert wurden (überprüfen Sie Ihre lokale Zeitzone, z.B. Oracle: "select sessiontimezone from dual;"), wenn Sie das Upgrade-Script laufen lassen. Zusätzlich für Ihre Bequemlichkeit sollten Sie in Oracle das Format der Sitzungszeitstempel auf den Wert zu setzen, den Sie wünschen, z.B. "alter session set nls_timestamp_format='YYYY-MM-DD HH24:MI:SS'; oder etwas ähnliches. Nun können Sie Einträge von verschiedenen Icinga-Installationen in unterschiedlichen Zeitzonen ansehen, die in der gleichen Datenbank gespeichert wurden, und unabhängig davon, von woher Sie darauf zugreifen, werden sie in Ihrer lokalen Zeitzone angezeigt.

Installation der Pakete

Sie können diese Pakete mit Hilfe der folgenden Befehle installieren (als root oder mit sudo).



Anmerkung

Unglücklicherweise ändern sich manchmal die Paketnamen zwischen verschiedenen Ausgaben der gleichen Distribution, so dass Sie die Suchoption Ihres Paket-Managers nutzen sollten, falls Sie die Fehlermeldung bekommen, dass eins der Pakete nicht gefunden wurde.

- yum search <package name> (*Fedora/RHEL/CentOS*)
- apt-cache search <package name> (*Debian/Ubuntu*)
- zypper search <package name> (*openSUSE/SLES*)

Fedora / RedHat / CentOS:

```
#> yum install httpd gcc glibc glibc-common gd gd-devel
#> yum install libjpeg libjpeg-devel libpng libpng-devel
```



Anmerkung

ggf. sind libjpeg-turbo bzw. libjpeg-turbo-devel zu installieren

- **MySQL:**

```
#> yum install mysql mysql-server libdbi libdbi-devel libdbi-drivers libdbi-dbd-mysql
```

- **PostgreSQL:**

```
#> yum install postgresql postgresql-server libdbi libdbi-devel libdbi-drivers libdbi-dbd-psql
```

Debian / Ubuntu:

```
#> apt-get install apache2 build-essential libgd2-xpm-dev
#> apt-get install libjpeg62 libjpeg62-dev libpng12 libpng12-dev
```



Anmerkung

Die Zahlen <62/12> können je nach Distribution abweichen



Anmerkung

Ab Ubuntu 10.10 heißt das Paket libpng12-0, der Name des dev-Pakets ändert sich nicht.

- **MySQL:**

```
#> apt-get install mysql-server mysql-client libdbi0 libdbi0-dev libdbd-mysql
```

- **PostgreSQL:**

```
#> apt-get install postgresql libdbi0 libdbi-dev libdbd-pgsql libpq-dev
```

openSuSE:

Bitte nutzen Sie YaST für die Installation der Pakete gd, gd-devel, libjpeg, libjpeg-devel, libpng, libpng-devel und -optional- net-snmp, net-snmp-devel und perl-Net-SNMP.

Die Nutzung von zypper sollte ebenfalls funktionieren:

```
#> zypper install gd gd-devel libjpeg libjpeg-devel libpng libpng-devel  
#> zypper install net-snmp net-snmp-devel perl-Net-SNMP
```



Anmerkung

Die devel-Pakete sind ggf. auf den SDK-DVDs zu finden.

- **MySQL:**

Benutzen Sie yast zur Installation der Pakete für das RDBMS, das Sie verwenden möchten, also "mysql", "mysql-devel" sowie die libdbi-Pakete "libdbi", "libdbi-devel", "libdbi-drivers" und "libdbi-dbd-mysql" oder alternativ zypper

```
#> zypper install mysql mysql-devel libdbi libdbi-devel libdbi-drivers libdbi-dbd-mysql
```

Bei OpenSuSE 11 (SLES 11) lautet der Name des Packages statt "mysql-devel" nun "libmysqlclient-devel".

- **PostgreSQL:**

Benutzen Sie yast zur Installation der Pakete für das RDBMS, das Sie verwenden möchten, also "postgresql", "postgresql-devel" und "postgresql-server" sowie die libdbi-Pakete "libdbi", "libdbi-devel" und "libdbi-drivers" oder alternativ zypper>.

```
#> zypper install postgresql postgresql-devel postgresql-server  
#> zypper install libdbi libdbi-devel libdbi-drivers libdbi-dbd-mysql
```

Bei alten OpenSuSE- (SLES) Version einschließlich Version 10 ist es ziemlich wahrscheinlich, dass es keine libdbi-Packages gibt, so dass Sie die Sourcen herunterladen und kompilieren müssen. Ersetzen Sie dabei <rdbm> durch Ihr RDBM wie mysql oder pgsql. Bitte beachten Sie, dass der Oracle-Treiber noch nicht funktioniert. Lesen Sie daher den entsprechenden Abschnitt zu ocilib (anstatt libdbi).

1. Laden Sie die tar.gz-Dateien herunter und entpacken Sie diese

<http://libdbi.sourceforge.net/download.html>

<http://libdbi-drivers.sourceforge.net/download.html>

```
#> tar xvzf libdbi-0.8.3.tar.gz
#> tar xvzf libdbi-drivers-0.8.3-1.tar.gz
```

2. Installieren Sie die libdbi. Möglicherweise brauchen Sie beim configure weitere Optionen (set --prefix=/usr ...)

```
#> cd libdbi-0.8.3
#> ./configure --disable-docs
#> make
#> make install
```

3. Installieren Sie die libdbi-Treiber

```
#> cd libdbi-drivers-0.8.3-1
#> ./configure --with-<rdbm> --disable-docs
#> make
#> make install
```



Anmerkung

Bei den 64-bit-Versionen müssen Sie die Pfade zu den include- und lib-dir-Verzeichnissen explizit angeben:

```
#> ./configure --with-<rdbm> \
--with-<rdbm>-incdir=/usr/include/<rdbm>/ \
--with-<rdbm>-libdir=/usr/lib64/ --disable-docs
```

Benutzerinformationen erstellen

Werden Sie zum root-Benutzer.

```
$> su -l
```

Erstellen Sie ein neues Benutzerkonto *icinga* und vergeben Sie ein Passwort:

```
#> /usr/sbin/useradd -m icinga
#> passwd icinga
```

Bei einigen Distributionen müssen Sie die Gruppe in einem gesonderten Schritt anlegen:

```
#> /usr/sbin/groupadd icinga
```

Damit Sie über das klassische Web-Interface Befehle an Icinga senden können, legen Sie noch eine neue Gruppe *icinga-cmd* an und fügen Sie den Webbenutzer und den Icingabenutzer dieser Gruppe hinzu.

```
#> /usr/sbin/groupadd icinga-cmd
#> /usr/sbin/usermod -a -G icinga-cmd icinga
#> /usr/sbin/usermod -a -G icinga-cmd www-data
```

(oder www, wwwrun, apache je nach Distribution)



Anmerkung

Bei einigen usermod-Versionen (z.B. OpenSuSE 11 bzw. SLES 11) fehlt die Option -a. In diesen Fällen kann sie entfallen.



Wichtig

Solaris unterstützt nur Gruppennamen bis max. 8 Zeichen, verwenden Sie icingcmd anstelle von icinga-cmd.

Icinga und die Plugins herunterladen

Wechseln Sie in Ihr lokales Source-Verzeichnis, z.B. /usr/src

```
#> cd /usr/src
```

Laden Sie die Sourcen von der [Icinga Website](#).

Vergessen Sie nicht die [Nagios Plugins](#).

Icinga und die IDOUtils kompilieren und installieren

Entpacken Sie das Icinga-Archiv

```
#> cd /usr/src/
#> tar xvzf icinga-1.6.tar.gz
#> cd icinga-1.6
```



Anmerkung

Dieser absolute Pfad ist gemeint, wenn im Nachfolgenden von '/path/to/icinga-src/' die Rede ist.

Führen Sie das Icinga-configure-Script aus. Durch die Nutzung des --help-Flags erhalten Sie Hilfe zu den Optionen.

```
#> ./configure --with-command-group=icinga-cmd --enable-idoutils
```



Wichtig

Das Kompilieren auf Solaris kann wegen unerfüllten Bibliotheksabhängigkeiten von gethostbyname fehlschlagen. Wenn dies der Fall ist, führen Sie folgenden Befehl vor configure aus:

```
#> export LIBS=-lsocket -lnsl
```



Anmerkung

Auch mit **PostgreSQL** sollten Sie nicht den Parameter --enable-pgsql verwenden, denn dieser wird noch nicht unterstützt.

Mit SSL-Verschlüsselung:

```
#> ./configure --with-command-group=icinga-cmd --enable-idoutils --enable-ssl
```

Mit Oracle-Datenbankunterstützung:

```
#> ./configure --with-command-group=icinga-cmd \
--enable-idoutils --enable-oracle
```

Wenn Ihre Oracle Libraries nicht in der Path-Variablen enthalten sind, können Sie sie im configure angeben:

```
#> ./configure --with-command-group=icinga-cmd \
--enable-idoutils --enable-oracle \
--with-oracle-lib=/path/to/instantclient
```

Wenn Sie die ocilib nicht im Standardpfad (/usr/local) installiert haben, können Sie configure die lib/inc Verzeichnisse angeben:

```
#> ./configure --with-command-group=icinga-cmd \
--enable-idoutils --enable-oracle \
--with-ocilib-lib=/path/to/ocilib/lib --with-ocilib-inc=/path/to/ocilib/include
```



Anmerkung

Wenn Sie von einer Oracle-Datenbank auf ein anderes RDBMS wechseln möchten, dann müssen Sie die IDOUtils erneut kompilieren und installieren!

```
#> make distclean
#> ./configure --enable-idoutils
```

Kompilieren und Installieren

Kompilieren Sie den Icinga-Source-Code. Es gibt auch eine extra Option für IDOUtils (*make idoutils*), wenn Sie nur dieses Module erneut kompilieren möchten. Um mögliche Optionen zu sehen, rufen Sie lediglich "make" auf.

```
#> make all
```

Installieren Sie die Binaries, das Init-Script, Beispiel-Konfigurationsdateien und setzen Sie die Berechtigungen für das External-Command-Verzeichnis.

```
#> make install
#> make install-init
#> make install-config
#> make install-commandmode
#> make install-idoutils
```

oder kürzer

```
#> make fullinstall
#> make install-config
```



Anmerkung

Ab Icinga 1.5.0 ist `make install-config` NICHT mehr in `make fullinstall` enthalten, um ein versehentliches Überschreiben der Konfigurationsdateien zu verhindern.



Anmerkung

Installieren Sie die IDOUtils und andere Ereignis-Broker-Module nur mit dem primären Ziel **make install**. Manuelles Kopieren und Überschreiben des vorhandenen Moduls erzeugt einen Segfault des Icinga Kerns mit Hilfe von idomod.o, da eine Verwendung einer temporären Kopie explizit verhindert werden soll. Dies ist nützlich für [OMD](#)



Anmerkung

Die Icinga-API diente als eine in PHP geschriebene Datenbank-Abstraktionsschicht. Seit Icinga ist sie durch eine interne Datenbank-Abstraktionsschicht ersetzt worden, so dass Icinga-API nicht mehr installiert werden muss.

Bitte starten Sie Icinga noch nicht - es gibt noch ein paar Dinge zu tun...

Anpassen der Konfiguration

Beispiel-Konfigurationsdateien werden durch

```
#> make install-config
```

in /usr/local/icinga/etc/ installiert.

Ändern Sie die */usr/local/icinga/etc/objects/contacts.cfg*-Konfigurationsdatei mit Ihrem bevorzugten Editor und passen die e-Mail-Adresse in der *icingaadmin*-Kontaktdefinition an, so dass sie die Adresse enthält, die im Falle von Alarmen benachrichtigt werden soll.

```
#> vi /usr/local/icinga/etc/objects/contacts.cfg
#> cd /usr/local/icinga/etc
#> mv idomod.cfg-sample idomod.cfg
#> mv ido2db.cfg-sample ido2db.cfg
```

Wenn Sie die IDOUtils mit ssl kompiliert haben, aktivieren Sie ssl in der *idomod.cfg* mit

```
use_ssl=1
output_type=tcpsocket
output=127.0.0.1
```

(Passen Sie die IP-Adresse an, wenn sich Ihre Datenbank nicht auf localhost befindet!) und der *ido2db.cfg* mit

```
use_ssl=1
socket_type=tcp
```



Anmerkung

Vergessen Sie nicht, alle anderen idomod-Clients auch neu zu kompilieren und auf ssl umzustellen, **anderenfalls werden Sie Daten verlieren!!!**

Aktivieren des idomod-Eventbroker-Moduls



Wichtig

Unter normalen Umständen gibt es im `modules`-Unterverzeichnis bereits die folgende module-Definition, so dass Sie keine Änderungen an der Hauptkonfigurationsdatei vornehmen müssen.

```
define module{
    module_name      ido_mod
    path            /usr/local/icinga/bin/idomod.o
    module_type     neb
    args           config_file=/usr/local/icinga/etc/idomod.cfg
}
```

Das bedeutet auch, dass ein entsprechender `broker_module`-Eintrag nicht aktiviert werden darf!

- **MySQL:**

Anlegen und konfigurieren von Datenbank, Benutzer und Berechtigungen



Anmerkung

Wenn Sie wie weiter oben beschrieben das Datenbanksystem neu installiert haben, dann müssen Sie den Datenbank-Server starten, bevor Sie eine Datenbank anlegen können. Im Falle von MySQL erfolgt der Start z.B. mit `/etc/init.d/mysqld start` (oder `/etc/init.d/mysql`, abhängig von Ihrer Distribution).

```
#> mysql -u root -p
mysql> CREATE DATABASE icinga;

GRANT USAGE ON *.* TO 'icinga'@'localhost'
IDENTIFIED BY 'icinga'
WITH MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0
MAX_UPDATES_PER_HOUR 0;

GRANT SELECT , INSERT , UPDATE , DELETE, DROP, CREATE VIEW
ON icinga.* TO 'icinga'@'localhost';

FLUSH PRIVILEGES ;

quit
```



Anmerkung

Ab Icinga 1.6 benötigt der icinga-User zusätzlich die Rechte `DROP` und `CREATE VIEW`.

```
#> cd /path/to/icinga-src/module/idouutils/db/mysql
#> mysql -u root -p icinga < mysql.sql

#> vi /usr/local/icinga/etc/ido2db.cfg

db_servertype=mysql
db_port=3306
db_user=icinga
db_pass=icinga
```

- **PostgreSQL:**

Anlegen von Datenbank und Benutzer

```
#> su - postgres
$ createlang plpgsql icinga;
$ psql
postgres=# CREATE USER icinga;
postgres=# ALTER USER icinga WITH PASSWORD 'icinga';
postgres=# CREATE DATABASE icinga;
```

Debian:

```
#> vi /etc/postgresql/8.x/main/pg_hba.conf
```

Fedora / RedHat / CentOS:

```
#> vi /var/lib/pgsql/data/pg_hba.conf
```

Editieren Sie die Konfiguration z.B. wie folgt (dem lokalen Benutzer muss vertraut werden)

```
# database administrative login by UNIX sockets
local    all      postgres          ident
# TYPE   DATABASE   USER      CIDR-ADDRESS   METHOD
#icinga
local    icinga    icinga           trust
# "local" is for Unix domain socket connections only
local    all      all              trust
# IPv4 local connections
host     all      all      127.0.0.1/32   trust
# IPV6 local connections
host     all      all      ::1/128       trust
```

Neuladen und konfigurieren des Datenbankschemas

```
#> /etc/init.d/postgresql-8.x reload
#> cd /path/to/icinga-src/module/idouutils/db/pgsql/
#> psql -U icinga -d icinga < pgsql.sql
```

Editieren der DB-Konfigurationsdatei, um die IDOUtils anzupassen

```
#> vi /usr/local/icinga/etc/ido2db.cfg
db_servertype=pgsql
db_port=5432
db_user=icinga
db_pass=icinga
```

- **Oracle:**

Erstellen Sie ein Datenbank-Schema und eine username/password-Kombination (lesen Sie dazu die Oracle-Dokumentation unter <http://www.oracle.com> oder fragen Sie Ihren DBA). Importieren Sie das Datenbank-Schema mit sqlplus (oder Ihrer bevorzugten Methode). Kopieren Sie module/idouutils/db/oracle/* nach \$ORACLE_HOME und ändern Sie in icinga_defines.sql die Werte zu Tablespace-Namen und Benutzerdaten.

```
#> su - oracle
$ sqlplus dbuser/dbpass
SQL> @oracle.sql
```

Editieren Sie das DB-Config-File, um die IDOUtils anzupassen. Denken Sie daran, dass Oracle den DB-Host ignoriert, nutzen Sie statt dessen db_name, um auf //DBSERVER/DBNAME zu verweisen

```
#> vi /usr/local/icinga/etc/ido2db.cfg
db_servertype=oracle
db_port=1521
db_user=icinga
db_pass=icinga
```

Installieren und konfigurieren des klassischen Web-Interface

Icinga stellt das klassische Web-Interface zur Verfügung ("Classic Web", "die CGIs"). Sie können dieses wie folgt installieren:

```
#> cd /path/to/icinga-src
#> make cgis
#> make install-cgis
#> make install-html
```

Wenn Sie (zusätzlich) das neue Icinga-Web installieren wollen, lesen Sie bitte [Installation des Web-Interface](#).

Installieren Sie die Icinga-Web-Konfigurationsdatei im Apache conf.d-Verzeichnis.

```
#> cd /path/to/icinga-src
#> make install-webconf
```

Legen Sie ein *icingaadmin*-Konto an, um sich am klassischen Web-Interface anmelden zu können. Merken Sie sich das Passwort, das Sie diesem Konto geben - Sie brauchen es später.

```
#> htpasswd -c /usr/local/icinga/etc/htpasswd.users icingaadmin
```



Anmerkung

Abhängig von der Apache-Version müssen Sie ggf. *htpasswd2* verwenden.

Wenn Sie das Passwort später ändern oder einen weiteren Benutzer hinzufügen möchten, verwenden Sie den folgenden Befehl:

```
#> htpasswd /usr/local/icinga/etc/htpasswd.users <USERNAME>
```

Starten Sie Apache neu, damit die Änderungen wirksam werden.

Fedora / RHEL / CentOS:

```
#> service httpd restart
```

Ubuntu / openSuSE:

```
#> service apache2 restart
```

Debian:

```
#> /etc/init.d/apache2 reload
```



Anmerkung

Prüfen Sie die Implementierung der verbesserten CGI-Sicherheitsmaßnahmen wie [hier](#) beschrieben, um sicherzustellen, dass Ihre Web-Authentifizierungsinformationen nicht kompromittiert werden.

Kompilieren und installieren der Nagios-Plugins

Entpacken Sie die Nagios-Plugins-Quellcode-Archivdatei.

```
#> cd /usr/src
#> tar xzf nagios-plugins-1.4.15.tar.gz
#> cd nagios-plugins-1.4.15
```

Kompilieren und installieren Sie die Plugins.

```
#> ./configure --prefix=/usr/local/icinga --with-cgiurl=/icinga/cgi-bin \
--with-htmurl=/icinga --with-nagios-user=icinga --with-nagios-group=icinga
#> make
#> make install
```

Anpassen der SELinux-Einstellungen

RHEL und ähnliche Distributionen wie Fedora oder CentOS werden mit installiertem SELinux (Security Enhanced Linux) ausgeliefert und laufen im "Enforcing"-Modus. Dies kann zu "Internal Server Error"-Fehlern führen, wenn Sie versuchen, die Icinga-CGIs aufzurufen.

Schauen Sie, ob SELinux im Enforcing-Modus läuft.

```
#> getenforce
```

Setzen Sie SELinux in den "Permissive"-Modus.

```
#> setenforce 0
```

Damit diese Änderung dauerhaft wird, müssen Sie diese Einstellung in */etc/selinux/config* anpassen und das System neustarten.

Statt SELinux zu deaktivieren oder es in den Permissive-Modus zu versetzen, können Sie den folgenden Befehl benutzen, um die CGIs im Enforcing/Targeted-Modus laufen zu lassen:

```
#> chcon -R -t httpd_sys_script_exec_t /usr/local/icinga/sbin/
#> chcon -R -t httpd_sys_content_t /usr/local/icinga/share/
#> chcon -t httpd_sys_script_rw_t /usr/local/icinga/var/rw/icinga.cmd
```

Besuchen Sie das NagiosCommunity.org-Wiki unter <http://www.nagioscommunity.org/wiki>, um Informationen darüber zu erhalten, wie die Icinga-CGIs im Enforcing-Modus mit einer Targeted-Richtlinie ausgeführt werden.

IDOUtils und Icinga starten

IDOUtils muss vor Icinga gestartet werden

IDOUtils starten

Fedora / openSuSE/Ubuntu:

```
#> service ido2db start
```

Debian:

```
#> /etc/init.d/ido2db start
```

IDOUtils beenden

Fedora / openSuSE/Ubuntu:

```
#> service ido2db stop
```

Debian:

```
#> /etc/init.d/ido2db stop
```

Automatischer Start von IDOUtils

Fügen Sie IDOUtils zu der Liste der System-Services hinzu und sorgen Sie für einen automatischen Start, wenn das System hochfährt (stellen Sie sicher, dass Sie vorher das Init-Script installiert haben).

Fedora / openSuSE:

```
#> chkconfig --add ido2db
#> chkconfig ido2db on
```

Debian / Ubuntu:

```
#> update-rc.d ido2db defaults
```

Icinga starten:**Start von Icinga**

Überprüfen Sie die Icinga-Beispielkonfigurationsdateien.

```
#> /usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
```

Wenn es dabei keine Fehler gibt, starten Sie Icinga.

Fedora / openSuSE/Ubuntu:

```
#> service icinga start
```

Debian:

```
#> /etc/init.d/icinga start
```

Automatischer Start von Icinga

Fügen Sie Icinga zu der Liste der System-Services hinzu und sorgen Sie für einen automatischen Start, wenn das System hochfährt (stellen Sie sicher, dass Sie vorher das Init-Script installiert haben).

Fedora / openSuSE:

```
#> chkconfig --add icinga
#> chkconfig icinga on
```

Debian / Ubuntu:

```
#> update-rc.d icinga defaults
```

Anpassen der SELinux-Einstellungen

Fedora wird mit installiertem SELinux (Security Enhanced Linux) ausgeliefert und läuft im "Enforcing"-Modus. Dies kann zu "Internal Server Error"-Fehlern führen, wenn Sie versuchen, die Icinga-CGIs aufzurufen.

Schauen Sie, ob SELinux im Enforcing-Modus läuft.

```
#> getenforce
```

Setzen Sie SELinux in den "Permissive"-Modus.

```
#> setenforce 0
```

Damit diese Änderung dauerhaft wird, müssen Sie diese Einstellung in `/etc/selinux/config` anpassen und das System neustarten.

Statt SELinux zu deaktivieren oder es in den Permissive-Modus zu versetzen, können Sie den folgenden Befehl benutzen, um die CGIs im Enforcing/Targeted-Modus laufen zu lassen:

```
#> chcon -R -t httpd_sys_content_t /usr/local/icinga/sbin/
#> chcon -R -t httpd_sys_content_t /usr/local/icinga/share/
```

Besuchen Sie das NagiosCommunity.org-Wiki unter <http://www.nagioscommunity.org/wiki>, um Informationen darüber zu erhalten, wie die Icinga-CGIs im Enforcing-Modus mit einer Targeted-Richtlinie ausgeführt werden.

Anmelden am klassischen Web-Interface

Sie sollten nun auf das klassische Icinga-Web-Interface zugreifen können. Sie werden nach dem Benutzernamen (`nagiosadmin`) und Passwort gefragt, das Sie vorhin angegeben haben.

```
http://localhost/icinga/
```

oder

```
http://yourdomain.com/icinga/
```

Klicken Sie auf den "Service Detail"-Verweis in der Navigationsleiste, um Details darüber zu erhalten, was auf Ihrer lokalen Maschine überwacht wird. Es wird ein paar Minuten dauern, bis Icinga alle mit Ihrer Maschine verbundenen Services geprüft hat, weil die Prüfungen über eine gewisse Zeit verteilt werden.

Andere Anpassungen:

Stellen Sie sicher, dass die Firewall-Einstellungen Ihrer Maschine einen Zugriff auf das klassische Web-Interface ermöglichen, wenn Sie von anderen Rechnern darauf zugreifen wollen.

```
#> iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
```

Die Konfiguration von e-Mail-Benachrichtigungen ist nicht Gegenstand dieser Anleitung. Icinga ist konfiguriert, um e-Mail-Benachrichtigungen zu versenden, aber möglicherweise ist auf Ihrem System noch kein Mail-Programm installiert bzw. konfiguriert. Schauen Sie in Ihre Systemdokumentation, suchen Sie im Web oder gucken Sie im [NagiosCommunity.org-Wiki](http://www.nagioscommunity.org/wiki) nach genauen Anweisungen, wie Ihr System konfiguriert werden muss, damit es e-Mail-Mitteilungen an externe Adressen versendet. Mehr Informationen zu Benachrichtigungen finden Sie [hier](#)

Fertig

Glückwunsch! Sie haben erfolgreich Icinga installiert. Ihre Reise in die Überwachung hat gerade begonnen. Sie werden ohne Zweifel mehr als nur Ihre lokale Maschine überwachen wollen, so dass Sie u.a. das folgende [Kapitel](#) lesen sollten...

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Icinga-Schnellstart auf FreeBSD](#)

[Zum Anfang](#)

[Icinga-Schnellstart mit IDOUtils
auf FreeBSD](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Icinga-Schnellstart mit IDOUtils auf FreeBSD

[Zurück](#)

Kapitel 2. Los geht's

[Weiter](#)

Icinga-Schnellstart mit IDOUtils auf FreeBSD

Einführung



Anmerkung

Anstatt Icinga von Grund auf zu installieren möchten Sie vielleicht ein Paket benutzen, das es möglicherweise für Ihr Betriebssystem gibt. Bitte werfen Sie einen Blick auf die [Tabelle der Pakete](#).

Falls Sie aus den Sourcen installieren möchten, dann benutzen Sie bitte die offiziellen Release-Tarballs.



Wichtig

Bitte benutzen Sie keine GIT-Snapshots, solange Sie kein Problem haben, das in der aktuellen Entwicklerversion ggf. gelöst ist.

Diese Schnellstartanleitung ist dazu gedacht, Ihnen einfache Anweisungen zu liefern, wie Sie Icinga innerhalb von 30 Minuten aus dem Quellcode installieren und Ihren lokalen Rechner damit überwachen.

Hier werden keine fortgeschrittenen Installationsoptionen vorgestellt - lediglich die Grundlagen, die für 95% aller Benutzer funktionieren, die anfangen wollen.

Diese Anleitung enthält Anweisungen für [FreeBSD 8.1-RELEASE](#). Dank an "ScotchTape" für die angepassten Anweisungen.

Spätere Distributionen von FreeBSD werden wahrscheinlich auch mit diesen Anweisungen funktionieren.



Anmerkung

Inzwischen gibt es einen FreeBSD-Port von Icinga (`net-mgmt/icinga`), so dass Sie den vielleicht installieren möchten ;-)

Was dabei herauskommt

Wenn Sie diesen Anweisungen folgen, werden Sie am Ende folgendes haben:

- Icinga und die Plugins werden unterhalb von /usr/local/icinga installiert sein
- Icinga wird so konfiguriert sein, dass es einige Dinge auf Ihrem lokalen System überwacht (CPU-Auslastung, Plattenbelegung, usw.)
- das klassische Icinga-Web-Interface ist erreichbar unter `http://localhost/nagios/` oder `http://yourdomain.com/icinga/`
- eine Datenbank, die von Icinga mit Hilfe der IDOUtils gefüllt wird

Voraussetzungen

Während einiger Teile der Installation benötigen Sie **root**-Zugang zu Ihrer Maschine.

IDOUtils benutzt die [libdbi](#) und die libdbi-Treiber für verschiedene Datenbanken. Die Development-Libraries werden ebenfalls benötigt. Die folgenden Beispiele zeigen, wie die IDOUtils mit libdbi für MySQL oder PostgreSQL installiert werden.

Stellen Sie sicher, dass die folgenden Pakete installiert sind, bevor Sie fortfahren.

- [Apache](#)
- GCC-compiler
- [GD development libraries](#)
- libdbi-Treiber, eine Datenbank wie z.B. MySQL oder PostgreSQL



Anmerkung

PostgreSQL: Aufgrund von Änderungen an Insert-Statements funktioniert PostgreSQL 8.1 nicht länger (und ist auch bereits "End-of-life" seit November 2010), also benutzen Sie bitte 8.2 oder besser noch 8.4. String escaping mit PostgreSQL 9.x befindet sich noch in einem experimentellen Stadium (siehe auch [issue #1974](#)).

Optional

Zu irgendeiner Zeit möchten Sie wahrscheinlich SNMP-basierte Prüfungen verwenden, so dass es eine gute Idee ist, die benötigten Pakete gleich zu installieren. Andernfalls werden die Plugins nicht kompiliert und sind nicht verfügbar, wenn Sie diese brauchen.

Neue Features für die IDOUtils:

SSL-Verschlüsselung zwischen idomod und ido2db

Wenn Sie **SSL-Verschlüsselung** verwenden möchten: diese ist bereits installiert.



Anmerkung

SSL muss auch bei allen idomod-Clients aktiviert werden, sonst gehen Daten verloren!!!

Oracle-Datenbankunterstützung

Wenn Sie Oracle als RDBMS einsetzen möchten, dann ist das unter FreeBSD leider nicht so ohne weiteres möglich.

Zeitzonenunterstützung

Ab Icinga 1.6 werden alle Zeit-/Datumsinformationen als lokale Zeitstempel in der Datenbank gespeichert. Vorher gab es für jedes Datenbanksystem unterschiedliche Vorgehensweisen beim Speichern und Abrufen der Daten, was schwierig für Frontend-Applikationen war. Deshalb wurden die Datentypen geändert: für MySQL von DATETIME in TIMESTAMP und für Oracle von DATE in LOCAL TIMESTAMP (0). PostgreSQL benutzt bereits TIMESTAMP, es wurde aber in TIMESTAMP WITH TIME ZONE geändert. IDO2DB wird die Sitzungszeitzone auf UTC setzen und alle Unix-Timestamps (die per Definition UTC sind) als UTC-basierte Werte speichern. Bitte stellen Sie sicher, dass Ihr System Unix-Timestamps als UTC-basierte Werte ausgibt (wie "date -u '+%s'").



Anmerkung

Stellen Sie sicher, dass Ihre Datenbanksitzung in der gleichen Zeitzone läuft, in der die existierenden Zeit-/Datumsinformationen gespeichert wurden (überprüfen Sie Ihre lokale Zeitzone, z.B. Oracle: "select sessiontimezone from dual;"), wenn Sie das Upgrade-Script laufen lassen. Zusätzlich für Ihre Bequemlichkeit sollten Sie in Oracle das Format der Sitzungszeitstempel auf den Wert zu setzen, den Sie wünschen, z.B. "alter session set nls_timestamp_format='YYYY-MM-DD HH24:MI:SS'; oder etwas ähnliches. Nun können Sie Einträge von verschiedenen Icinga-Installationen in unterschiedlichen Zeitzonen ansehen, die in der gleichen Datenbank gespeichert wurden, und unabhängig davon, von woher Sie darauf zugreifen, werden sie in Ihrer lokalen Zeitzone angezeigt.

Installation der Pakete

Sie können diese Pakete aus den ports installieren oder Sie nehmen fertige und evtl. ältere packages (siehe FreeBSD-Schnellstart):

Installieren Sie die Ports

Sie können diese Ports mit den folgenden Befehlen installieren, es empfiehlt sich aber, portupgrade oder portmaster zu verwenden (als root):

Bitte aktualisieren Sie Ihre Ports bevor Sie beginnen.

```
#> cd /usr/ports/devel/libtool/ && make all install clean
#> cd /usr/ports/graphics/jpeg && make all install clean
#> cd /usr/ports/graphics/png && make all install clean
#> cd /usr/ports/graphics/gd && make all install clean
```



Anmerkung

Bitte stellen Sie sicher, dass Apache installiert ist - das Vorgehen wird hier nicht beschrieben, aber ein Hinweis ist

```
#> cd /usr/ports/www/apache22 && make clean && make
```

.

```
#> cd /usr/ports-devel/gmake && make all install clean
#> cd /usr/ports-devel/libltdl && make all install clean <-- wenn noch nicht installiert
```

MySQL

```
#> cd /usr/ports-databases/mysql51-server && make all install clean
#> cd /usr/ports-databases/libdbi-drivers && make all install clean
```

dort den richtigen Treiber für die DB auswählen

PostgreSQL

```
#> cd /usr/ports-databases/postgresql84-server && make all install clean
#> cd /usr/ports-databases/libdbi-drivers && make all install clean
```

dort den richtigen Treiber für die DB auswählen

Benutzerinformationen erstellen

Werden Sie zum root-Benutzer.

```
#> su -l
```

Erstellen Sie ein neues Benutzerkonto *icinga* ohne Passwort und ohne die Möglichkeit, sich anzumelden (setzen Sie kein Passwort, wenn Sie danach gefragt werden):

```
#> adduser -D -w no -s nologin
```

Damit Sie über das klassische Webinterface Befehle an Icinga senden können, legen Sie noch eine neue Gruppe *icinga-cmd* an und fügen Sie den Web-Server-Benutzer (*www*) und den Icinga-Benutzer dieser Gruppe hinzu:

```
#> pw groupadd -n icinga-cmd -M icinga,www
```

Icinga und die Plugins herunterladen

Wechseln Sie in Ihr lokales Source-Verzeichnis, z:b. `~/src`

```
#> mkdir ~/src
#> cd ~/src
```

Laden Sie die Sourcen von der [Icinga Website](#).

Vergessen Sie nicht die [Nagios-Plugins](#).

Icinga und IDOUtils kompilieren und installieren

Entpacken Sie das Icinga-Archiv (oder wechseln Sie in den GIT-Snapshot)

```
#> cd ~/src/
#> tar xvzf icinga-1.6.tar.gz
#> cd icinga-1.6
```



Anmerkung

Dieser absolute Pfad ist gemeint, wenn im Nachfolgenden von `'/path/to/icinga-src/'` die Rede ist.

Führen Sie das Icinga-configure-Script aus. Durch die Nutzung des --help-Flags erhalten Sie Hilfe zu den Optionen.

```
#> ./configure --with-command-group=icinga-cmd \
--enable-idoutils CPPFLAGS=-I/usr/local/include \
CFLAGS="-I/usr/local/include -L/usr/local/lib" \
--with-dbi-lib=/usr/local/lib --with-dbi-inc=/usr/local/include
```



Wichtig

Das angehängte `CPPFLAGS=-I/usr/local/include` ist wichtig für die IDOUtils bzw. das Broker-Modul.



Anmerkung

Sie sollten `CFLAGS=... " wie oben angegeben benutzen. Andernfalls finden Sie ggf. später folgende Zeilen in icinga.log:`

```
Error: Module '/usr/local/icinga/bin/idomod.o' is using an old or unspecified version of the event broker API. Module will
be unloaded.
Event broker module '/usr/local/icinga/bin/idomod.o' deinitialized successfully.
```

Mehr Informationen zu diesem Fehler finden Sie [hier](#).

Mit SSL-Verschlüsselung:

```
#> ./configure --with-command-group=icinga-cmd \
--enable-idoutils --enable-ssl CPPFLAGS=-I/usr/local/include \
--with-dbi-lib=/usr/local/lib --with-dbi-inc=/usr/local/include
```

Kompilieren Sie den Icinga-Source-Code. Es gibt auch eine extra Option für IDOUtils (*make idoutils*), wenn Sie nur dieses Modul erneut kompilieren möchten. Um mögliche Optionen zu sehen, rufen Sie lediglich "make" auf.

```
#> gmake all
```

Installieren Sie die Binaries, das Init-Script, Beispiel-Konfigurationsdateien und setzen Sie die Berechtigungen für das External-Command-Verzeichnis.

```
#> gmake install
#> gmake install-init
#> gmake install-config
#> gmake install-commandmode
#> gmake install-idoutils
```

oder kürzer

```
#> gmake fullinstall
#> gmake install-config
```



Anmerkung

Ab Icinga 1.5.0 ist `gmake install-config` NICHT mehr in `gmake fullinstall` enthalten, um ein versehentliches Überschreiben der Konfigurationsdateien zu verhindern.



Anmerkung

Die Icinga-API diente als eine in PHP geschriebene Datenbank-Abstraktionsschicht. Seit Icinga ist sie durch eine interne Datenbank-Abstraktionsschicht ersetzt worden, so dass Icinga-API nicht mehr installiert werden muss.

Starten Sie Icinga noch nicht - es gibt noch ein paar Dinge zu tun...

Anpassen der Konfiguration

Beispiel-Konfigurationsdateien werden durch

```
#> gmake install-config
```

in /usr/local/icinga/etc/ installiert.

Ändern Sie die */usr/local/icinga/etc/objects/contacts.cfg*-Konfigurationsdatei mit Ihrem bevorzugten Editor und passen die e-Mail-Adresse in der *icingaadmin*-Kontaktdefinition an, so dass sie die Adresse enthält, die im Falle von Alarmen benachrichtigt werden soll.

```
#> vi /usr/local/icinga/etc/objects/contacts.cfg
#> cd /usr/local/icinga/etc
#> mv idomod.cfg-sample idomod.cfg
#> mv ido2db.cfg-sample ido2db.cfg
```

Wenn Sie die IDOUtils mit SSL kompiliert haben, aktivieren Sie SSL in der idomod.cfg mit

```
use_ssl=1
output_type=tcpsocket
output=127.0.0.1
```

(passen Sie die IP-Adresse an, wenn sich Ihre Datenbank nicht auf localhost befindet!) und in der ido2db.cfg mit

```
use_ssl=1
socket_type=tcp
```



Anmerkung

Vergessen Sie nicht, alle anderen idomod-Clients auch neu zu kompilieren und auf ssl umzustellen, **anderenfalls werden Sie Daten verlieren!!!**

Aktivieren Sie das idomod-Eventbroker-Modul



Wichtig

Unter normalen Umständen gibt es im `modules`-Unterverzeichnis bereits die folgende module-Definition, so dass Sie keine Änderungen an der Hauptkonfigurationsdatei vornehmen müssen.

```
define module{
    module_name      ido_mod
    path             /usr/local/icinga/bin/idomod.o
    module_type      neb
    args            config_file=/usr/local/icinga/etc/idomod.cfg
}
```

Das bedeutet auch, dass ein entsprechender broker_module-Eintrag nicht aktiviert werden darf!

Konfigurieren von Datenbank und IDOUtils

MySQL:

Anlegen von Datenbank, Benutzer und Berechtigungen



Anmerkung

Falls Sie gerade ein neues Datenbanksystem installiert haben, dann müssen Sie den Datenbank-Server-Prozess starten, bevor Sie eine Datenbank anlegen können. Im Falle von MySQL benutzen Sie `/usr/local/etc/rc.d/mysql-server start`.

```
#> mysql -u root -p

mysql> CREATE DATABASE icinga;

GRANT USAGE ON *.* TO 'icinga'@'localhost'
    IDENTIFIED BY 'icinga'
    WITH MAX_QUERIES_PER_HOUR 0
    MAX_CONNECTIONS_PER_HOUR 0
    MAX_UPDATES_PER_HOUR 0;

GRANT SELECT , INSERT , UPDATE , DELETE
    ON icinga.* TO 'icinga'@'localhost';

FLUSH PRIVILEGES ;

quit

#> cd /path/to/icinga-src/module/idoutils/db/mysql
#> mysql -u root -p icinga < mysql.sql

#> vi /usr/local/icinga/etc/ido2db.cfg

db_servertype=mysql
db_port=3306
db_user=icinga
db_pass=icinga
```

PostgreSQL:

To Do

Installieren und konfigurieren des klassischen Web-Interface

Icinga stellt das klassische Web-Interface zur Verfügung ("Classic Web", "die CGIs"). Sie können dieses wie folgt installieren:

```
#> cd /path/to/icinga-src
#> gmake cgis
#> gmake install-cgis
#> gmake install-html
```

Wenn Sie (zusätzlich) das neue Icinga-Web installieren wollen, lesen Sie bitte [Installation des Web-Interface](#).

Installieren Sie die Icinga-Web-Konfigurationsdatei im Apache-Includes-Verzeichnis.

```
#> cd /path/to/icinga-src
#> gmake install-webconf
```

Legen Sie ein `icingaadmin`-Konto an, um sich am klassischen Web-Interface anmelden zu können. Merken Sie sich das Passwort, das Sie diesem Konto geben - Sie brauchen es später.

```
#> htpasswd -c /usr/local/icinga/etc/htpasswd.users icingadmin
```

Wenn Sie das Passwort später ändern oder einen weiteren Benutzer hinzufügen möchten, verwenden Sie den folgenden Befehl:

```
#> htpasswd /usr/local/icinga/etc/htpasswd.users <USERNAME>
```

Starten Sie Apache neu, damit die Änderungen wirksam werden.

```
#> service apache22 reload
```



Anmerkung

Prüfen Sie die Implementierung der verbesserten CGI-Sicherheitsmaßnahmen wie [hier](#) beschrieben, um sicherzustellen, dass Ihre Web-Authentifizierungsinformationen nicht kompromittiert werden.

Kompilieren und installieren der Nagios-/Perl-Plugins

Entpacken Sie die Nagios-Plugins-Quellcode-Archivdatei.

```
#> cd ~/src
#> tar xvzf nagios-plugins-1.4.15.tar.gz
#> cd nagios-plugins-1.4.15
```

Kompilieren und installieren Sie die Plugins

```
#> ./configure --prefix=/usr/local/icinga --with-cgiurl=/icinga/cgi-bin \
--with-htmurl=/icinga --with-nagios-user=icinga --with-nagios-group=icinga
#> make
#> make install
```



Anmerkung

Es gibt auch einen port für die Plugins, allerdings installiert dieser die Plugins in ein anderes Verzeichnis. Man kann dessen make zwar mit Variablen bestücken, muss aber trotzdem später manuell umkopieren.

Kompilieren und installieren Sie das Perl-Plugin:

```
#> cd /usr/ports/net-mgmt/p5-Nagios-Plugin
#> make all install clean
```

IDOUtils und Icinga starten

IDOUtils muss vor Icinga gestartet werden

IDOUtils starten

```
#> /usr/local/etc/rc.d/ido2db start
```

IDOUtils beenden

```
#> /usr/local/etc/rc.d/ido2db stop
```

Icinga starten

Fügen Sie Icinga zur Liste der System-Services hinzu, damit es automatisch beim Start des Systems gestartet wird (stellen Sie sicher, dass Sie das Init-Script vorher installiert haben).

```
#> echo icinga_enable=\"YES\" >> /etc/rc.conf
```

Überprüfen Sie die Icinga-Konfigurationsdateien.

```
#> /usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
```

Wenn es dabei keine Fehler gibt, starten Sie Icinga.

```
#> /usr/local/etc/rc.d/icinga start
```

Anmelden am klassischen Web-Interface

Sie sollten nun auf das klassische Icinga-Web-Interface zugreifen können. Sie werden nach dem Benutzernamen (*icingaadmin*) und Passwort gefragt, das Sie vorhin angegeben haben.

<http://localhost/icinga/>

oder

<http://yourdomain.com/icinga/>

Klicken Sie auf den "Service Detail"-Verweis in der Navigationsleiste, um Details darüber zu erhalten, was auf Ihrer lokalen Maschine überwacht wird. Es wird ein paar Minuten dauern, bis Icinga alle mit Ihrer Maschine verbundenen Services geprüft hat, weil die Prüfungen über eine gewisse Zeit verteilt werden.

Fertig

Glückwunsch! Sie haben erfolgreich Icinga installiert. Ihre Reise in die Überwachung hat gerade begonnen.

Sie werden ohne Zweifel mehr als nur Ihre lokale Maschine überwachen wollen, so dass Sie u.a. das folgende [Kapitel](#) lesen sollten...

Pakete für Icinga

Compiler-Optionen für Icinga mit IDOUtils

```
./configure --with-httdp-conf=/usr/local/etc/apache22/Includes/ \
--with-gd-lib=/usr/local/lib/ --with-gd-inc=/usr/local/include/ \
--with-command-group=icinga-cmd --enable-idoutils \
--with-dbi-inc=/usr/local/include --with-dbu-lib=/usr/local/lib \
CPPFLAGS=-I/usr/local/include CFLAGS=-fPIC
```

Compiler-Optionen für Nagios-Plugins (ports)

```
make install NAGIOSUSER=icinga NAGIOSGROUP=icinga \
PREFIX=/usr/local/icinga
```

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Icinga-Schnellstart mit IDOUtils

[Zum Anfang](#)

[Links zu weiteren Howtos](#)

**Links zu weiteren Howtos**[Zurück](#)[Kapitel 2. Los geht's](#)[Weiter](#)

Links zu weiteren Howtos

Hier findet ihr eine Liste von veröffentlichten Howtos



Anmerkung

Einige dieser Howtos beschreiben veraltete Versionen von Icinga, Add-ons und/oder Plugins, so dass verschiedene Anweisungen überholt sein können. Im Zweifelsfall ist es ratsam, einen Blick in die aktuelle Dokumentation zu werfen.

[Icinga-Reporting-Integration \(mit JasperServer\) \[en\]](#) (Icinga Development Team)

[Setting up DNX with Icinga \[en\]](#) (Icinga Development Team)

[Setting up mod gearman with Icinga \[en\]](#) (Icinga Development Team)

[Developing modules for Icinga-Web \[en\]](#) (Icinga Development Team)

[Developing cronks for Icinga-Web \[en\]](#) (Icinga Development Team)

[Setting up AD authentication for Icinga-Web \[en\]](#) (Sebastian Waitz)

[Icinga and Oracle](#) (Icinga Development Team)

[Step By Step install using RPMS for fc13](#) (Icinga Development Team)

[Installing Icinga with IDOUtils, MySQL and NConf \[en\]](#) (Andreas Lehr)

[Icinga Installation, quick and dirty, without database \[en\]](#) (Moritz Tanzer)

[Icinga, IDOUtils und MySQL-Installations-HowTo \[de\]](#) (Moritz Tanzer)

[Icinga Installation unter Debian 5 \(lenny\) mit IDOUtils und MySQL DB](#) (Patrick Schoyswohl)

[Icinga- und Icinga-Web-Installation unter Ubuntu](#) (Patrick Gotthard)

[Icinga in einer Microsoft Hyper-V Maschine installieren](#) (Helmut Thurnhofer)

[Diverse Icinga Monitoring PDFs](#) (Helmut Thurnhofer)

Dort findet ihr:

- "ICINGA in einer virtuellen Umgebung mit Ubuntu 9.10 Desktop installieren"
- "ICINGA 1.0.1 in einer virtuellen Umgebung mit Ubuntu 10.04 Server installieren"
- "ICINGA 1.0.2 in einer Microsoft Hyper-V Maschine mit Ubuntu 10.04 Desktop installieren"
- "Zusatzkomponenten/Plug-Ins und Perl Skripte für ICINGA 1.0.2 installieren"

als PDF's zum downloaden.

Herzlichen Dank an ALLE, die ihre Dokumentationen anderen zur Verfügung stellen!



Anmerkung

Im [Icinga-Wiki](#) gibt es einen eigenen [HowTo-Bereich](#), der verschiedene Aspekte behandelt, u.a.:

- Addons
- Backup
- Classic UI
- Collections
- Installation and Configuration Guides
- Integration
- Notifications
- Performance Tuning
- Plugins

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Icinga-Schnellstart mit IDOUtils
auf FreeBSD

[Zum Anfang](#)

[Icinga aktualisieren](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Icinga aktualisieren

[Zurück](#)[Kapitel 2. Los geht's](#)[Weiter](#)

Icinga aktualisieren

Inhalt

[Aktualisierung von Icinga](#)[Upgrade von Nagios 3.x](#)[Upgrade von Nagios 2.x](#)[Aktualisierung einer RPM-Installation](#)

Anmerkung

Wenn Sie die IDOUtils benutzen, dann müssen Sie auch diese aktualisieren. Weitere Informationen finden Sie im Abschnitt [IDOUtils-Datenbank aktualisieren](#).



Anmerkung

Wenn Sie Icinga-Web benutzen, dann müssen Sie auch diese aktualisieren. Weitere Informationen finden Sie im Abschnitt [Aktualisierung von Icinga-Web und Icinga-Web-Datenbank](#).

Aktualisierung von Icinga

Sobald neuere Versionen von Icinga herauskommen, sollten Sie dringend über eine Aktualisierung nachdenken. Neuere Ausgaben enthalten Behebungen kritischer Fehler, so dass es wichtig ist, aktuell zu sein. Wenn Sie bereits Icinga, wie in den Schnellstartanleitungen beschrieben, aus dem Quellcode installiert haben, dann können Sie einfach neuere Versionen installieren. Sie müssen dazu noch nicht einmal root-Berechtigungen haben, weil bereits alles passiert ist, was als root-Benutzer getan werden muss. Das ist allerdings abhängig davon, welche Präferenzen Sie diesbezüglich haben.

Stellen Sie sicher, dass Sie eine gute Datensicherung Ihrer bestehenden Icinga-Installation und der Konfigurationsdateien haben. Wenn irgendetwas schief geht oder nicht funktioniert, dann können Sie auf diese Weise schnell Ihre alte Icinga-Version wiederherstellen.

Werden Sie der icinga-Benutzer. Debian/Ubuntu-Benutzer sollten sudo -s icinga benutzen.

```
$> su -l icinga
```

Holen Sie sich das Quellcode-Archiv der letzten Icinga-Version (besuchen Sie <http://www.icinga.org/> für den Verweis auf die letzte Version) und entpacken Sie das Quellcode-Archiv.

Starten Sie das Icinga-configure-Script mit den gleichen Optionen wie bei der letzten Installation, z.B. so:

```
#> ./configure --with-command-group=icinga-cmd --enable-idoutils
```

Kompilieren Sie den Icinga-Quellcode.

```
#> make all
```

Installieren Sie aktualisierte Programme, Dokumentation und Web-Interface. Ihre vorhandenen Konfigurationsdaten werden in diesem Schritt nicht überschrieben.

```
#> make install install-base install-cgis install-html install-init install-commandmode install-idoutils
```

Überprüfen Sie Ihre Konfigurationsdateien und starten Sie Icinga erneut.

```
#> /etc/init.d/icinga checkconfig
#> /etc/init.d/icinga restart
```

Das war's - Sie sind fertig!

Upgrade von Nagios 3.x

Icinga ist aus Nagios 3.x hervorgegangen, so dass die Aktualisierung problemlos sein sollte.

Falls Sie einen Upgrade von Nagios-Version 3.0.x durchführen, dann fehlt Ihnen ggf. PHP.

Debian / Ubuntu

```
#> apt-get install php5 libapache2-mod-php5
```

Fedora / RedHat

```
#> yum install php mod_php
```

openSuSE / SLES: Nutzen Sie yast zur Installation der Pakete *php5* und *apache2-mod_php5* oder benutzen Sie zypper

```
#> zypper install php5 apache2-mod_php5
```

Stellen Sie sicher, dass Sie eine gute Datensicherung Ihrer bestehenden Nagios-Installation und der Konfigurationsdateien haben. Wenn irgendetwas schief geht oder nicht funktioniert, dann können Sie auf diese Weise schnell Ihre alte Nagios-Version wiederherstellen.

Bitte installieren Sie Icinga anhand der [Schnellstart-Anleitung](#). Bitte beachten Sie, dass

- der Default-Präfix nun "/usr/local/icinga" heißt
- die Umgebungs-Makros nun mit ICINGA_ beginnen

PNP4Nagios berücksichtigt das seit 0.6rc1 (2009.09.20), aber Sie benötigen die Makros lediglich im "sync"-Modus.

check_multi bietet Unterstützung seit 0.21 (2010.06.03), aber Sie müssen dazu die Installation mit beginnend mit dem Schritt 'configure --with-nagios_name=icinga' erneut durchführen, damit die check_multi-Prozedur mit geänderten Werten erstellt wird. Stattdessen können Sie auch die Option "-s" zur Übergabe von Werten benutzen.

- die Konfigurationsdateien der (verbesserten) IDOUtils heißen nun idomod.cfg/ido2db.cfg anstatt ndomod.cfg/ndo2db.cfg

Werden Sie der nagios-Benutzer. Debian/Ubuntu-Benutzer sollten *sudo -s nagios* benutzen.

```
$ su -l nagios
```

Holen Sie sich das Quellcode-Archiv der letzten Icinga-Version (besuchen Sie <http://www.icinga.org/> für den Verweis auf die letzte Version).

```
#> wget http://osdn.dl.sourceforge.net/sourceforge/icinga/icinga-1.6.tar.gz
```

Entpacken Sie das Quellcode-Archiv.

```
#> tar xzf icinga-1.6.tar.gz
#> cd icinga-1.6
```

Starten Sie das Icinga-configure-Script mit den Optionen, die Sie beim ./configure von Nagios benutzt haben. Den Aufruf finden Sie in der Datei config.log. Beispiel:

```
#> ./configure --with-command-group=nagcmd
```

Kompilieren Sie den Icinga-Quellcode.

```
#> make all
```

Installieren Sie aktualisierte Programme, Dokumentation, Web-Interface und das Init-Script. Ihre vorhandenen Konfigurationsdaten werden in diesem Schritt nicht überschrieben.

```
#> make cgis
#> make install
#> make install-cgis
#> make install-init
```

oder einfach

```
#> make fullinstall
```



Wichtig

Erst ab Icinga 1.5.0 ist make install-config NICHT mehr in make fullinstall enthalten, um ein versehentliches Überschreiben der Konfigurationsdateien zu verhindern.

Kopieren Sie Ihre Konfigurationsdateien nach /usr/local/icinga/etc bzw. /usr/local/icinga/etc/object. Vor dem Start von Icinga müssen Sie noch ein paar Dinge anpassen:

- Benennen Sie die Hauptkonfigurationsdatei nagios.cfg in icinga.cfg um und ändern Sie in /usr/local/icinga/etc/icinga.cfg die Namen der Direktiven "nagios_user" in "icinga_user" und "nagios_group" in "icinga_group". Das betrifft ggf. auch die Pfade in der Datei.

```
#> sed -i 's/nagios/icinga/g' ./icinga.cfg/
```

- Ändern Sie in der CGI-Konfigurationsdatei cgi.cfg die Pfad-Angaben.

```
#> sed -i 's/nagios/icinga/g' ./cgi.cfg/
```

Kopieren Sie andere relevante Dateien von Ihrer Nagios-Installation zum neuen Standort. Falls Sie unsicher bezüglich der Pfade sind, dann werfen Sie einen Blick in die Konfigurationsdateien nagios.cfg und/oder icinga.cfg.

- retention.dat (sie enthält bestimmte Statusinformationen, Kommentare und andere "bleibende" Dinge)
- nagios.log (bitte umbenennen in icinga.log)
- archives/nagios-<date>.log-Dateien (Icinga ist in der Lage, sowohl nagios-<date>.log als auch icinga-<date>.log-Dateien zu verarbeiten, so dass Sie die Dateien nicht umbenennen müssen)
- Sie müssen die Dateien status.dat und/oder objects.cache nicht kopieren, weil sie jeweils beim Neustart erstellt werden. Bitte erstellen Sie objects.precache vor dem Neustart ([falls nötig](#)) anstatt die Datei zu kopieren

Überprüfen Sie Ihre Konfigurationsdateien und starten Sie Icinga.

```
#> /usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
#> /etc/init.d/icinga start
```

Das war's - Sie sind fertig!

Bitte beachten Sie, dass

- der Aufruf im Browser nun <http://localhost/icinga/> lautet (für das klassische UI)
- der Name des Admin-Benutzers nun icingaadmin lautet

Upgrade von Nagios 2.x

Es sollte nicht allzu schwierig sein, von Nagios 2.x auf Icinga 1.6 zu aktualisieren. Die Aktualisierung ist im Wesentlichen die gleiche wie die von bestehenden Nagios 3.x-Versionen. Allerdings müssen Sie Ihre Konfigurationsdateien ein wenig ändern, damit sie mit Icinga 1.6 funktionieren:

- Die alte *service_reaper_frequency*-Variable in der Hauptkonfigurationsdatei wurde umbenannt in [check_result_reaper_frequency](#).
- Das alte \$NOTIFICATIONNUMBER\$-Makro entfällt zugunsten der \$HOSTNOTIFICATIONNUMBER\$- und \$SERVICENOTIFICATIONNUMBER\$-Makros.
- Die alte *parallelize*-Direktive in Service-Definitionen ist veraltet und wird nicht länger benutzt, weil alle Service-Prüfungen parallel ablaufen.
- Die alte *aggregate_status_updates*-Option wurde entfernt. Alle Statusdatei-Aktualisierungen werden nun mit einem minimalen Intervall von einer Sekunde zusammengefasst.
- Erweiterte Host- und erweiterte Service-Definitionen sind veraltet. Sie werden noch von Icinga gelesen und verarbeitet, aber es wird empfohlen, dass Sie diese Direktiven in die entsprechenden Host- und Service-Definitionen verschieben.

- Die alte *downtime_file*-Dateivariable in der Hauptkonfigurationsdatei wird nicht länger unterstützt, weil Einträge von geplanten Ausfallzeiten (downtimes) nun in der **Aufbewahrungsdatei** (retention file) gespeichert werden. Um bestehende Einträge zu erhalten, stoppen Sie Nagios 2.x und hängen Sie den Inhalt Ihrer alten Downtime-Datei an das "retention file".
- Die alte *comment_file*-Dateivariable in der Hauptkonfigurationsdatei wird nicht länger unterstützt, weil Kommentare nun in der **Aufbewahrungsdatei** (retention file) gespeichert werden. Um bestehende Einträge zu erhalten, stoppen Sie Nagios 2.x und hängen Sie den Inhalt Ihrer alten Kommentar-Datei an die "Aufbewahrungsdatei" (retention file).
- Die Hauptkonfigurationsdatei heißt nun icinga.cfg. Innerhalb der Datei ist "nagios_user" gegen "icinga_user" und nagios_group" gegen "icinga_group" auszutauschen.

Stellen Sie außerdem sicher, dass Sie den "[Was gibt's Neues](#)"-Abschnitt in der Dokumentation lesen. Er beschreibt all die Änderungen am Icinga-Code.

Aktualisierung einer RPM-Installation

Wenn Sie momentan eine RPM- oder Debian/Ubuntu-APT-paketbasierte Nagios-Installation haben und nun den Übergang zu einer Installation aus dem offiziellen Quellcode machen wollen, dann sind hier die grundlegenden Schritte:

1. Sichern Sie Ihre existierende Nagios-Installation
2. Konfigurationsdateien
 - - Hauptkonfigurationsdatei (normalerweise nagios.cfg)
 - Ressource-Konfigurationsdatei (normalerweise resource.cfg)
 - CGI-Konfigurationsdatei (normalerweise cgi.cfg)
 - all Ihre Objektdefinitionsdateien
 - Aufbewahrungsdatei (normalerweise retention.dat)
 - die aktuelle Nagios-Protokolldatei (normalerweise nagios.log)
 - archivierte Nagios-Protokolldateien
3. Deinstallieren Sie die originalen RPM- oder APT-Pakete
4. Installieren Sie Icinga aus dem Quellcode, indem Sie der [Schnellstartanleitung](#) folgen
5. Sichern Sie Ihre Original-Nagios-Konfigurationsdateien, Aufbewahrungs- und Protokolldateien wieder zurück
6. Benennen Sie die Hauptkonfigurationsdatei nagios.conf in icinga.conf um und ändern Sie in /usr/local/icinga/etc/icinga.cfg die Namen der Direktiven "nagios_user" in "icinga_user" und "nagios_group" in "icinga_group".
7. [Überprüfen](#) Sie Ihre Konfiguration und [starten](#) Sie Icinga

Beachten Sie, dass verschiedene RPM- oder APT-Pakete Nagios auf verschiedene Weisen oder an verschiedenen Orten installieren. Stellen Sie sicher, dass Sie all Ihre kritischen Nagios-Dateien gesichert haben, bevor Sie das Original-RPM- oder APT-Paket entfernen, so dass Sie darauf zurückgreifen können, wenn Sie auf Probleme stoßen.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Links zu weiteren Howtos](#)

[Zum Anfang](#)

[IDOUtils-Datenbank aktualisieren](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



IDOUtils-Datenbank aktualisieren

[Zurück](#)

Kapitel 2. Los geht's

[Weiter](#)

IDOUtils-Datenbank aktualisieren

Es mag einen Bug im Datenbankschema geben, der behoben wurde. Wenn Sie eine ältere IDOUtils-Version aktualisieren, dann müssen Sie außerdem diese Anpassungen manuell ausführen. Wenn Sie rpm/deb-Pakete benutzen, lesen Sie bitte die Hinweise und/oder fragen Sie den Maintainer, ob er diese Anpassungen in der Installationsroutine hinzugefügt hat.



Anmerkung

Abhängig von den Änderungen und der Größe Ihrer Datenbank kann es eine Weile dauern, die Anpassungen durchzuführen. Bitte haben Sie ein wenig Geduld und brechen Sie das Script nicht ab, weil sonst ggf. Ihre Daten unbrauchbar sind.

Die Update-Dateien finden Sie zusammen mit den Datenbank-Installationsdateien in
/path/to/icinga-src/module/idoutils/db/DeineDB/

Die Syntax ist wie folgt

```
<rdbm>-upgrade-<version>.sql
```

wobei <rdbm> mysql, pgsql oder oracle sein kann und <version> zeigt auf die Version, auf die Sie aktualisieren wollen.



Anmerkung

Wenn Sie eine ältere Version aktualisieren wollen und zwischen dieser und der aktuellen noch andere Versionen liegen, dann sollten Sie beachten, dass Sie auch die dazwischen liegenden Updates inkrementell installieren müssen!

Sie haben z.B. 1.0RC1 installiert und möchten auf 1.0.1 aktualisieren - Sie müssen dann zuerst auf 1.0 Stable updaten und dann die Aktualisierung auf 1.0.1 durchführen.

1. Sichern Sie Ihre aktuelle Datenbank vor der Aktualisierung!
2. Prüfen Sie die laufende IDOUtils-Version und die Zielversion. Prüfen Sie, ob zwischen diesen beiden Versionen noch andere Versionen liegen und aktualisieren Sie ggf. schrittweise.
3. Führen Sie die Aktualisierung(en) mit einem Benutzer durch, der über die notwendigen Berechtigungen verfügt. Sie können das upgradedb-Script verwenden, aber das wird nicht empfohlen (betrifft nur MySQL).

MySQL:

```
$ mysql -u root -p <dbname> < /path/to/icinga-src/module/idoutils/db/mysql/mysql-upgrade-<version>.sql
```

PostgreSQL:

```
# su - postgres
$ psql -U icinga -d icinga < /path/to/icinga-src/module/idoutils/db/pgsql/pgsql-upgrade-<version>.sql
```

Oracle:

```
# su - oracle
$ sqlplus dbuser/dbpass
SQL> @oracle-upgrade-<version>.sql
```

Aktualisierung der IDOUtils auf 1.5**Oracle****Anmerkung**

Zum Einsatz von IDOUtils 1.5.0 mit Oracle benötigen Sie OCILIB 3.9.2 - installieren Sie nicht 3.9.0 oder 3.9.1, denn diese Version sind fehlerbehaftet.

Passen Sie `oracle-upgrade-1.5.0.sql` an und definieren Sie Ihre Tablespaces ähnlich den folgenden Zeilen. Sie können auch Ihre bestehenden Tablespace-Namen für alle Definitionen benutzen.

```
/* set real TBS names on which you have quota, no checks are implemented!*/
define DATATBS='icinga';
define LOBTBS='icinga';
define IXTBTS='icinga';
```

**Anmerkung**

Detailliertere Informationen zu IDOUtils mit Oracle finden Sie in unserer Wiki-HowTo-Serie zu [Oracle](#).

Aktualisierung der IDOUtils auf 1.4**Oracle**

- Die minimal erforderliche Version ist Oracle 10g R2. Ältere Versionen können ggf. ebenfalls funktionieren, werden aber nicht unterstützt.
- Die optionale Trennung von Daten-, Index- und LOB-Tablespaces wird eingeführt. Passen Sie `oracle-upgrade-1.4.0.sql` an und definieren Sie Ihre Tablespaces. Sie können auch Ihre bestehenden Tablespace-Namen für alle Definitionen benutzen.

Aktionen:

- entfernen der Beschränkungen von Zahlenwerten
- entfernen der meisten bestehenden NOT NULL-Constraints
- benennen von Constraints

- anlegen von Index und LOBs in neuen Tablespaces
- Sequenzen auf NOCACHE setzen
- Oracle-Funktion anpassen, um NO_DATA-Exceptions zu erzeugen



Achtung

Aktualisierung der IDOUtils auf Oracle 1.4 erfordert ein bisschen "Magie". Stellen Sie sicher, dass

1. Sie den kompletten upgrade-Ordner kopieren
2. Sie das Script `oracle-upgrade-1.4.0.sql` anpassen und die Werte für DATA, LOB und IXT setzen
3. Sie anschließend das Upgrade-Script starten

Aktualisieren der IDOUtils auf 1.3

Mit IDOUtils 1.3 wird die Verwendung der Tabelle dbversion im IDOUtils Schema wieder eingeführt. Das Aktualisierungsskript stellt sicher, dass die Tabelle dbversion die aktuelle Version enthält. Ido2db vergleicht die Programversion mit der Datenbankversion und gibt einen Fehler im Syslog aus, wenn die Versionen voneinander abweichen.

Verwenden Sie das Aktualisierungsskript für die IDOUtils 1.3 unter `module/idouils/db/<rdbms>/<rdbm>-upgrade-1.3.sql` gegen Ihr aktuelles Datenbankschema, bitte beachten Sie die inkrementelle Vorgehensweise wie oben beschrieben.

Nach der Aktualisierung sollten Sie die Datenbank-Version prüfen.

```
SQL> SELECT * FROM icinga_dbversion
```



Anmerkung

Der Oracle-Tabellename lautet "dbversion" (anstatt "icinga_dbversion").

Aktualisieren der IDOUtils auf 1.0.3

Es gab ein paar kleinere Änderungen:

- `display_name` wurde geändert zu `varchar(255)` für mysql/oracle
- Update des pgsql schema, ersetzen von `varchar(n)` durch `text`
- Geänderter Wert für Konfigdateivariablen bis 1024 Länge für MySQL/Oracle.

Bitte verwenden Sie das Update-Skript für ihre Datenbank wie oben beschrieben.

Aktualisieren der IDOUtils auf 1.0.2

Es gab einen signifikanten Bug in den IDOUtils, der erst in Icinga 1.0.2 bereinigt werden konnte.

Bei jedem Core Restart wurde die gesamte Menge an Objekten in der Objekttabelle erneut hinzugefügt, anstelle die alten weiterhin zu verwenden und wie Relationen auf den neuesten Stand zu bringen.

Beispielsweise bei 4000 Objekten (Hosts, Services, Contacts, etc) hat ein zweimaliger Core Restartet $4000+4000+4000 = 12000$ Objekte bedeutet.

In Bezug auf die Konfiguration und die Statusdaten ist dies nicht relevant, da deren Relationen zur Objekttabelle bei jedem Neustart bereinigt werden.

Historische Daten allerdings behalten diese unterschiedliche Beziehung zur Objekttabelle bei - vor und nach dem Restart sind die Relationen unterschiedlich.

Diese Dateninkonsistenz ist natürlich nicht wünschenswert und es wurde dementsprechend versucht, eine einfache Lösungsmöglichkeit zu finden.

Neben den normale SQL Scripts für 1.0.2 (z.B. mysql-upgrade-1.0.2.sql) stehen erweiterte SQL Scripts zur Verfügung.

Das Script arbeitet jeweils auf einer historischen Tabelle und holt sich mit Hilfe einer gestaffelten Query die notwendigen Daten aus den beiden Tabellen - historisch 1..1 Objekte. Desweiteren werden kaputte Eintraege zur Zeit des Restarts bereinigt.

Bitte verwenden Sie diese Scripts wie Sie möchten - wahlweise direkt ausgeführt oder Schritt für Schritt, wie der Ablauf innerhalb des Scripts ist. Beachten Sie allerdings bitte, dass diese Scripts ohne Garantie auf ihr eigenes Risiko verwendet werden können.

Falls Sie lediglich Livedaten verwenden, ist unter Umständen eine Neuinstallation des Datenbankschemas die einfachere Option.

* <rdbms>-upgrade-1.0.2-fix-object-relations.sql

Das "normale" Upgrade Script ist in 1.0.2 nur für MySQL verfügbar. Es wurden binäre Casts entfernt, da case sensitiv Vergleichen auch mit einer Anpassung der Collation erreicht werden kann und so massive Performanceeinbrüche verhindert werden können.

* mysql-upgrade-1.0.2.sql

Aktualisieren der IDOUtils auf 1.0.1

Bitte vergewissern Sie sich, dass Sie bereits auf Icinga IDOUtils 1.0 aktualisiert haben, bevor Sie diesen Abschnitt weierlesen! Es gab einige (große) Veränderungen für alle unterstützten RDBMS, deshalb lesen Sie diesen Abschnitt bitte sehr sorgfältig. Alle Datenbank- Skripte sind nun in entsprechenden Unterzeichnissen zu finden. Für alle RDBMS wurden mehr Indizes gesetzt, außerdem wurde die Größe der command_line Spalte in mehreren Tabellen, die 255 Zeichen überschritten, angepasst.

RDBMS spezifische Änderungen und HowTo's:

MySQL:

Änderung der Datenbank- Engine von MYISAM zu InnoDB. Der Grund ist die Umgehung von Zeilen- Sperren/Transaktionen/Rollbacks im Gegensatz zu einer kleinen Geschwindigkeitseinbuße während der Inserts.

Das Upgrade-Skript führt eine ALTER TABLE- Anweisung aus. Falls Ihnen diese Idee nicht gefällt, können Sie auch folgendes tun:

- Dump erstellen der existierenden Datenbank:

```
# mysqldump -u root -p icinga > icinga.sql
```

- Ändern Sie alle Einträge von "MYISAM" zu "InnoDB"
- Import des angepassten Datensatzes in eine neue Datenbank (wenn Sie die alte Datenbank verwenden möchten, löschen Sie als erstes den originalen Datensatz und rekreieren Sie die Datenbank).

PostgreSQL:

Der Tabelle systemcommands fehlte die Spalte der Namens Ausgabe. Diese wird während des Upgrades hinzugefügt.

Oracle:

Um die Performance in mehreren Bereichen zu verbessern, muß der Wert für open_cursors höher gesetzt werden (Standardwert ist 50). Das Aktualisierungsskript enthält zwei neue, in DML geschriebene, Prozeduren für die DELETE- Anweisungen.

Darüber hinaus gab es umfangreiche Änderungen bezüglich der Autoincrement- Sequenz und der AFTER INSERT- Trigger (Emulation des MySQL Autoincrement auf Primärschlüssel). Die alte Routine wurde komplett verworfen, d.h. alle Trigger und die Autoincrement- Sequenz werden während des Updates entfernt. Als Ersatz werden für jede Tabelle neue Sequenzen hinzugefügt und in den IDOUtils für Oracle verwendet.

Bei bestehenden Datensätzen wird dies beim Importieren zu Problemen führen. Die Sequenzen starten mit dem Wert 1 wo hingegen der primäre Key (id) einen Maximalwert gesetzt hat. Aus diesem Grund wird eine Basisfunktion bereitgestellt, die das folgende tut: Diese extrahiert den maximalen Wert der id plus eins aus der angegebenen Tabelle und verändert den jeweiligen Sequence Start auf diesen berechneten Wert.

Bitte verwenden Sie diese Prozedur so, wie Sie es benötigen - auf alle Tabellen und Sequenzen oder auf separierte Teile. Die Prozedur ist auskommentiert, und wird ohne Garantie auf Datenkonsistenz zur Verfügung gestellt. Ziehen Sie Ihren DBA zu Rate, wenn Sie bestehende Datensätze importieren wollen.

Aktualisieren der IDOUtils auf 1.0

Es gab einen Unique-Key-Fehler durch den Fork, der bei einigen Tabellen zu doppelten und nutzlosen Zeilen führt. Dies betrifft die folgenden Tabellen:

- timedevents, timedeventqueue
- servicechecks
- systemcommands

Wenn Sie sich z.B. Definition der Tabelle servicechecks ansehen:

```
mysql> show create table icinga_servicechecks;
```

sollten Sie etwa folgendes sehen

```
PRIMARY KEY ('servicecheck_id'),
KEY 'instance_id' ('instance_id'),
KEY 'service_object_id' ('service_object_id'),
KEY 'start_time' ('start_time')
```

Um die o.g. Definition zu etwas wie diesem

```
PRIMARY KEY ('servicecheck_id'),
UNIQUE KEY 'instance_id' ('instance_id','service_object_id','start_time','start_time_usec')
```

zu ändern, befolgen Sie bitte den folgenden Ablauf!

Wenn Sie von IDOUtils 1.0RC aktualisieren, dann benutzen Sie bitte module/idoutils/db/mysql/mysql-upgrade-1.0.sql - wenn Sie von einer älteren Version aktualisieren, dann führen Sie vorher die notwendigen Schritte durch, um auf 1.0RC zu aktualisieren!

Bitte sichern Sie Ihre Datenbank und stoppen Sie ido2db vor der Ausführung des Patches!

1. /etc/init.d/ido2db stop
2. mysql -u root -p icinga < /path/to/icinga-src/module/idoutils/db/mysql/mysql-upgrade-1.0.sql

Der Patch erledigt das Folgende mit Hilfe von MySQL-Befehlen:

- hinzufügen einer temporären Spalte 'active', um die aktualisierte Zeile zu kennzeichnen
- ermitteln der benötigten Informationen zweier doppelter Zeilen basierend auf dem unique constraint, aktualisieren der zweiten Zeile und markieren durch first=inactive, second=active
- löschen aller als 'inactive' gekennzeichneten Zeilen
- entfernen der fehlerhaften Key-Definitionen
- hinzufügen des Unique Key
- entfernen der temporären Spalte 'active'

Diese Prozedur wird für jede Tabelle durchgeführt, so dass es eine Weile dauern kann, abhängig von Ihren Tabellengrößen und/oder DB-Spezifikationen.

Falls Sie vorher etwas an den Keys verändert haben, dann stellen Sie sicher, dass Sie das gleiche DB-Schema wie in IDOUtils 1.0RC benutzen, andernfalls wird das Script fehlschlagen.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Icinga aktualisieren](#)

[Zum Anfang](#)

[Windows-Maschinen überwachen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Windows-Maschinen überwachen

[Zurück](#)

Kapitel 2. Los geht's

[Weiter](#)

Windows-Maschinen überwachen

Einführung

Dieses Dokument beschreibt, wie Sie "private" Dienste und Attribute von Windows-Rechnern überwachen können, wie z.B.:

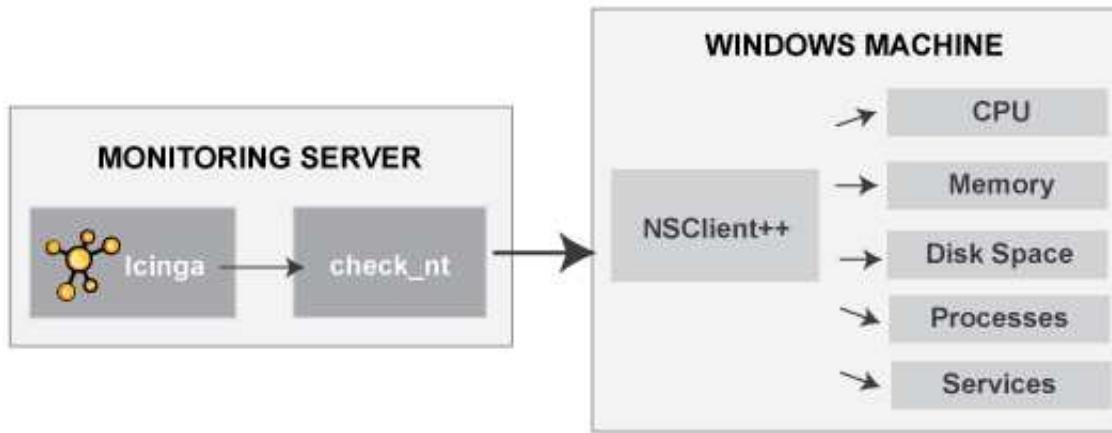
- Speicherbelegung
- CPU-Auslastung
- Plattenbelegung
- Zustände von Diensten
- laufende Prozesse
- etc.

Öffentlich nutzbare Dienste, die von Windows-Rechnern zur Verfügung gestellt werden (HTTP, FTP, POP3, etc.), können einfach mit Hilfe der Dokumentation [öffentliche Dienste überwachen](#) kontrolliert werden.



Anmerkung: Diese Anweisungen gehen davon aus, dass Sie Icinga anhand der [Schnellstartanleitung](#) installiert haben. Die nachfolgenden Beispiel-Konfigurationseinträge beziehen sich auf Objekte, die in den Beispiel-Konfigurationsdateien (*commands.cfg*, *templates.cfg*, etc.) definiert sind. Diese Dateien werden installiert, wenn Sie der Schnellstartanleitung folgen.

Überblick



Die Überwachung von privaten Diensten oder Attributen eines Windows-Rechners erfordert die Installation eines Agenten. Dieser Agent dient als ein Bindeglied zwischen der Überwachung und dem eigentlichen Dienst oder Attribut auf dem Windows-Rechner. Ohne diesen Agenten wäre Icinga nicht in der Lage, private Dienste oder Attribute auf dem Window-Rechner zu überwachen.

Für dieses Beispiel installieren wir das [NSClient++-Addon](#) auf dem Windows-Rechner und werden das *check_nt*-Plugin zur Kommunikation mit dem NSClient++-Addon benutzen. Das *check_nt*-Plugin sollte bereits auf dem Icinga-Server installiert sein, wenn Sie der Schnellstartanleitung gefolgt sind.

Andere Windows-Agenten (wie [NC_Net](#)) können statt NSClient++ genutzt werden, wenn Sie möchten - vorausgesetzt, Sie passen die Befehls- und Service-Definitionen usw. entsprechend an. Aus Gründen der Einfachheit werden wir nur das NSClient++-Addon in diesen Anweisungen berücksichtigen.

Schritte

Es gibt einige Schritte, die Sie durchführen müssen, um einen neuen Windows-Rechner zu überwachen. Das sind:

1. erfüllen Sie einmalige Voraussetzungen
2. installieren Sie einen Überwachungsagenten auf dem Windows-Rechner
3. erstellen Sie neue Host- und Service-Definitionen zur Überwachung des Windows-Rechners
4. starten Sie den Icinga-Daemon neu

Was bereits für Sie vorbereitet wurde

Um Ihnen das Leben ein wenig zu erleichtern, wurden bereits ein paar Konfigurationsaufgaben für Sie erledigt:

- Eine *check_nt*-Befehlsdefinition ist in der *commands.cfg*-Datei vorhanden. Das erlaubt Ihnen die Nutzung des *check_nt*-Plugins zur Überwachung von Windows-Diensten.
- Eine Host-Vorlage für Windows-Server (namens *windows-server*) wurde bereits in der *templates.cfg*-Datei erstellt. Das erlaubt es Ihnen, Windows-Host-Definitionen auf einfache Weise hinzuzufügen.

Die o.g. Konfigurationsdateien finden Sie im `/usr/local/icinga/etc/objects/-Verzeichnis`. Sie können diese und andere Definitionen anpassen, damit Sie Ihren Anforderungen besser entsprechen. Allerdings empfehlen wir Ihnen, noch ein wenig damit zu warten, bis Sie besser mit der Konfiguration von Icinga vertraut sind. Für den Moment folgen Sie einfach den nachfolgenden Anweisungen und Sie werden im Nu Ihre Windows-Rechner überwachen.

Voraussetzungen

Wenn Sie Icinga das erste Mal konfigurieren, um einen Windows-Rechner zu überwachen, dann müssen Sie ein paar zusätzliche Dinge tun. Denken Sie daran, dass Sie dies nur für den *ersten* Windows-Rechner machen müssen, den Sie überwachen wollen.

Editieren Sie die Hauptkonfigurationsdatei.

```
#> vi /usr/local/icinga/etc/icinga.cfg
```

Entfernen Sie das führende Hash-(#)-Zeichen der folgenden Zeile in der Hauptkonfigurationsdatei:

```
#cfg_file=/usr/local/icinga/etc/objects/windows.cfg
```

Speichern Sie die Datei und verlassen den Editor.

Was haben Sie gerade getan? Sie haben Icinga mitgeteilt, in der `/usr/local/icinga/etc/objects/windows.cfg`-Datei nach weiteren Objektdefinitionen zu schauen. Dort werden Sie Host- und Service-Definitionen für Windows-Rechner einfügen. Diese Konfigurationsdatei enthält bereits einige Beispiel-Host-, Hostgroup- und Service-Definitionen. Für den *ersten* Windows-Rechner, den Sie überwachen, passen Sie einfach die Beispiel-Host- und Service-Definitionen an, statt neue zu erstellen.

Installation des Windows-Agenten

Bevor Sie mit der Überwachung von privaten Diensten und Attributen von Windows-Rechnern beginnen, müssen Sie einen Agenten auf diesen Rechnern installieren. Wir empfehlen das NSClient++-Addon zu nutzen, das Sie unter <http://sourceforge.net/projects/nscplus> finden. Diese Anweisungen werden Sie durch eine Basisinstallation des NSClient++-Addons und die Icinga-Konfiguration für die Überwachung des Windows-Rechners führen.

1. Laden Sie die letzte stabile Version des NSClient++-Addons von <http://sourceforge.net/projects/nscplus>
2. Entpacken Sie die NSClient++-Dateien in ein neues C:\NSClient++-Verzeichnis
3. Gehen Sie auf die Kommandozeile und wechseln Sie in das C:\NSClient++-Verzeichnis
4. Registrieren Sie den NSClient++-Dienst mit dem folgenden Befehl:

```
nsclient++ /install
```

5. Öffnen Sie die Dienste-Applikation und stellen Sie sicher, dass der NSClient++-Dienst mit dem Desktop kommunizieren darf (Reiter "Anmelden", Häkchen bei "Datenaustausch zwischen Dienst und Desktop zulassen" gesetzt). Setzen Sie ggf. das Häkchen.



6. Editieren Sie die NSC.INI-Datei (im C:\NSClient++-Verzeichnis) und machen Sie folgende Änderungen:

- entfernen Sie die Kommentarzeichen (;) im [modules]-Abschnitt, außer für CheckWMI.dll und RemoteConfiguration.dll
- definieren Sie optional ein Passwort für Clients, indem Sie die 'password'-Option im [Settings]-Abschnitt setzen.
- entfernen Sie das Kommentarzeichen (;) vor der 'allowed_hosts'-Option im [Settings]-Abschnitt. Fügen Sie die IP-Adresse des Icinga-Servers ein, mit ip.add.ress/Bits einen Bereich oder lassen Sie diese Angabe leer, so dass sich alle Hosts verbinden können.
- entfernen Sie ggf. das Kommentarzeichen vor der 'port'-Option im [NSClient]-Abschnitt und setzen Sie den Wert auf '12489' (Standard).

7. Starten Sie den NSClient++-Dienst mit dem folgenden Befehl:

```
nsclient++ /start
```

8. Geschafft! Der Windows-Rechner kann nun der Icinga-Überwachungskonfiguration hinzugefügt werden...

Icinga konfigurieren

Nun ist es Zeit, einige [Objektdefinitionen](#) in Ihren Icinga-Konfigurationsdateien anzulegen, um den neuen Windows-Rechner zu überwachen.

Editieren Sie die *windows.cfg*-Datei.

```
#> vi /usr/local/icinga/etc/objects/windows.cfg
```

Fügen Sie eine neue [Host](#)-Definition für den Windows-Rechner hinzu, den Sie überwachen möchten. Wenn dies der *erste* Windows-Rechner ist, den Sie überwachen, dann können Sie einfach die Beispiel-Definitionen in der *windows.cfg*-Datei anpassen. Ändern Sie die *host_name*-,

alias- und *address*-Felder auf die entsprechenden Werte des Windows-Rechners.

```
define host{
    ; Standard-Werte von einer Windows-Server-Vorlage erben
    use           windows-server ; diese Zeile nicht löschen!
    host_name     winserver
    alias         My Windows Server
    address       192.168.1.2
}
```

Gut. Nun können Sie (in der gleichen Konfigurationsdatei) einige Service-Definitionen hinzufügen, um Icinga mitzuteilen, welche Dinge auf dem Windows-Server zu überwachen sind. Wenn dies der *erste* Windows-Rechner ist, den Sie überwachen, dann können Sie einfach die Beispiel-Definitionen in der *windows.cfg*-Datei anpassen.



Anmerkung

Ersetzen Sie "winserver" in den folgenden Beispiel-Definitionen durch den Namen, den Sie in der *host_name*-Direktive der Host-Definitionen angegeben haben, die Sie gerade hinzugefügt haben.

Fügen Sie die folgende Service-Definition hinzu, um die Version des NSClient++-Addons zu überwachen, das auf dem Windows-Rechner läuft. Dies ist nützlich, wenn Sie Ihre Windows-Server mit einer neueren Version des Addons aktualisieren möchten, weil Sie sehen können, welche Windows-Rechner noch auf die neueste Version des NSClient++-Addon aktualisiert werden muss.

```
define service{
    use           generic-service
    host_name     winserver
    service_description NSClient++ Version
    check_command  check_nt!CLIENTVERSION
}
```

Fügen Sie die folgende Service-Definition hinzu, um die Laufzeit des Windows-Servers zu überwachen.

```
define service{
    use           generic-service
    host_name     winserver
    service_description Uptime
    check_command  check_nt!UPTIME
}
```

Fügen Sie die folgende Service-Definition hinzu, um die CPU-Belastung des Windows-Servers zu überwachen und einen CRITICAL-Alarm zu erzeugen, wenn die 5-Minuten-Belastung mindestens 90% beträgt oder einen WARNING-Alarm, wenn die 5-Minuten-Belastung mindestens 80% beträgt.

```
define service{
    use           generic-service
    host_name     winserver
    service_description CPU Load
    check_command  check_nt!CPULOAD!-1 5,80,90
}
```

Fügen Sie die folgende Service-Definition hinzu, um die Speicherbelegung des Windows-Servers zu überwachen und einen CRITICAL-Alarm zu erzeugen, wenn die Belegung mindestens 90% beträgt oder einen WARNING-Alarm, wenn die Belegung mindestens 80% beträgt.

```
define service{
    use          generic-service
    host_name   winserver
    service_description Memory Usage
    check_command  check_nt!MEMUSE! -w 80 -c 90
}
```

Fügen Sie die folgende Service-Definition hinzu, um die Plattenbelegung von Laufwerk C: des Windows-Servers zu überwachen und einen CRITICAL-Alarm zu erzeugen, wenn die Belegung mindestens 90% beträgt oder einen WARNING-Alarm, wenn die Belegung mindestens 80% beträgt.

```
define service{
    use          generic-service
    host_name   winserver
    service_description C:\ Drive Space
    check_command  check_nt!USEDISKSPACE! -l c -w 80 -c 90
}
```

Fügen Sie die folgende Service-Definition hinzu, um den W3SVC-Dienst des Windows-Servers zu überwachen und einen CRITICAL-Alarm zu erzeugen, wenn der Dienst gestoppt ist.

```
define service{
    use          generic-service
    host_name   winserver
    service_description W3SVC
    check_command  check_nt!SERVICESTATE! -d SHOWALL -l W3SVC
}
```

Fügen Sie die folgende Service-Definition hinzu, um den Explorer.exe-Prozess des Windows-Servers zu überwachen und einen CRITICAL-Alarm zu erzeugen, wenn der Prozess nicht läuft.

```
define service{
    use          generic-service
    host_name   winserver
    service_description Explorer
    check_command  check_nt!PROCSTATE! -d SHOWALL -l Explorer.exe
}
```



Anmerkung

Nun ja. Eigentlich ist es ziemlich unsinnig, zu überwachen, ob der Explorer läuft. Allerdings lässt sich auf diese Weise sehr einfach prüfen, ob alles wie gewünscht funktioniert ;-)

Das war es vorerst. Sie haben einige grundlegende Dienste hinzugefügt, die auf dem Windows-Rechner überwacht werden sollen. Speichern Sie die Konfigurationsdatei.

Passwortschutz

Wenn Sie ein Passwort in der NSClient++-Konfigurationsdatei auf dem Windows-Rechner angegeben haben, dann müssen Sie die *check_nt*-Befehlsdefinition anpassen, damit sie das Passwort enthält. Öffnen Sie die *commands.cfg*-Datei.

```
#> vi /usr/local/icinga/etc/objects/commands.cfg
```

Ändern Sie die Definition des *check_nt*-Befehls, damit sie das "-s <PASSWORD>"-Argument enthält (wobei PASSWORD das Passwort ist, das Sie auf dem Windows-Rechner angegeben haben):

```
define command{
    command_name      check_nt
    command_line      $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -s $PASSWORD -v $ARG1$ $ARG2$}
```

Speichern Sie die Datei

Icinga neu starten

Sie sind fertig mit der Anpassung der Icinga-Konfiguration, so dass Sie nun [die Konfigurationsdateien überprüfen](#) und [Icinga neu starten](#) müssen.

Wenn die Überprüfung irgendwelche Fehler enthält, dann müssen Sie diese beheben, bevor Sie fortfahren. Stellen Sie sicher, dass Sie Icinga nicht (erneut) starten, bevor die Überprüfung ohne Fehler durchgelaufen ist!

[Zurück](#)[Nach oben](#)[Weiter](#)[IDOUtils-Datenbank aktualisieren](#)[Zum Anfang](#)[Linux/Unix-Rechner überwachen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Linux/Unix-Rechner überwachen

[Zurück](#)

Kapitel 2. Los geht's

[Weiter](#)

Linux/Unix-Rechner überwachen

Einführung

Dieses Dokument beschreibt, wie Sie "private" Dienste und Attribute auf Linux/UNIX-Servern überwachen, wie z.B.:

- CPU-Auslastung
- Speichernutzung
- Plattenbelegung
- angemeldete Benutzer
- laufende Prozesse
- etc.

Öffentlich nutzbare Dienste, die von Linux-Servern zur Verfügung gestellt werden (HTTP, FTP, SSH, SMTP, etc.), können einfach mit Hilfe der Dokumentation [öffentliche zugängliche Dienste überwachen](#) kontrolliert werden.

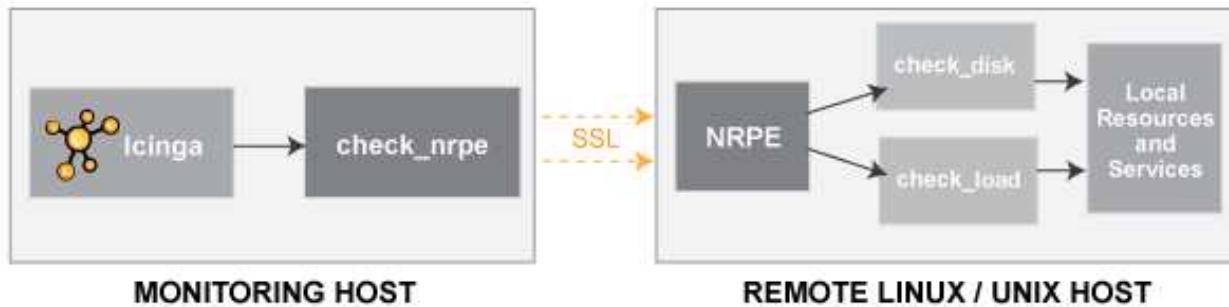


Anmerkung: Diese Anweisungen gehen davon aus, dass Sie Icinga anhand der [Schnellstartanleitung](#) installiert haben. Die nachfolgenden Beispiel-Konfigurationseinträge beziehen sich auf Objekte, die in den Beispiel-Konfigurationsdateien (*commands.cfg*, *templates.cfg*, etc.) definiert sind. Diese Dateien werden installiert, wenn Sie der Schnellstartanleitung folgen.

Überblick

[Anmerkung: Dieses Dokument ist noch nicht vollständig. Wir würden empfehlen, die Dokumentation zum [NRPE-Addon](#) zu lesen, um zu sehen, wie ein entfernter Linux/Unix-Server zu überwachen ist.]

Es gibt verschiedene Wege, Attribute oder entfernte Linux/Unix-Server zu überwachen. Einer benutzt gemeinsame SSH-Schlüssel und das *check_by_ssh*-Plugin auf entfernten Servern. Diese Methode wird hier nicht behandelt, kann aber zu hoher Last auf Ihrem Überwachungs-Server führen, wenn Sie hunderte oder tausende von Services überwachen. Der Overhead durch das Auf- und Abbauen von SSH-Verbindungen ist der Grund dafür.



Eine andere gebräuchliche Methode der Überwachung von entfernten Linux/Unix-Hosts ist die Nutzung des [NRPE-Addons](#). NRPE erlaubt Ihnen, Plugins auf entfernten Linux/Unix-Hosts auszuführen. Das ist nützlich, wenn Sie lokale Ressourcen/Attribute wie z.B. Plattenbelegung, CPU-Auslastung, Speichernutzung auf einem entfernten Host überwachen wollen.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Windows-Maschinen überwachen](#)

[Zum Anfang](#)

[Netware-Server überwachen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Netware-Server überwachen

[Zurück](#)[Kapitel 2. Los geht's](#)[Weiter](#)

Netware-Server überwachen

Einführung

Dieses Dokument enthält Informationen, wie Sie Novell-Netware-Servern überwachen können.

Externe Ressourcen

Sie finden Informationen zur Überwachung von Netware-Servern mit Icinga auf der Novell-[Cool Solutions](#)-Website, darunter:

- [MRTGEXT: NLM module for MRTG and Nagios](#)
- [Nagios: Host and Service Monitoring Tool](#)
- [Nagios and NetWare: SNMP-based Monitoring](#)
- [Monitor DirXML/IDM Driver States with Nagios](#)
- [Check NDS Login ability with Nagios](#)
- [NDPS/iPrint Print Queue Monitoring by Nagios](#)
- [check_gwiaRL Plugin for Nagios 2.0](#)



Hinweis: Wenn Sie Novells [Cool Solutions](#)-Site besuchen, suchen Sie nach "Nagios", um mehr Artikel und Software-Komponenten zu finden, die sich auf Überwachung beziehen.

Dank an [Christian Mies](#), [Rainer Brunold](#) und andere, die auf der Novell-Site Nagios- und Netware-Dokumentation, Addons usw. beigetragen haben!

[Zurück](#)[Nach oben](#)[Weiter](#)[Linux/Unix-Rechner überwachen](#)[Zum Anfang](#)[Netzwerk-Drucker überwachen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Netzwerk-Drucker überwachen

[Zurück](#)[Kapitel 2. Los geht's](#)[Weiter](#)

Netzwerk-Drucker überwachen

Einführung



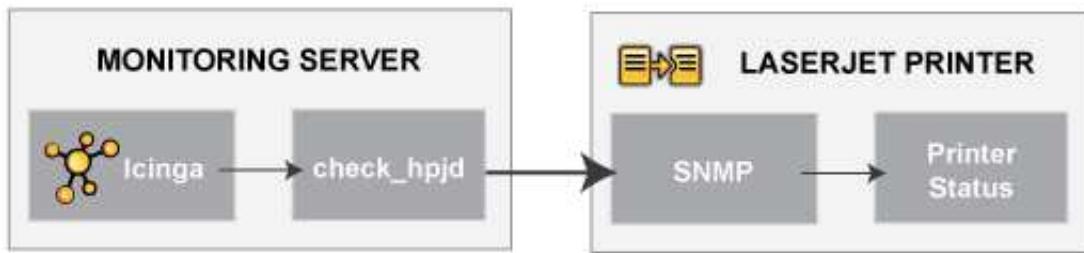
Dieses Dokument beschreibt, wie Sie den Status von Netzwerkdruckern überwachen können. HP-Drucker haben interne/externe JetDirect-Karten/Devices, andere Print-Server (wie der Troy PocketPro 100S oder der Netgear PS101) unterstützen das JetDirect-Protokoll.

Das *check_hpjd*-Plugin (das Bestandteil der Icinga-Distribution ist), erlaubt Ihnen die Überwachung des Zustands von JetDirect-fähigen Druckern, auf denen SNMP aktiviert ist. Das Plugin kann die folgenden Druckerzustände erkennen:

- Papierstau
- Kein Papier mehr
- Drucker Offline
- Benutzereingriff erforderlich
- Tonerstand niedrig
- Speicher unzureichend
- Klappe offen
- Ausgabefach voll
- und weitere...

 Anmerkung: Diese Anweisungen gehen davon aus, dass Sie Icinga anhand der [Schnellstartanleitung](#) installiert haben. Die nachfolgenden Beispiel-Konfigurationseinträge beziehen sich auf Objekte, die in den Beispiel-Konfigurationsdateien (*commands.cfg*, *templates.cfg*, etc.) definiert sind. Diese Dateien werden installiert, wenn Sie der Schnellstartanleitung folgen.

Überblick



Die Überwachung des Zustands eines Netzwerkdruckers ist ziemlich einfach. Bei JetDirect-fähigen Druckern ist normalerweise SNMP aktiviert, so dass Icinga ihren Zustand mit Hilfe des *check_hpjd*-Plugins überwachen kann.



Wichtig

Das *check_hpjd*-Plugin wird nur dann kompiliert und installiert, wenn Sie die net-snmp- und net-snmp-utils-Pakete auf Ihrem System haben. Stellen Sie sicher, dass das Plugin im `/usr/local/icinga/libexec`-Verzeichnis existiert, bevor Sie fortfahren. Falls nicht, installieren Sie net-snmp und net-snmp-utils und kompilieren und installieren Sie die Icinga-Plugins erneut, nachdem Sie "make clean" im Source-Verzeichnis ausgeführt haben. Einzelheiten finden Sie in der [Schnellstartanleitung](#).

Schritte

Es gibt einige Schritte, die Sie durchführen müssen, um einen neuen Netzwerkdrucker zu überwachen. Das sind:

1. erfüllen Sie einmalige Voraussetzungen
2. erstellen Sie neue Host- und Service-Definitionen zur Überwachung des Druckers
3. starten Sie den Icinga-Daemon neu

Was bereits für Sie vorbereitet wurde

Um Ihnen das Leben ein wenig zu erleichtern, wurden bereits ein paar Konfigurationsaufgaben für Sie erledigt:

- Eine *check_hpjd*-Befehlsdefinition ist in der `commands.cfg`-Datei vorhanden. Das erlaubt Ihnen die Nutzung des *check_hpjd*-Plugins zur Überwachung von Netzwerkdruckern.
- Eine Host-Vorlage für Drucker (namens `generic-printer`) wurde bereits in der `templates.cfg`-Datei erstellt. Das erlaubt es Ihnen, Drucker-Host-Definitionen auf einfache Weise hinzuzufügen.

Die o.g. Konfigurationsdateien finden Sie im `/usr/local/icinga/etc/objects/-Verzeichnis`. Sie können diese und andere Definitionen anpassen, damit Sie Ihren Anforderungen besser entsprechen. Allerdings empfehlen wir Ihnen, noch ein wenig damit zu warten, bis Sie besser mit der Konfiguration von Icinga vertraut sind. Für den Moment folgen Sie einfach den nachfolgenden Anweisungen und Sie werden im Nu Ihre Netzwerkdrucker überwachen.

Voraussetzungen

Wenn Sie Icinga das erste Mal konfigurieren, um einen Netzwerkdrucker zu überwachen, dann müssen Sie ein paar zusätzliche Dinge tun. Denken Sie daran, dass Sie dies nur für den *ersten* Netzwerkdrucker machen müssen, den Sie überwachen wollen.

Editieren Sie die Hauptkonfigurationsdatei.

```
vi /usr/local/icinga/etc/nagios.cfg
```

Entfernen Sie das führende Hash-(#)-Zeichen der folgenden Zeile in der Hauptkonfigurationsdatei:

```
#cfg_file=/usr/local/icinga/etc/objects/printer.cfg
```

Speichern Sie die Datei und verlassen den Editor.

Was haben Sie gerade getan? Sie haben Icinga mitgeteilt, in der */usr/local/icinga/etc/objects/printer.cfg*-Datei nach weiteren Objektdefinitionen zu schauen. Dort werden Sie Drucker-Host- und Service-Definitionen einfügen. Diese Konfigurationsdatei enthält bereits einige Beispiel-Host-, Hostgroup- und Service-Definitionen. Für den *ersten* Netzwerkdrucker, den Sie überwachen, passen Sie einfach die Beispiel-Host- und Service-Definitionen an, statt neue zu erstellen.

Icinga konfigurieren

Sie müssen einige [Objektdefinitionen anlegen](#), um einen neuen Drucker zu überwachen.

Öffnen Sie die *printer.cfg*-Datei.

```
vi /usr/local/icinga/etc/objects/printer.cfg
```

Fügen Sie eine neue [Host](#)-Definition für den Netzwerkdrucker hinzu, den Sie überwachen möchten. Wenn dies der *erste* Netzwerkdrucker ist, den Sie überwachen, dann können Sie einfach die Beispiel-Definitionen in der *printer.cfg*-Datei anpassen. Ändern Sie die *host_name*-, *alias*- und *address*-Felder auf die entsprechenden Werte des Netzwerkdruckers.

```
define host{
    use          generic-printer ; Inherit default values from a template
    host_name    hplj2605dn      ; The name we're giving to this printer
    alias        HP LaserJet 2605dn ; A longer name associated with the printer
    address      192.168.1.30     ; IP address of the printer
    hostgroups   allhosts        ; Host groups this printer is associated with
}
```

Nun können Sie (in der gleichen Konfigurationsdatei) einige Service-Definitionen hinzufügen, um Icinga mitzuteilen, welche Dinge auf dem Drucker zu überwachen sind. Wenn dies der *erste* Drucker ist, den Sie überwachen, dann können Sie einfach die Beispiel-Definitionen in der *printer.cfg*-Datei anpassen.

 Anmerkung: Ersetzen Sie "hplj2605dn" in der folgenden Beispiel-Definition durch den Namen, den Sie in der *host_name*-Direktive der Host-Definition angegeben haben, die Sie gerade hinzugefügt haben.

Fügen Sie die folgende Service-Definition hinzu, um den Zustand des Druckers zu prüfen. Der Service benutzt das *check_hpjd*-Plugin, um den Drucker alle zehn Minuten zu prüfen. Der Wert für die SNMP-Community lautet in diesem Beispiel "public".

```
define service{
    use generic-service ; Inherit values from a template
    host_name hplj2605dn ; The name of the host the service is associated with
    service_description Printer Status ; The service description
    check_command check_hpjd!-C public ; The command used to monitor the service
    check_interval 10 ; Check the service every 10 minutes under normal conditions
    retry_interval 1 ; Re-check every minute until its final/hard state is determined
}
```

Fügen Sie die folgende Service-Definition hinzu, um alle zehn Minuten einen Ping an den Drucker zu senden. Das ist nützlich, um die generelle Netzwerkverbindung und Werte für RTA und Paketverlust zu überwachen.

```
define service{
    use generic-service
    host_name hplj2605dn
    service_description PING
    check_command check_ping!3000.0,80%!5000.0,100%
    check_interval 10
    retry_interval 1
}
```

Speichern Sie die Datei.

Icinga neu starten

Sobald Sie die neuen Host- und Service-Definitionen in der *printer.cfg*-Datei hinzugefügt haben, sind Sie bereit, mit der Überwachung des Druckers zu beginnen. Um dies zu tun, müssen Sie [die Konfigurationsdateien überprüfen](#) und [Icinga neu starten](#).

Wenn die Überprüfung irgendwelche Fehler enthält, dann müssen Sie diese beheben, bevor Sie fortfahren. Stellen Sie sicher, dass Sie Icinga nicht (erneut) starten, bevor die Überprüfung ohne Fehler durchgelaufen ist!

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Netware-Server überwachen

[Zum Anfang](#)

Router und Switches überwachen

© 2009-2011 Icinga Development Team, <http://www.icinga.org>

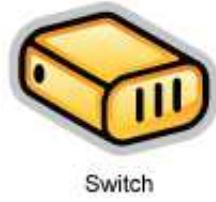


Router und Switches überwachen

[Zurück](#)[Kapitel 2. Los geht's](#)[Weiter](#)

Router und Switches überwachen

Einführung



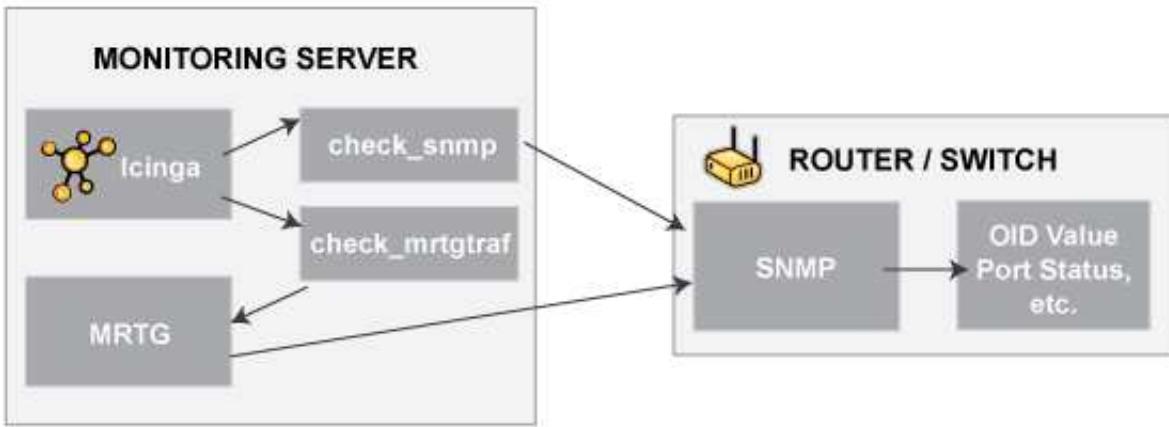
Dieses Dokument beschreibt, wie Sie den Zustand von Netzwerk-Switches und Routern überwachen können. Einige preiswerte "unmanaged" Switches und Router haben keine IP-Adresse und sind in Ihrem Netzwerk nicht sichtbar, so dass es keinen Weg gibt, um sie zu überwachen. Teurere Switches und Router haben eigene Adressen und können durch Ping überwacht oder über SNMP nach Statusinformationen abgefragt werden.

Ich werde beschreiben, wie Sie die folgenden Dinge auf "managed" Switches, Hubs und Routern überwachen können:

- Paketverlust, durchschnittliche Umlaufzeiten (round trip average, RTA)
- SNMP-Statusinformationen
- Bandbreite / Übertragungsrate (traffic rate)

 Anmerkung: Diese Anweisungen gehen davon aus, dass Sie Icinga anhand der [Schnellstartanleitung](#) installiert haben. Die nachfolgenden Beispiel-Konfigurationseinträge beziehen sich auf Objekte, die in den Beispiel-Konfigurationsdateien (*commands.cfg*, *templates.cfg*, etc.) definiert sind. Diese Dateien werden installiert, wenn Sie der Schnellstartanleitung folgen.

Überblick



Die Überwachung von Switches und Routern kann entweder einfach oder auch aufwändiger sein - abhängig davon, welches Equipment Sie haben und was Sie überwachen wollen. Da es sich um kritische Infrastrukturkomponenten handelt, werden Sie diese ohne Zweifel mindestens in grundlegender Art und Weise überwachen.

Switches und Router können einfach per "Ping" überwacht werden, um Paketverlust, RTA usw. zu ermitteln. Wenn Ihr Switch SNMP unterstützt, können Sie mit dem *check_snmp*-Plugin z.B. den Port-Status und (wenn Sie MRTG benutzen) mit dem *check_mrtgtraf*-Plugin die Bandbreite überwachen.

Das *check_snmp*-Plugin wird nur dann kompiliert und installiert, wenn Sie die net-snmp- und net-snmp-utils-Pakete auf Ihrem System haben. Stellen Sie sicher, dass das Plugin im */usr/local/icinga/libexec*-Verzeichnis existiert, bevor Sie fortfahren. Falls nicht, installieren Sie net-snmp und net-snmp-utils und kompilieren und installieren Sie die Icinga-Plugins erneut.

Schritte

Es gibt einige Schritte, die Sie durchführen müssen, um einen neuen Router oder Switch zu überwachen. Das sind:

1. erfüllen Sie einmalige Voraussetzungen
2. erstellen Sie neue Host- und Service-Definitionen zur Überwachung des Geräts
3. starten Sie den Icinga-Daemon neu

Was bereits für Sie vorbereitet wurde

Um Ihnen das Leben ein wenig zu erleichtern, wurden bereits ein paar Konfigurationsaufgaben für Sie erledigt:

- Zwei Befehlsdefinitionen (*check_snmp* und *check_local_mrtgtraf*) sind bereits in der *commands.cfg*-Datei vorhanden. Das erlaubt Ihnen die Nutzung des *check_snmp*- bzw. *check_mrtgtraf*-Plugins zur Überwachung von Routern und Switches.
- Eine Host-Vorlage für Switches (namens *generic-switch*) wurde bereits in der *templates.cfg*-Datei erstellt. Das erlaubt es Ihnen, Router/Switch-Host-Definitionen auf einfache Weise hinzuzufügen.

Die o.g. Konfigurationsdateien finden Sie im */usr/local/icinga/etc/objects*-Verzeichnis. Sie können diese und andere Definitionen anpassen, damit Sie Ihren Anforderungen besser entsprechen. Allerdings empfehlen wir Ihnen, noch ein wenig damit zu warten, bis Sie besser mit der Konfiguration von Icinga vertraut sind. Für den Moment folgen Sie einfach den nachfolgenden

Anweisungen und Sie werden im Nu Ihre Router/Switches überwachen.

Voraussetzungen

Wenn Sie Icinga das erste Mal konfigurieren, um einen Netzwerk-Switch zu überwachen, dann müssen Sie ein paar zusätzliche Dinge tun. Denken Sie daran, dass Sie dies nur für den **ersten** Switch machen müssen, den Sie überwachen wollen.

Editieren Sie die Hauptkonfigurationsdatei.

```
#> vi /usr/local/icinga/etc/nagios.cfg
```

Entfernen Sie das führende Hash-(#)-Zeichen der folgenden Zeile in der Hauptkonfigurationsdatei:

```
#cfg_file=/usr/local/icinga/etc/objects/switch.cfg
```

Speichern Sie die Datei und verlassen den Editor.

Was haben Sie gerade getan? Sie haben Icinga mitgeteilt, in der */usr/local/icinga/etc/objects/switch.cfg*-Datei nach weiteren Objektdefinitionen zu schauen. Dort werden Sie Host- und Service-Definitionen für Router- und Switches einfügen. Diese Konfigurationsdatei enthält bereits einige Beispiel-Host-, Hostgroup- und Service-Definitionen. Für den **ersten** Router/Switch, den Sie überwachen, passen Sie einfach die Beispiel-Host- und Service-Definitionen an, statt neue zu erstellen.

Icinga konfigurieren

Sie müssen einige [Objektdefinitionen](#) anlegen, um einen neuen Router/Switch zu überwachen.

Öffnen Sie die *switch.cfg*-Datei.

```
#> vi /usr/local/icinga/etc/objects/switch.cfg
```

Fügen Sie eine neue [Host](#)-Definition für den Switch hinzu, den Sie überwachen möchten. Wenn dies der **erste** Switch ist, den Sie überwachen, dann können Sie einfach die Beispiel-Definitionen in der *switch.cfg*-Datei anpassen. Ändern Sie die *host_name*-, *alias*- und *address*-Felder auf die entsprechenden Werte des Switches.

```
define host{
    use          generic-switch           ; Inherit default values from a template
    host_name    linksys-srw224p         ; The name we're giving to this switch
    alias        Linksys SRW224P Switch   ; A longer name associated with the switch
    address      192.168.1.253           ; IP address of the switch
    hostgroups   allhosts,switches       ; Host groups this switch is associated with
}
```

Services überwachen

Nun können Sie einige Service-Definitionen hinzufügen (in der gleichen Konfigurationsdatei), um Icinga mitzuteilen, welche Dinge auf dem Switch zu überwachen sind. Wenn dies der **erste** Switch ist, den Sie überwachen, dann können Sie einfach die Beispiel-Definitionen in der *switch.cfg*-Datei anpassen.

 Anmerkung: Ersetzen Sie "linksys-srw224p" in der folgenden Beispiel-Definition durch den Namen, den Sie in der *host_name*-Direktive der Host-Definition angegeben haben, die Sie gerade hinzugefügt haben.

Paketverlust und RTA überwachen

Fügen Sie die folgende Service-Definition hinzu, um unter normalen Bedingungen alle fünf Minuten Paketverlust und Round-Trip-Average zwischen dem Icinga-Host und dem Switch zu überwachen.

```
define service{
    use generic-service ; Inherit values from a template
    host_name linksys-srw224p ; The name of the host the service is associated with
    service_description PING ; The service description
    check_command check_ping!200.0,20%!600.0,60% ; The command used to monitor the service
    check_interval 5 ; Check the service every 5 minutes under normal conditions
    retry_interval 1 ; Re-check every minute until its final/hard state is determined
}
```

Dieser Service wird:

- CRITICAL, falls der Round-Trip-Average (RTA) größer als 600 Millisekunden oder der Paketverlust 60% oder mehr ist
- WARNING, falls der Round-Trip-Average (RTA) größer als 200 Millisekunden oder der Paketverlust 20% oder mehr ist
- OK, falls der Round-Trip-Average (RTA) kleiner als 200 Millisekunden oder der Paketverlust kleiner als 20% ist

SNMP-Statusinformationen überwachen

Wenn Ihr Switch oder Router SNMP unterstützt, können Sie eine Menge an Informationen mit dem *check_snmp*-Plugin überwachen. Wenn nicht, dann überspringen Sie diesen Abschnitt.

Fügen Sie die folgende Service-Definition hinzu, um die Laufzeit des Switches zu überwachen.

```
define service{
    use generic-service ; Inherit values from a template
    host_name linksys-srw224p
    service_description Uptime
    check_command check_snmp!-C public -o sysUpTime.0
}
```

In der *check_command*-Direktive der obigen Service-Definition sagt "-C public", dass der zu benutzende SNMP-Community-Name "public" lautet und "-o sysUpTime.0" gibt an, welche OID überprüft werden soll.

Wenn Sie sicherstellen wollen, dass sich ein bestimmter Port/ein bestimmtes Interface des Switches in einem "UP"-Zustand befindet, dann sollten Sie eine Service-Definition hinzufügen:

```
define service{
    use generic-service ; Inherit values from a template
    host_name linksys-srw224p
    service_description Port 1 Link Status
    check_command check_snmp!-C public -o ifOperStatus.1 -r 1 -m RFC1213-MIB
}
```

In dem obigen Beispiel bezieht sich "-o ifOperStatus.1" auf die OID des Betriebszustands von Port 1 des Switches. Die "-r 1"-Option teilt dem *check_snmp*-Plugin mit, einen OK-Zustand zurückzuliefern, wenn "1" im SNMP-Ergebnis gefunden wird (1 deutet einen "UP"-Zustand des Ports an) und CRITICAL, wenn es nicht gefunden wird. "-m RFC1213-MIB" ist optional und teilt dem *check_snmp*-Plugin mit, nur die "RFC1213-MIB" zu laden statt jeder einzelnen MIB, die auf Ihrem System installiert ist, was die Dinge beschleunigen kann.

Das war's mit dem SNMP-Überwachungsbeispiel. Es gibt eine Million Dinge, die mit SNMP überwacht werden können, also liegt es an Ihnen zu entscheiden, was Sie brauchen und was Sie überwachen wollen. Viel Erfolg!



Hinweis: Normalerweise können Sie mit dem folgenden Befehl die OIDs eines Switches (oder eines anderen SNMP-fähigen Gerätes) herausfinden, die überwacht werden können (ersetzen Sie 192.168.1.253 durch die IP-Adresse des Switches): `snmpwalk -v1 -c public 192.168.1.253 -m ALL .1`

Bandbreite / Übertragungsrate überwachen

Wenn Sie die Bandbreitennutzung Ihres Switches oder Routers mit **MRTG** überwachen, dann können Sie durch Icinga alarmiert werden, wenn die Übertragungsraten Schwellwerte überschreiten, die Sie angeben. Mit dem `check_mrtgtraf`-Plugin (das in der Icinga-Plugin-Distribution enthalten ist) können Sie das tun.

Sie müssen dem `check_mrtgtraf`-Plugin mitteilen, in welcher Log-Datei die MRTG-Daten gespeichert sind, zusammen mit Schwellwerten, usw. In unserem Beispiel überwachen wir einen Port eines Linksys-Switches. Die MRTG-Log-Datei ist abgelegt unter `/var/lib/mrtg/192.168.1.253_1.log`. Hier ist die Service-Definition, die wir benutzen, um die Bandbreitendaten zu überwachen, die in der Log-Datei gespeichert sind...

```
define service{
    use generic-service ; Inherit values from a template
    host_name linksys-srw224p
    service_description Port 1 Bandwidth Usage
    check_command check_local_mrtgtraf!'/var/lib/mrtg/192.168.1.253_1.log!AVG!1000000,2000000!5000000,5000000!10
}
```

In dem obigen Beispiel teilt `/var/lib/mrtg/192.168.1.253_1.log` im `check_local_mrtgtraf`-Befehl dem Plugin mit, welche MRTG-Log-Datei auszulesen ist. Die "AVG"-Option gibt an, dass Durchschnitts-Bandbreitenstatistiken verwendet werden sollen. "1000000,2000000" sind die Schwellwerte (in Bytes) für Warnungen bei eingehenden Übertragungsraten. "5000000,5000000" sind die kritischen Schwellwerte (in Bytes) bei ausgehenden Übertragungsraten. "10" gibt an, dass das Plugin einen CRITICAL-Zustand zurückliefern soll, wenn die MRTG-Log-Datei älter als zehn Minuten ist (sie sollte alle fünf Minuten aktualisiert werden).

Speichern Sie die Datei.

Icinga neu starten

Sobald Sie die neuen Host- und Service-Definitionen in der `switch.cfg`-Datei hinzugefügt haben, sind Sie bereit, mit der Überwachung des Routers/Switches zu beginnen. Um dies zu tun, müssen Sie [die Konfigurationsdateien überprüfen](#) und [Icinga neu starten](#).

Wenn die Überprüfung irgendwelche Fehler enthält, dann müssen Sie diese beheben, bevor Sie fortfahren. Stellen Sie sicher, dass Sie Icinga nicht (erneut) starten, bevor die Überprüfung ohne Fehler durchgelaufen ist!

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Netzwerk-Drucker überwachen](#)

[Zum Anfang](#)

[Öffentlich zugängliche Dienste überwachen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Öffentlich zugängliche Dienste überwachen

[Zurück](#)

[Kapitel 2. Los geht's](#)

[Weiter](#)

Öffentlich zugängliche Dienste überwachen

Einführung

Dieses Dokument beschreibt, wie Sie öffentlich zugängliche Dienste, Applikationen und Protokolle überwachen können. Mit "öffentliche" meinen wir Dienste, die über das Netzwerk zugänglich sind - entweder das lokale Netzwerk oder das größere Internet. Beispiele von öffentlichen Diensten umfassen u.a. HTTP, POP3, IMAP, FTP und SSH. Es gibt viele weitere öffentliche Dienste, die Sie wahrscheinlich jeden Tag benutzen. Diese Dienste und Applikationen, genau wie ihre zu Grunde liegenden Protokolle, können normalerweise mit Icinga ohne spezielle Zugangsvoraussetzungen überwacht werden.

Private Dienste können im Gegensatz dazu nicht ohne einen dazwischen geschalteten Agenten überwacht werden. Beispiele von mit Hosts verbundenen privaten Diensten sind Dinge wie CPU-Auslastung, Speicherbelegung, Plattenbelegung, angemeldete Benutzer, Prozessinformationen usw. Diese privaten Dienste oder Attribute von Hosts werden normalerweise nicht an externe Clients offengelegt. Diese Situation erfordert, dass ein zwischengeschalteter Überwachungsagent auf jedem Host installiert wird, den Sie überwachen müssen. Mehr Informationen zur Überwachung von privaten Diensten auf verschiedenen Arten von Hosts finden Sie in der Dokumentation zu:

- [Windows-Rechner überwachen](#)
- [Netware-Server überwachen](#)
- [Linux/Unix-Rechner überwachen](#)



Hinweis: Gelegentlich werden Sie feststellen, dass Informationen zu privaten Diensten und Applikationen mit SNMP überwacht werden können. Der SNMP-Agent erlaubt Ihnen, entfernt liegende anderenfalls private (und unzugängliche) Informationen des Hosts zu überwachen. Mehr Informationen zur Überwachung von Diensten mit SNMP finden Sie in der Dokumentation zur [Überwachung von Switches und Routern](#).



Anmerkung: Diese Anweisungen gehen davon aus, dass Sie Icinga anhand der [Schnellstartanleitung](#) installiert haben. Die nachfolgenden Beispiel-Konfigurationseinträge beziehen sich auf Objekte, die in den Beispiel-Konfigurationsdateien (`commands.cfg` und `localhost.cfg`) definiert sind. Diese Dateien werden installiert, wenn Sie der Schnellstartanleitung folgen.

Plugins zur Überwachung von Services

Wenn Sie feststellen, dass Sie eine bestimmte Applikation, einen Service oder ein Protokoll überwachen müssen, dann stehen die Chancen gut, dass bereits ein [Plugin](#) existiert. Die offizielle Icinga-Plugin-Distribution enthält Plugins, mit denen eine Reihe von Services und Protokollen überwacht werden können. Es gibt auch eine große Zahl von Plugins, die andere Leute beigetragen haben, die Sie im *contrib*-Unterverzeichnis der Plugin-Distribution finden. Die [IcingaExchange.org](#)-Website stellt eine Reihe von zusätzlichen Plugins bereit, die andere Benutzer geschrieben haben, also schauen Sie vorbei, wenn Sie Zeit finden.

Wenn Sie zufällig kein entsprechendes Plugin für das finden, was Sie überwachen möchten, dann können Sie immer Ihr eigenes schreiben. Plugins sind einfach zu schreiben, also lassen Sie sich nicht von diesem Gedanken abschrecken. Lesen Sie dazu die Dokumentation über die [Entwicklung von Plugins](#).

Ich werde Sie durch die Überwachung von einigen grundlegenden Diensten führen, die Sie vielleicht früher oder später brauchen. Jeder dieser Services kann mit einem der Plugins überwacht werden, die als Teil der Icinga-Plugin-Distribution installiert werden. Lassen Sie uns beginnen...

erstellen einer Host-Definition

Bevor Sie einen Service überwachen können, müssen Sie einen [Host](#) definieren, der mit dem Service verbunden ist. Sie können Host-Definitionen in jeder Objektkonfigurationsdatei platzieren, die mit einer [cfg_file](#)-Direktive definiert ist oder in einem Verzeichnis, das in einer [cfg_dir](#)-Direktive angegeben ist. Wenn Sie bereits eine Host-Definition angelegt haben, dann können Sie diesen Schritt überspringen.

Lassen Sie uns für dieses Beispiel annehmen, dass Sie eine Reihe von Services auf einem entfernten Host überwachen wollen. Lassen Sie uns diesen Host *remotehost* nennen. Die Host-Definition kann in einer eigenen Datei abgelegt oder zu einer bereits existierenden Objektkonfigurationsdatei hinzugefügt werden. Hier nun, wie die Host-Definition für *remotehost* aussehen könnte:

```
define host{
    use          generic-host           ; Inherit default values from a template
    host_name    remotehost            ; The name we're giving to this host
    alias        Some Remote Host      ; A longer name associated with the host
    address      192.168.1.50          ; IP address of the host
    hostgroups   allhosts              ; Host groups this host is associated with
}
```

Nachdem für den Host eine Definition hinzugefügt wurde, können wir mit der Definition von zu überwachenden Services beginnen. Genau wie Host-Definitionen können auch Service-Definitionen in jeder Objektkonfigurationdatei abgelegt werden.

erstellen von Service-Definitionen

Für jeden Service, den Sie überwachen wollen, müssen Sie in Icinga einen [Service](#) definieren, der mit der Host-Definition verbunden ist, die Sie gerade angelegt haben. Sie können Host-Definitionen in jeder Objektkonfigurationsdatei platzieren, die mit einer [cfg_file](#)-Direktive definiert ist oder in einem Verzeichnis, das in einer [cfg_dir](#)-Direktive angegeben ist.

Einige Beispiel-Service-Definitionen zur Überwachung von gebräuchlichen Services (HTTP, FTP, usw.) finden Sie nachfolgend.

HTTP überwachen

Wahrscheinlich werden Sie zu irgendeinem Zeitpunkt Web-Server überwachen wollen - entweder Ihre eigenen oder die von anderen. Das *check_http*-Plugin macht genau das. Es versteht HTTP und kann Antwortzeiten, Fehler-Codes, Zeichenketten im zurückgelieferten HTML, Server-Zertifikate und vieles mehr überwachen.

Die *commands.cfg*-Datei enthält eine Befehlsdefinition für das *check_http*-Plugin. Sie lautet:

```
define command{
    name          check_http
    command_name  check_http
    command_line   $USER1$/check_http -I $HOSTADDRESS$ $ARG1$}
```

Eine einfache Service-Definition, um den HTTP-Service auf dem *remotehost*-Rechner zu überwachen, würde so aussehen:

```
define service{
    use           generic-service      ; Inherit default values from a template
    host_name     remotehost
    service_description  HTTP
    check_command  check_http
}
```

Diese einfache Service-Definition wird den auf *remotehost* laufenden HTTP-Service überwachen. Es werden Alarne erzeugt, wenn der Web-Server nicht innerhalb von 10 Sekunden antwortet bzw. wenn HTTP-Fehler-Codes (403, 404, usw.) zurückgeliefert werden. Das ist alles, was Sie für eine einfache Überwachung brauchen. Ziemlich simpel, oder?



Hinweis: Für eine erweiterte Überwachung starten Sie das *check_http*-Plugin manuell mit *--help* als Kommandozeilenargument, um alle Optionen zu sehen, die das Plugin unterstützt. Diese *--help*-Syntax funktioniert bei allen Plugins, die wir in diesem Dokument behandeln werden.

Eine fortgeschrittenere Definition zur Überwachung des HTTP-Service finden Sie nachfolgend. Diese Service-Definition wird prüfen, ob der URI /download/index.php die Zeichenkette "latest-version.tar.gz" enthält. Falls die Zeichenkette nicht gefunden wird, der URI nicht gültig ist oder der Web-Server länger als fünf Sekunden für die Antwort braucht, wird ein Fehler erzeugt.

```
define service{
    use           generic-service      ; Inherit default values from a template
    host_name     remotehost
    service_description  Product Download Link
    check_command  check_http!-u /download/index.php -t 5 -s "latest-version.tar.gz"
}
```

FTP überwachen

Wenn Sie FTP-Server überwachen müssen, können Sie das *check_ftp*-Plugin benutzen. Die *commands.cfg*-Datei enthält eine Befehlsdefinition für das *check_ftp*-Plugin. Sie lautet:

```
define command{
    command_name  check_ftp
    command_line   $USER1$/check_ftp -H $HOSTADDRESS$ $ARG1$}
```

Eine einfache Service-Definition, um den FTP-Server auf dem *remotehost*-Rechner zu überwachen, würde so aussehen:

```
define service{
    use          generic-service      ; Inherit default values from a template
    host_name   remotehost
    service_description  FTP
    check_command   check_ftp
}
```

Diese Service-Definition wird den FTP-Service überwachen und Alarme erzeugen, wenn der FTP-Server nicht innerhalb von 10 Sekunden antwortet.

Eine fortgeschrittenere Definition finden Sie nachfolgend. Dieser Service wird den FTP-Server prüfen, der auf Port 1023 auf *remotehost* läuft. Falls der FTP-Server nicht innerhalb von fünf Sekunden antwortet oder die Server-Antwort nicht die Zeichenkette "Pure-FTPd [TLS]" enthält, wird ein Fehler erzeugt.

```
define service{
    use          generic-service      ; Inherit default values from a template
    host_name   remotehost
    service_description  Special FTP
    check_command   check_ftp!-p 1023 -t 5 -e "Pure-FTPd [TLS]"
}
```

SSH überwachen

Wenn Sie SSH-Server überwachen müssen, können Sie das *check_ssh*-Plugin benutzen. Die *commands.cfg*-Datei enthält eine Befehlsdefinition für das *check_ssh*-Plugin. Sie lautet:

```
define command{
    command_name  check_ssh
    command_line   $USER1$/check_ssh $ARG1$ $HOSTADDRESS$
```

Eine einfache Service-Definition, um den SSH-Server auf dem *remotehost*-Rechner zu überwachen, würde so aussehen:

```
define service{
    use          generic-service      ; Inherit default values from a template
    host_name   remotehost
    service_description  SSH
    check_command   check_ssh
}
```

Diese Service-Definition wird den SSH-Service überwachen und Alarme erzeugen, wenn der SSH-Server nicht innerhalb von 10 Sekunden antwortet.

Eine fortgeschrittenere Definition finden Sie nachfolgend. Dieser Service wird den SSH-Server prüfen und einen Fehler erzeugen, wenn der Server nicht innerhalb von fünf Sekunden antwortet oder die Server-Antwort nicht mit der Zeichenkette "OpenSSH_4.2" übereinstimmt.

```
define service{
    use          generic-service      ; Inherit default values from a template
    host_name   remotehost
    service_description  SSH Version Check
    check_command   check_ssh!-t 5 -r "OpenSSH_4.2"
}
```

SMTP

Das *check_smtp*-Plugin kann genutzt werden, um Ihren e-Mail-Server zu überwachen. Die *commands.cfg*-Datei enthält eine Befehlsdefinition für das *check_smtp*-Plugin. Sie lautet:

```
define command{
    command_name      check_smtp
    command_line      $USER1$/check_smtp -H $HOSTADDRESS$ $ARG1$
```

Eine einfache Service-Definition, um den SMTP-Server auf dem *remotehost*-Rechner zu überwachen, würde so aussehen:

```
define service{
    use                  generic-service          ; Inherit default values from a template
    host_name            remotehost
    service_description  SMTP
    check_command        check_smtp
```

Diese Service-Definition wird den SMTP-Service überwachen und Alarme erzeugen, wenn der SMTP-Server nicht innerhalb von 10 Sekunden antwortet.

Eine fortgeschrittenere Definition finden Sie nachfolgend. Dieser Service wird den SMTP-Server prüfen und einen Fehler erzeugen, wenn der Server nicht innerhalb von fünf Sekunden antwortet oder die Server-Antwort nicht die Zeichenkette "mygreatmailserver" enthält.

```
define service{
    use                  generic-service          ; Inherit default values from a template
    host_name            remotehost
    service_description  SMTP Response Check
    check_command        check_smtp!-t 5 -e "mygreatmailserver.com"
```

POP3 überwachen

Das *check_pop*-Plugin kann genutzt werden, um den POP3-Service Ihres e-Mail-Servers zu überwachen. Die *commands.cfg*-Datei enthält eine Befehlsdefinition für das *check_pop*-Plugin. Sie lautet:

```
define command{
    command_name      check_pop
    command_line      $USER1$/check_pop -H $HOSTADDRESS$ $ARG1$
```

Eine einfache Service-Definition, um den POP3-Service auf dem *remotehost*-Rechner zu überwachen, würde so aussehen:

```
define service{
    use                  generic-service          ; Inherit default values from a template
    host_name            remotehost
    service_description  POP3
    check_command        check_pop
```

Diese Service-Definition wird den POP3-Service überwachen und Alarme erzeugen, wenn der POP3-Server nicht innerhalb von 10 Sekunden antwortet.

Eine fortgeschrittenere Definition finden Sie nachfolgend. Dieser Service wird den POP3-Service prüfen und einen Fehler erzeugen, wenn der Server nicht innerhalb von fünf Sekunden antwortet oder die Server-Antwort nicht die Zeichenkette "mygreatmailserver.com" übereinstimmt.

```
define service{
    use                  generic-service          ; Inherit default values from a template
    host_name            remotehost
    service_description  POP3 Response Check
    check_command        check_pop!-t 5 -e "mygreatmailserver.com"
```

IMAP überwachen

Das *check_imap*-Plugin kann genutzt werden, um den IMAP4-Service Ihres e-Mail-Servers zu überwachen. Die *commands.cfg*-Datei enthält eine Befehlsdefinition für das *check_imap*-Plugin. Sie lautet:

```
define command{
    command_name      check_imap
    command_line      $USER1$/check_imap -H $HOSTADDRESS$ $ARG1$
}
```

Eine einfache Service-Definition, um den IMAP4-Server auf dem *remotehost*-Rechner zu überwachen, würde so aussehen:

```
define service{
    use                  generic-service          ; Inherit default values from a template
    host_name           remotehost
    service_description IMAP
    check_command       check_imap
}
```

Diese Service-Definition wird den IMAP4-Service überwachen und Alarme erzeugen, wenn der IMAP-Server nicht innerhalb von 10 Sekunden antwortet.

Eine fortgeschrittenere Definition finden Sie nachfolgend. Dieser Service wird den IMAP4-Service prüfen und einen Fehler erzeugen, wenn der Server nicht innerhalb von fünf Sekunden antwortet oder die Server-Antwort nicht die Zeichenkette "mygreatmailserver.com" enthält.

```
define service{
    use                  generic-service          ; Inherit default values from a template
    host_name           remotehost
    service_description IMAP4 Response Check
    check_command       check_imap!-t 5 -e "mygreatmailserver.com"
}
```

Icinga erneut starten

Sobald Sie die neuen Host- und Service-Definitionen zu Ihrer/n Konfigurationsdatei(en) hinzugefügt haben, sind Sie bereit, sie zu überwachen. Um dies zu tun, müssen Sie [die Konfiguration überprüfen](#) und [Icinga erneut starten](#).

Wenn der Überprüfungsprozess irgendwelche Fehler produziert, dann verbessern Sie Ihre Konfigurationsdatei, bevor Sie fortfahren. Stellen Sie sicher, dass Sie Icinga nicht erneut starten, bevor der Überprüfungsprozess ohne Fehler durchläuft!

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Router und Switches überwachen](#)

[Zum Anfang](#)

[Kapitel 3. Icinga konfigurieren](#)



Kapitel 3. Icinga konfigurieren

[Zurück](#)[Weiter](#)

Kapitel 3. Icinga konfigurieren

Inhaltsverzeichnis

[Konfigurationsüberblick](#)[Optionen der Hauptkonfigurationsdatei](#)[Überblick Objektkonfiguration](#)[Objektdefinitionen](#)[Host-Definition](#)[Hostgruppen-Definition](#)[Service-Definition](#)[Servicegruppen-Definition](#)[Kontakt-Definition](#)[Kontaktgruppen-Definition](#)[Zeitfenster-Definition \(timeperiod\)](#)[Befehls-Definition \(command\)](#)[Service-Abhängigkeits-Definition \(servicedependency\)](#)[Serviceescalations-Definition](#)[Host-Abhängigkeits-Definition \(hostdependency\)](#)[Host-Escalations-Definition](#)[erweiterte Hostinformations-Definition \(hostextinfo\)](#)[erweiterte Serviceinformations-Definition \(serviceextinfo\)](#)[Module-Definition](#)[Maßgeschneiderte Objektvariablen](#)[Optionen CGI-Konfigurationsdatei](#)[Authentifizierung und Autorisierung in den CGIs](#)[Zurück](#)[Weiter](#)[Öffentlich zugängliche Dienste
überwachen](#)[Zum Anfang](#)[Konfigurationsüberblick](#)



Konfigurationsüberblick

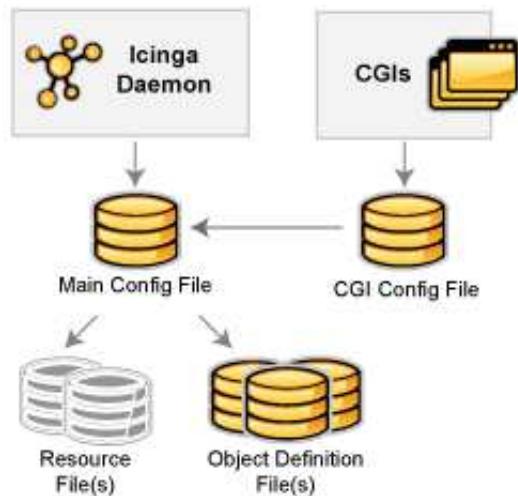
[Zurück](#)
[Kapitel 3. Icinga konfigurieren](#)
[Weiter](#)

Konfigurationsüberblick

Einführung

Es gibt verschiedene Konfigurationsdateien, die Sie erstellen oder editieren müssen, bevor Sie irgendetwas überwachen können. Haben Sie Geduld! Icinga zu konfigurieren kann eine Weile dauern, besonders wenn Sie ein Neuling sind. Sobald Sie herausgefunden haben, wie die Dinge funktionieren, werden Sie feststellen, dass es die Mühe wert ist. :-)

 Anmerkung: Beispiel-Konfigurationsdateien werden im `/usr/local/icinga/etc/`-Verzeichnis installiert, wenn Sie der [Schnellstart-Installationsanleitung](#) folgen.



Hauptkonfigurationsdatei

Die Hauptkonfigurationsdatei enthält eine Reihe von Direktiven, die die Arbeitsweise des Icinga-Daemon beeinflussen. Diese Konfigurationsdatei wird vom Icinga-Daemon und den CGIs gelesen. Hier werden Sie in Ihr Konfigurationsabenteuer starten wollen.

Dokumentation zur Hauptkonfigurationsdatei finden Sie [hier](#).

Ressource-Datei(en)

Ressource-Dateien können zur Speicherung von benutzerdefinierten Makros genutzt werden. Der Hauptgrund für Ressource-Dateien liegt darin, dass sie genutzt werden können, um sensible Informationen (wie z.B. Passworte) zu speichern, ohne dass sie für CGIs zugänglich

sind, weil diese Dateien nicht von den CGIs gelesen werden.

Sie können eine oder mehrere optionale Ressource-Dateien mit Hilfe der [resource_file](#)-Direktive in Ihrer Hauptkonfigurationsdatei angeben.

Objektdefinitionen-Dateien

Objektdefinitionen-Dateien werden genutzt, um Hosts, Services, Hostgruppen, Kontakte, Kontaktgruppen, Befehle usw. zu definieren. Hier definieren Sie, welche Dinge Sie überwachen wollen und wie Sie diese überwachen wollen.

Sie können eine oder mehrere Objektdefinitionen-Dateien mit Hilfe der [cfg_file](#)- und/oder [cfg_dir](#)-Direktiven in Ihrer Hauptkonfigurationsdatei angeben.

Eine Einführung zu Objektdefinitionen und wie sie in Beziehung zu einander stehen, finden Sie [hier](#).

Ihre Objektkonfigurationsdateien können wiederum andere Dateien einschließen mit Hilfe der [include_file](#)- oder [include_dir](#)-Direktiven. Sie können lediglich außerhalb der eigentlichen Objektdefinitionen auftreten und verhalten sich analog zu den [cfg_file](#)- und [cfg_dir](#)-Direktiven in der Hauptkonfigurationsdatei.

CGI-Konfigurationsdatei

Die CGI-Konfigurationsdatei enthält eine Reihe von Direktiven, die die Arbeitsweise der [CGIs](#) beeinflussen. Sie enthält auch einen Verweis auf die Hauptkonfigurationsdatei, so dass die CGIs wissen, wie Sie Icinga konfiguriert haben und wo Ihre Objektdefinitionen gespeichert sind.

Dokumentation zur CGI-Konfigurationsdatei finden Sie [hier](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Kapitel 3. Icinga konfigurieren

[Zum Anfang](#)

Optionen der
Hauptkonfigurationsdatei

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Optionen der Hauptkonfigurationsdatei

[Zurück](#)
[Kapitel 3. Icinga konfigurieren](#)
[Weiter](#)

Optionen der Hauptkonfigurationsdatei

Anmerkungen

Bei der Erstellung und/oder Änderung von Konfigurationsdateien sollten Sie folgendes beachten:

1. Zeilen, die mit einem '#'-Zeichen beginnen, werden als Kommentar angesehen und nicht verarbeitet
2. Variablennamen müssen am Anfang der Zeile beginnen - "White space" vor dem Namen sind nicht erlaubt
3. Variablennamen sind abhängig von Groß- und Kleinschreibung (case-sensitive)

Beispiel-Konfigurationsdatei



Hinweis: eine Beispiel-Hauptkonfigurationsdatei (`/usr/local/icinga/etc/icinga.cfg`) wird installiert, wenn Sie der [Schnellstartanleitung](#) folgen.

Position der Konfigurationsdatei

Die Hauptkonfigurationsdatei heißt üblicherweise `icinga.cfg` und ist im `/usr/local/icinga/etc/-Verzeichnis` zu finden.

Variablen der Konfigurationsdatei

Nachfolgend finden Sie Beschreibungen jeder Option der Icinga-Hauptkonfigurationsdatei...

Protokolldatei (Log File)

Format: `log_file=<file_name>`

Beispiel: `log_file=/usr/local/icinga/var/icinga.log`

Diese Variable gibt an, wo Icinga die Hauptprotokolldatei anlegen soll. Dies sollte die erste Variable sein, die Sie in Ihrer Konfigurationsdatei definieren, weil Icinga versucht, dorthin die Fehler zu schreiben, die es in den übrigen Konfigurationsdaten findet. Wenn Sie [Log-Rotation](#) aktiviert haben, dann wird diese Datei automatisch jede Stunde, jeden Tag, jede Woche oder jeden Monat rotiert.

Objektkonfigurationsdatei (Object Configuration File)

Format: `cfg_file=<file_name>`
`cfg_file=/usr/local/icinga/etc/hosts.cfg`

Beispiel: `cfg_file=/usr/local/icinga/etc/services.cfg`
`cfg_file=/usr/local/icinga/etc/commands.cfg`

Diese Direktive wird benutzt, um eine [Objektkonfigurationsdatei](#) anzugeben, die Objektdefinitionen enthält, die Icinga zur Überwachung nutzen soll. Objektkonfigurationsdateien enthalten Definitionen für Hosts, Hostgruppen, Kontakte, Kontaktgruppe, Services, Befehle usw. Sie können Ihre Konfigurationsinformationen in verschiedene Dateien aufteilen und mehrere `cfg_file=`-Einträge angeben, um jede einzelne zu verarbeiten.

Objektkonfigurationsverzeichnis (Object Configuration Directory)

Format: `cfg_dir=<directory_name>`
`cfg_dir=/usr/local/icinga/etc/commands`

Beispiel: `cfg_dir=/usr/local/icinga/etc/services`
`cfg_dir=/usr/local/icinga/etc/hosts`

Diese Direktive wird benutzt, um ein Verzeichnis anzugeben, das [Objektkonfigurationsdateien](#) enthält, die Icinga zur Überwachung nutzen soll. Alle Dateien in dem Verzeichnis mit einer `.cfg`-Endung werden als Objektkonfigurationsdateien verarbeitet. Zusätzlich wird Icinga rekursiv alle Konfigurationsdateien in den Unterverzeichnissen des Verzeichnisses verarbeiten, das Sie angegeben haben. Sie können Ihre Konfigurationsinformationen in verschiedene Verzeichnisse aufteilen und mehrere `cfg_dir=`-Einträge angeben, um alle Konfigurationsdateien jedes einzelnen Verzeichnisses zu verarbeiten.

Objekt-Cache-Datei (Object Cache File)

Format: `object_cache_file=<file_name>`
Beispiel: `object_cache_file=/usr/local/icinga/var/objects.cache`

Diese Direktive wird benutzt, um eine Datei anzugeben, in der eine zwischengespeicherte (cached) Kopie der [Objektdefinitionen](#) abgelegt wird. Die Cache-Datei wird jedes Mal (erneut) angelegt, wenn Icinga (erneut) gestartet wird, und wird von den CGIs benutzt. Sie ist dazu gedacht, die Zwischenspeicherung der Konfigurationsdateien in den CGIs zu beschleunigen und es Ihnen zu erlauben, die [Objektkonfigurationsdateien](#) zu editieren, während Icinga läuft, ohne die Ausgaben der CGIs zu beeinflussen.

vorgespeicherte Objektdatei (Precached Object File)

Format: `precached_object_file=<file_name>`
Beispiel: `precached_object_file=/usr/local/icinga/var/objects.precache`

Diese Direktive wird benutzt, um eine Datei anzugeben, die eine vorverarbeitete (pre-processed), vorgespeicherte (pre-cached) Kopie von [Objektdefinitionen](#) enthält. Diese Datei kann genutzt werden, um drastisch den Startvorgang in großen/komplexen Icinga-Installationen zu beschleunigen. Lesen Sie [hier](#), wie der Startvorgang beschleunigt werden kann.

Ressource-Datei (Resource File)

Format: **resource_file=<file_name>**

Beispiel: **resource_file=/usr/local/icinga/etc/resource.cfg**

Dies wird benutzt, um eine optionale Ressource-Datei anzugeben, die \$USERn\$-[Makro](#)-Definitionen enthalten kann. \$USER\$-Makros sind sinnvoll zur Speicherung von Benutzernamen, Passwörtern und Objekten, die häufig in Befehlsdefinitionen (wie z.B. Verzeichnispfade) benutzt werden. Die CGIs werden *nicht* versuchen, Ressource-Dateien zu lesen, so dass Sie die Berechtigungen beschränken können (600 oder 660), um sensible Informationen zu schützen. Sie können mehrere Ressource-Dateien angeben, indem Sie mehrere resource_file-Einträge in die Hauptkonfigurationsdatei aufnehmen. Icinga wird sie alle verarbeiten. Schauen Sie in die resource.cfg-Datei im *sample-config*-Unterverzeichnis der Icinga-Distribution, um ein Beispiel für die Definition von \$USER\$-Makros zu sehen.

temporäre Datei (Temp File)

Format: **temp_file=<file_name>**

Beispiel: **temp_file=/usr/local/icinga/var/nagios.tmp**

Dies ist eine temporäre Datei, die Icinga periodisch anlegt, wenn Kommentardaten, Statusdaten usw. aktualisiert werden. Die Datei wird gelöscht, wenn sie nicht länger benötigt wird.

temporärer Pfad (Temp Path)

Format: **temp_path=<dir_name>**

Beispiel: **temp_path=/tmp**

Dies ist ein Verzeichnis, das Icinga als "Schmierblock" (scratch space) benutzen kann, um während des Überwachungsprozesses temporäre Dateien anlegen zu können. Sie sollten *tmpwatch* oder ein ähnliches Programm ausführen, um in diesem Verzeichnis Dateien zu löschen, die älter als 24 Stunden sind.

Status-Datei (Status File)

Format: **status_file=<file_name>**

Beispiel: **status_file=/usr/local/icinga/var/status.dat**

Dies ist die Datei, die Icinga nutzt, um den aktuellen Zustand, Kommentar- und Ausfallzeitinformationen zu speichern. Diese Datei wird von den CGIs genutzt, so dass der aktuelle Überwachungszustand über ein Web-Interface berichtet werden kann. Die CGIs müssen Lesezugriff auf diese Datei haben, um richtig funktionieren zu können. Diese Datei wird jedes Mal gelöscht, wenn Icinga endet und neu angelegt, wenn Icinga startet.

Statusdatei-Aktualisierungsintervall (Status File Update Interval)

Format: **status_update_interval=<seconds>**

Beispiel: **status_update_interval=15**

Diese Einstellung legt fest, wie oft (in Sekunden) Icinga Statusdaten in der [Statusdatei](#) aktualisiert. Das kleinste Aktualisierungsintervall ist eine Sekunde.

Icinga-Benutzer (Icinga User)

Format: **icinga_user=<username/UID>**

Beispiel: **icinga_user=nagios**

Dies wird benutzt, um den "eigentlichen" (effective) Benutzer zu setzen, mit dem der Icinga-Prozess laufen soll. Nach dem anfänglichen Programmstart und bevor irgendeine Überwachung beginnt, wird Icinga die vorhandenen Berechtigungen "fallen lassen" (drop) und als dieser Benutzer laufen. Sie können entweder einen Benutzernamen oder eine UID angeben.

Icinga-Gruppe (Icinga Group)

Format: **icinga_group=<groupname/GID>**

Beispiel: **icinga_group=nagios**

Dies wird benutzt, um die "eigentliche" (effective) Gruppe zu setzen, mit der der Icinga-Prozess laufen soll. Nach dem anfänglichen Programmstart und bevor irgendeine Überwachung beginnt, wird Icinga die vorhandenen Berechtigungen "fallen lassen" (drop) und als diese Gruppe laufen. Sie können entweder einen Gruppennamen oder eine GID angeben.

Benachrichtigungsoption (Notifications Option)

Format: **enable_notifications=<0/1>**

Beispiel: **enable_notifications=1**

Diese Option legt fest, ob Icinga [Benachrichtigungen](#) versendet. Wenn diese Optionen deaktiviert ist, wird Icinga nach dem (Neu-) Start keine Benachrichtigungen für irgendeinen Host oder Service versenden. Anmerkung: Wenn Sie [Statusinformationsaufbewahrung](#) (retain state information) aktiviert haben, wird Icinga diese Einstellung ignorieren, wenn es (erneut) startet und die letzte bekannte Einstellung dieser Option nutzen (wie sie in der [Statusaufbewahrungsdatei](#) abgelegt ist), *es sei denn*, Sie haben die [use_retained_program_state](#)-Option deaktiviert. Wenn Sie diese Option ändern möchten, während die Statusaufbewahrung aktiviert ist (und die Option [use_retained_program_state](#) aktiviert ist), müssen Sie den entsprechenden [externen Befehl](#) benutzen oder sie über das Web-Interface ändern. Die Werte sind wie folgt:

- 0 = Benachrichtigungen deaktivieren
- 1 = Benachrichtigungen aktivieren (Default)

Option Service-Prüfungen ausführen (Service Check Execution Option)

Format: **execute_service_checks=<0/1>**

Beispiel: **execute_service_checks=1**

Diese Option legt fest, ob Icinga nach dem (Neu-) Start Service-Prüfungen ausführt. Wenn diese Option deaktiviert ist, wird Icinga nicht aktiv irgendwelche Service-Prüfungen ausführen und in einer Art von "Schlafmodus" verbleiben (es kann weiterhin [passive Prüfungen](#) empfangen, es sei denn, Sie haben [diese deaktiviert](#)). Diese Option wird oft benutzt, wenn Ersatz-Überwachungs-Server (backup monitoring server) konfiguriert werden, wie dies in der Dokumentation zu [Redundanz](#) beschrieben ist, oder wenn Sie eine [verteilte](#)-Überwachungsumgebung aufbauen. Anmerkung: wenn Sie [Statusinformationsaufbewahrung](#) (retain state information) aktiviert haben, wird Icinga diese Einstellung ignorieren, wenn es (erneut) startet und die letzte bekannte Einstellung dieser Option nutzen (wie sie in der [Statusaufbewahrungsdatei](#) abgelegt ist), *es sei denn*, Sie haben die [use_retained_program_state](#)-Option deaktiviert. Wenn Sie diese Option ändern möchten, während die Statusaufbewahrung aktiviert ist (und die Option [use_retained_program_state](#) aktiviert ist), müssen Sie den entsprechenden [externen Befehl](#) benutzen oder sie über das Web-Interface ändern. Die Werte sind wie folgt:

- 0 = keine Service-Prüfungen ausführen
- 1 = Service-Prüfungen ausführen (Default)

Option passive Service-Prüfungen akzeptieren (Passive Service Check Acceptance Option)

Format: **accept_passive_service_checks=<0/1>**

Beispiel: **accept_passive_service_checks=1**

Diese Option legt fest, ob Icinga nach dem (Neu-) Start [passive Service-Prüfungen](#) akzeptiert. Wenn diese Option deaktiviert ist, wird Icinga keine passiven Service-Prüfungen akzeptieren. Anmerkung: wenn Sie [Statusinformationsaufbewahrung](#) (retain state information) aktiviert haben, wird Icinga diese Einstellung ignorieren, wenn es (erneut) startet und die letzte bekannte Einstellung dieser Option nutzen (wie sie in der [Statusaufbewahrungsdatei](#) abgelegt ist), *es sei denn*, Sie haben die [use_retained_program_state](#)-Option deaktiviert. Wenn Sie diese Option ändern möchten, während die Statusaufbewahrung aktiviert ist (und die Option [use_retained_program_state](#) aktiviert ist), müssen Sie den entsprechenden [externen Befehl](#) benutzen oder sie über das Web-Interface ändern. Die Werte sind wie folgt:

- 0 = keine passiven Service-Prüfungen akzeptieren
- 1 = passive Service-Prüfungen akzeptieren (Default)

Option Host-Prüfungen ausführen (Host Check Execution Option)

Format: **execute_host_checks=<0/1>**

Beispiel: **execute_host_checks=1**

Diese Option legt fest, ob Icinga nach dem (Neu-) Start nach Bedarf oder regelmäßig geplante Host-Prüfungen ausführt. Wenn diese Option deaktiviert ist, wird Icinga nicht aktiv irgendwelche Host-Prüfungen ausführen, obwohl es weiterhin [passive Host-Prüfungen](#) empfangen wird, es sei denn, Sie haben [diese deaktiviert](#). Diese Option wird am meisten genutzt, wenn Ersatz-Überwachungs-Server (backup monitoring server) konfiguriert werden, wie dies in der Dokumentation zu [Redundanz](#) beschrieben ist, oder wenn Sie eine

verteilte-Überwachungsumgebung aufbauen. Anmerkung: wenn Sie [Statusinformationsaufbewahrung](#) (retain state information) aktiviert haben, wird Icinga diese Einstellung ignorieren, wenn es (erneut) startet und die letzte bekannte Einstellung dieser Option nutzen (wie sie in der [Statusaufbewahrungsdatei](#) abgelegt ist), *es sei denn*, Sie haben die [use_retained_program_state](#)-Option deaktiviert. Wenn Sie diese Option ändern möchten, während die Statusaufbewahrung aktiviert ist (und die Option [use_retained_program_state](#) aktiviert ist), müssen Sie den entsprechenden [externen Befehl](#) benutzen oder sie über das Web-Interface ändern. Die Werte sind wie folgt:

- 0 = keine Host-Prüfungen ausführen
- 1 = Host-Prüfungen ausführen (Default)

Option passive Host-Prüfungen akzeptieren (Passive Host Check Acceptance Option)

Format: **accept_passive_host_checks=<0/1>**

Beispiel: **accept_passive_host_checks=1**

Diese Option legt fest, ob Icinga nach dem (Neu-) Start [passive Host-Prüfungen](#) akzeptiert. Wenn diese Option deaktiviert ist, wird Icinga keine passiven Host-Prüfungen akzeptieren. Anmerkung: wenn Sie [Statusinformationsaufbewahrung](#) (retain state information) aktiviert haben, wird Icinga diese Einstellung ignorieren, wenn es (erneut) startet und die letzte bekannte Einstellung dieser Option nutzen (wie sie in der [Statusaufbewahrungsdatei](#) abgelegt ist), *es sei denn*, Sie haben die [use_retained_program_state](#)-Option deaktiviert. Wenn Sie diese Option ändern möchten, während die Statusaufbewahrung aktiviert ist (und die Option [use_retained_program_state](#) aktiviert ist), müssen Sie den entsprechenden [externen Befehl](#) benutzen oder sie über das Web-Interface ändern. Die Werte sind wie folgt:

- 0 = keine passiven Host-Prüfungen akzeptieren
- 1 = passive Host-Prüfungen akzeptieren (Default)

Eventhandler-Option (Event Handler Option)

Format: **enable_event_handlers=<0/1>**

Beispiel: **enable_event_handlers=1**

Diese Option legt fest, ob Icinga nach dem (Neu-) Start [Eventhandler](#) ausführt. Wenn diese Option deaktiviert ist, wird Icinga keine Host- oder Service-Eventhandler ausführen. Anmerkung: wenn Sie [Statusinformationsaufbewahrung](#) (retain state information) aktiviert haben, wird Icinga diese Einstellung ignorieren, wenn es (erneut) startet und die letzte bekannte Einstellung dieser Option nutzen (wie sie in der [Statusaufbewahrungsdatei](#) abgelegt ist), *es sei denn*, Sie haben die [use_retained_program_state](#)-Option deaktiviert. Wenn Sie diese Option ändern möchten, während die Statusaufbewahrung aktiviert ist (und die Option [use_retained_program_state](#) aktiviert ist), müssen Sie den entsprechenden [externen Befehl](#) benutzen oder sie über das Web-Interface ändern. Die Werte sind wie folgt:

- 0 = Eventhandler deaktivieren
- 1 = Eventhandler aktivieren (Default)

Protokollrotationsmethode (Log Rotation Method)

Format: **log_rotation_method=<n/h/d/w/m>**

Beispiel: **log_rotation_method=d**

Dies ist die Rotationsmethode, die Icinga für Ihre Protokolldatei nutzen soll. Die Werte sind wie folgt:

- n = keine ("none" - die Protokolldatei nicht rotieren, das ist der Standard)
- h = stündlich ("hourly" - die Protokolldatei jede volle Stunde rotieren)
- d = täglich ("daily" - die Protokolldatei jeden Tag um Mitternacht rotieren)
- w = wöchentlich ("weekly" - die Protokolldatei jeden Samstag um Mitternacht rotieren)
- m = monatlich ("monthly" - die Protokolldatei am letzten Tag des Monats um Mitternacht rotieren)

Protokollarchiv-Pfad (Log Archiv Path)

Format: **log_archive_path=<path>**

Beispiel: **log_archive_path=/usr/local/icinga/var/archives/**

Dies ist das Verzeichnis, in dem Icinga die Protokolldateien ablegen soll, die rotiert wurden. Diese Option wird ignoriert, wenn Sie die Funktionalität der [Protokollrotation](#) (log rotation) nicht nutzen.

Option externe Befehle prüfen (External Command Check Option)

Format: **check_external_commands=<0/1>**

Beispiel: **check_external_commands=1**

Diese Option legt fest, ob Icinga das [command file](#) auf auszuführende Befehle prüfen soll. Diese Option muss aktiviert sein, wenn Sie planen, das [Command-CGI](#) zu nutzen, um Befehle über das Web-Interface zu erteilen. Mehr Informationen zu externen Befehlen finden Sie [hier](#).

- 0 = nicht auf externe Befehle prüfen
- 1 = auf externe Befehle prüfen (Default)

Prüfintervall externe Befehle (External Command Check Interval)

Format: **command_check_interval=<xxx>[s]**

Beispiel: **command_check_interval=1**

Wenn Sie eine Zahl mit einem angehängten "s" angeben (z.B. 30s), dann ist dies die Zahl in *Sekunden*, die zwischen Prüfungen auf externe Befehle gewartet werden soll. Wenn Sie das "s" weglassen, ist dies die Zahl von "Zeiteinheiten", die zwischen den Prüfungen auf externe Befehle gewartet werden soll. Solange Sie nicht den Standardwert (60) der [interval_length](#)-Direktive geändert haben (wie weiter unten definiert), bedeutet dieser Wert Minuten.

Anmerkung: durch das Setzen dieses Wertes auf **-1** wird Icinga so oft wie möglich auf externe Befehle prüfen. Jedes Mal, wenn Icinga auf externe Befehle prüft, wird es alle im [command file](#) befindlichen Befehle lesen und verarbeiten, bevor es mit anderen Aufgaben fortfährt. Mehr Informationen zu externen Befehlen finden Sie [hier](#).

externe Befehlsdatei (External Command File)

Format: **command_file=<file_name>**

Beispiel: **command_file=/usr/local/icinga/var/rw/nagios.cmd**

Dies ist die Datei, die Icinga auf zu verarbeitende externe Befehle prüfen wird. Das [Command-CGI](#) schreibt Befehle in diese Datei. Die externe Befehlsdatei ist als "named pipe" (FIFO) implementiert, die beim Start von Icinga angelegt und beim Herunterfahren wieder gelöscht wird. Wenn die Datei beim Start von Icinga existiert, wird der Icinga-Prozess mit einer Fehlermeldung enden. Mehr Informationen zu externen Befehlen finden Sie [hier](#).

externe Befehlpuffer-Slots (External Command Buffer Slots)

Format: **external_command_buffer_slots=<n>**

Beispiel: **external_command_buffer_slots=512**

Anmerkung: dies ist ein fortgeschrittenes Feature. Diese Option legt fest, wie viele Puffer-Slots Icinga für die Zwischenspeicherung von externen Befehlen reserviert, die von einem "worker thread" aus der externen Befehlsdatei gelesen, aber noch nicht vom "main thread" des Icinga-Daemons verarbeitet wurden. Jeder Slot kann einen externen Befehl enthalten, so dass diese Option im Wesentlichen bestimmt, wie viele Befehle gepuffert werden können. Bei Installationen, wo Sie eine große Zahl von passiven Prüfungen verarbeiten (z.B. [verteilten Setups](#)), müssen Sie ggf. diese Zahl erhöhen. Sie sollten den Einsatz von PNP4Nagios erwägen, um die Nutzung der externen Befehlpuffer grafisch darzustellen. Mehr zur Konfiguration der grafischen Darstellung finden Sie [hier](#).

Update-Prüfungen (Update Checks)

Format: **check_for_updates=<0/1>**

Beispiel: **check_for_updates=1**

Diese Option legt bei Nagios fest, ob automatisch geprüft werden soll, ob neue Updates-Versionen verfügbar sind. Es wird bei Nagios empfohlen, dass Sie diese Option aktivieren.

Bei Icinga hat diese Option keine Funktion, ab Version 1.2 wird eine entsprechende Warnung ausgegeben.

Nur Update-Prüfungen (Bare Update Checks)

Format: **bare_update_checks=<0/1>**

Beispiel: **bare_update_checks=0**

Diese Option legt bei Nagios fest, welche Daten an api.nagios.org gesandt werden. Es wird bei Nagios empfohlen, dass Sie diese Option aktivieren.

Bei Icinga hat diese Option keine Funktion, ab Version 1.2 wird eine entsprechende Warnung ausgegeben.

Sperrdatei (Lock File)

Format: **lock_file=<file_name>**

Beispiel: **lock_file=/tmp/icinga.lock**

Diese Option gibt die Position der Sperrdatei an, die Icinga anlegen sollte, wenn es als Daemon läuft (wenn es mit der -d Kommandozeilenoption gestartet wurde). Diese Datei enthält die Prozess-ID (PID) des laufenden Icinga-Prozesses.

Statusaufbewahrungsoption (State Retention Option)

Format: **retain_state_information=<0/1>**

Beispiel: **retain_state_information=1**

Diese Option legt fest, ob Icinga Statusinformationen für Hosts und Services zwischen Programmneustarts aufbewahren soll. Wenn Sie diese Option aktivieren, sollten Sie ein Wert für die [state_retention_file](#)-Variable angeben. Wenn sie aktiviert ist, wird Icinga alle Statusinformationen für Hosts und Services sichern, bevor es beendet (oder neu gestartet) wird und vorher gespeicherte Statusinformationen einlesen, wenn es neu gestartet wird.

- 0 = Statusinformationen nicht aufbewahren
- 1 = Statusinformationen aufbewahren (Default)

Statusaufbewahrungsdatei (State Retention File)

Format: **state_retention_file=<file_name>**

Beispiel: **state_retention_file=/usr/local/icinga/var/retention.dat**

Dies ist die Datei, die Icinga für die Speicherung von Status-, Ausfallzeit- und Kommentarinformationen nutzt, bevor es endet. Wenn Icinga neu startet, wird es die in dieser Datei gespeicherten Informationen nutzen, um die anfänglichen Zustände von Services und Hosts zu setzen, bevor es mit der Überwachung beginnt. Damit Icinga Statusinformationen zwischen Programmneustarts aufbewahrt, müssen Sie die [retain_state_information](#)-Option aktivieren.

Sync-Aufbewahrungsdatei (Sync Retention File)

Format: **sync_retention_file=<file_name>**

Beispiel: **sync_retention_file=/usr/local/icinga/var/retention.dat**

Dies ist eine fortgeschrittene Option, die wie state_retention_file arbeitet, so dass Sie eine Untermenge von Aufbewahrungsinformationen wie Status, Bestätigung, Ausfallzeiten und Kommentaren laden können (Sie müssen den Inhalt dieser Datei außerhalb von Icinga erstellen). Wenn Icinga erneut gestartet wird, liest es die Informationen aus der durch

`sync_retention_file` definierten Datei und aktualisiert den angegebenen Host oder Service, wenn die `Last_update`-Zeit in der Sync-Datei neuer ist als in der `state_retention_file`-Datei, anderenfalls wird die Information ignoriert. Nach dem Lesen der Datei wird diese gelöscht. Um die Option zu deaktivieren, kommentieren Sie sie aus. Diese Option gibt es seit Icinga 1.0.2.

automatisches Statusaufbewahrungs-Aktualisierungsintervall (Automatic State Retention Update Interval)

Format: **retention_update_interval=<minutes>**

Beispiel: **retention_update_interval=60**

Diese Einstellung legt fest, wie oft (in Minuten) Icinga automatisch während des normalen Betriebs die Aufbewahrungsdaten aktualisiert. Wenn Sie einen Wert von Null angeben, wird Icinga nicht in regelmäßigen Intervallen die Aufbewahrungsdaten sichern, aber es wird die Aufbewahrungsdaten vor der Beendigung oder dem Neustart sichern. Wenn Sie die Statusaufbewahrung deaktiviert haben (mit der `retain_state_information`-Option), hat diese Option keine Auswirkung.

Option aufbewahrten Programmzustand nutzen (Use Retained Program State Option)

Format: **use_retained_program_state=<0/1>**

Beispiel: **use_retained_program_state=1**

Diese Einstellung legt fest, ob Icinga verschiedene programmweite Statusvariablen auf Basis der Aufbewahrungsdatei setzen soll. Einige dieser programmweiten Statusvariablen, die normalerweise über Programmstarts hinweg gesichert werden, wenn Statusaufbewahrung aktiviert ist, umfassen die `enable_notifications`-, `enable_flap_detection`-, `enable_event_handlers`-, `execute_service_checks`- und `accept_passive_service_checks`-Optionen. Wenn Sie **Statusaufbewahrung** deaktiviert haben, hat diese Option keine Auswirkung.

- 0 = keinen aufbewahrten Programmzustand nutzen
- 1 = aufbewahrten Programmzustand nutzen (Default)

Option aufbewahrte Planungsinformationen nutzen (Use Retained Scheduling Info Option)

Format: **use_retained_scheduling_info=<0/1>**

Beispiel: **use_retained_scheduling_info=1**

Diese Einstellung legt fest, ob Icinga Planungsinformationen für Hosts und Services aufbewahrt, wenn es neu startet. Wenn Sie eine große Zahl (oder einen großen Anteil) von Hosts oder Services hinzufügen, empfehlen wir diese Option zu deaktivieren, wenn Sie das erste Mal Icinga neu starten, weil es nachteilig die Verteilung von initialen Prüfungen beeinflussen kann. Andernfalls werden Sie diese Option wahrscheinlich aktiviert lassen.

- 0 = keine aufbewahrten Planungsinformationen nutzen
- 1 = aufbewahrten Planungsinformationen nutzen (Default)

aufbewahrte Host- und Service-Attributmasken (Retained Host and Service Attribute Masks)

Format:

```
retained_host_attribute_mask=<number>
retained_service_attribute_mask=<number>
retained_host_attribute_mask=0
```

Beispiel:

```
retained_service_attribute_mask=0
```

WARNUNG: dies ist ein fortgeschrittenes Feature. Sie müssen den Icinga-Quellcode lesen, um diese Option effizient nutzen zu können.

Diese Option legt fest, welche Host- oder Service-Attribute NICHT über Programmneustarts hinweg aufbewahrt werden. Die Werte für diese Optionen sind ein bitweises AND der durch die "MODATTR_"-Definitionen angegebenen Werte in den include/common.h-Quellcode-Dateien. Per Default werden alle Host- und Service-Attribute aufbewahrt.

aufbewahrte Prozessattributmasken (Retained Process Attribute Masks)

Format:

```
retained_process_host_attribute_mask=<number>
retained_process_service_attribute_mask=<number>
retained_process_host_attribute_mask=0
```

Beispiel:

```
retained_process_service_attribute_mask=0
```

WARNUNG: dies ist ein fortgeschrittenes Feature. Sie müssen den Icinga-Quellcode lesen, um diese Option effizient nutzen zu können.

Diese Option legt fest, welche Prozessattribute NICHT über Programmneustarts hinweg aufbewahrt werden. Es gibt zwei Masken, weil es oft separate Host- und Service-Prozessattribute gibt, die geändert werden können. Beispielsweise können Host-Prüfungen auf Programmebene deaktiviert werden, während Service-Prüfungen weiterhin aktiviert sind. Die Werte für diese Optionen sind ein bitweises AND der durch die "MODATTR_"-Definitionen angegebenen Werte in den include/common.h-Quellcode-Dateien. Per Default werden alle Prozessattribute aufbewahrt.

aufbewahrte Kontaktattributmasken (Retained Contact Attribute Masks)

Format:

```
retained_contact_host_attribute_mask=<number>
retained_contact_service_attribute_mask=<number>
retained_contact_host_attribute_mask=0
```

Beispiel:

```
retained_contact_service_attribute_mask=0
```

WARNUNG: dies ist ein fortgeschrittenes Feature. Sie müssen den Icinga-Quellcode lesen, um diese Option effizient nutzen zu können.

Diese Option legt fest, welche Kontaktattribute NICHT über Programmneustarts hinweg aufbewahrt werden. Es gibt zwei Masken, weil es oft separate Host- und Service-Prozessattribute gibt, die geändert werden können. Die Werte für diese Optionen sind ein bitweises AND der durch die "MODATTR_"-Definitionen angegebenen Werte in den include/common.h-Quellcode-Dateien. Per Default werden alle Kontaktattribute aufbewahrt.

Syslog-Protokollierungsoption (Syslog Logging Option)Format: **use_syslog=<0/1>**Beispiel: **use_syslog=1**

Diese Variable legt fest, ob Meldungen im Syslog des lokalen Hosts protokolliert werden sollen. Die Werte sind wie folgt:

- 0 = Syslog nicht nutzen
- 1 = Syslog nutzen [Default]

Syslog Local Facility OptionFormat: **use_local_syslog_facility=<0/1>**Beispiel: **use_syslog_local_facility=1**

Wenn Sie use_syslog aktiviert haben, dann können Sie Icinga anweisen, eine local-Facility zu benutzen statt des Defaults. Werte sind wie folgt:

- 0 = Syslog Local Facility nicht nutzen
- 1 = Syslog Local Facility benutzen

Diese Option gibt es seit Icinga 1.0.2.

Syslog Local Facility WertFormat: **syslog_local_facility=<0|1|2|3|4|5|6|7>**Beispiel: **syslog_local_facility=1**

Wenn Sie use_syslog_local_facility aktiviert haben, können Sie auswählen, welche Local-Facility benutzt werden soll. Gültige Werte sind von 0 bis 7. Diese Option gibt es seit Icinga 1.0.2.

Benachrichtigungsprotokollierungsoption (Notification Logging Option)Format: **log_notifications=<0/1>**Beispiel: **log_notifications=1**

Diese Variable legt fest, ob Benachrichtigungsmeldungen protokolliert werden. Wenn Sie eine Menge von Kontakten oder ständigen Service-Ausfällen haben, dann wird Ihre Protokolldatei relativ schnell wachsen. Benutzen Sie diese Option, um die Protokollierung von (Kontakt-)Benachrichtigungen zu verhindern.

- 0 = keine Benachrichtigungen protokollieren
- 1 = Benachrichtigungen protokollieren [Default]

Option Service-Wiederholungsprüfungen protokollieren (Service Check Retry Logging Option)

Format: **log_service_retries=<0/1>**

Beispiel: **log_service_retries=1**

Diese Variable legt fest, ob Service-Wiederholungsprüfungen protokolliert werden. Service-Wiederholungsprüfungen treten auf, wenn ein Service-Prüfergebnis einen nicht-OK-Status ergibt, Sie Icinga aber so konfiguriert haben, dass die Prüfung mehr als einmal wiederholt wird, bevor ein Fehler gemeldet wird. Services in diesem Zustand befinden sich in einem "Soft"-Status. Die Protokollierung von Service-Wiederholungsprüfungen ist meist dann sinnvoll, wenn Sie versuchen, Icinga zu debuggen, oder Service-[Eventhandler](#) zu testen.

- 0 = keine Service-Wiederholungsprüfungen protokollieren [Default-H]
- 1 = Service-Wiederholungsprüfungen protokollieren [Default-C]

Option Host-Wiederholungsprüfungen-protokollieren (Host Check Retry Logging Option)

Format: **log_host_retries=<0/1>**

Beispiel: **log_host_retries=1**

Diese Variable legt fest, ob Host-Wiederholungsprüfungen protokolliert werden. Die Protokollierung von Host-Wiederholungsprüfungen ist meist dann sinnvoll, wenn Sie versuchen, Icinga zu debuggen, oder Host-[Eventhandler](#) zu testen.

- 0 = keine Host-Wiederholungsprüfungen protokollieren [Default-H]
- 1 = Host-Wiederholungsprüfungen protokollieren [Default-C]

Option Eventhandler-protokollieren (Event Handler Logging Option)

Format: **log_event_handlers=<0/1>**

Beispiel: **log_event_handlers=1**

Diese Variable legt fest, ob Service- und Host-[Eventhandlers](#) protokolliert werden. Eventhandler sind optionale Befehle, die ausgeführt werden können, wenn sich der Zustand eines Hosts oder Service ändert. Die Protokollierung von Eventhandlern ist meist dann sinnvoll, wenn Sie versuchen, Icinga zu debuggen, oder Ihre [Eventhandler](#)-Scripts zu testen.

- 0 = Eventhandler nicht protokollieren
- 1 = Eventhandler protokollieren [Default]

Option initiale Zustände protokollieren (Initial States Logging Option)

Format: **log_initial_states=<0/1>**

Beispiel: **log_initial_states=1**

Diese Variable legt fest, ob Icinga alle anfänglichen Host- und Service-Zustände protokolliert, selbst wenn sie in einem OK-Zustand sind. Anfängliche Service- und Host-Zustände werden normalerweise nur dann protokolliert, wenn es bei der ersten Prüfung ein Problem gibt. Die Aktivierung dieser Option ist sinnvoll, wenn Sie eine Applikation benutzen, die die Protokolldatei abfragt, um Langzeit-Zustandsstatistiken für Services und Hosts zu erstellen.

- 0 = keine anfänglichen Zustände protokollieren (Default)
- 1 = anfängliche Zustände protokollieren

Option externe Befehle protokollieren (External Command Logging Option)

Format: **log_external_commands=<0/1>**

Beispiel: **log_external_commands=1**

Diese Variable legt fest, ob Icinga [externe Befehle](#) protokolliert, die es aus der [externen Befehlsdatei](#) erhält. Anmerkung: diese Option kontrolliert nicht, ob [passive Service-Prüfungen](#) (die eine Art von externen Befehlen sind) protokolliert werden. Zur Aktivierung oder Deaktivierung der Protokollierung von passiven Prüfungen nutzen Sie die [log_passive_checks](#)-Option.

- 0 = keine externen Befehle protokollieren
- 1 = externe Befehle protokollieren (Default)

Option Benutzer von externen Kommandos protokollieren

Format: **log_external_commands_user=<0/1>**

Beispiel: **log_external_commands_user=1**

Diese Option erlaubt es Ihnen, den Namen des Benutzers zu protokollieren, der aktuell externe Kommandos ausführt. Anstatt CMD;cmdargs wird nun CMD;username;cmdargs in die Log-Datei geschrieben, wenn die externe Applikation das korrekt sendet. Weil dies die Kompatibilität mit bestehenden Log-Parsern gefährdet, ist diese Option per Default deaktiviert.



Anmerkung

Diese Option ist ab Icinga 1.4 veraltet, weil Sie nun die Direktive [use_logging](#) benutzen können, um CGI-Befehle zu protokollieren.

- 0 = Den Namen des aktuellen Benutzers von externen Kommandos nicht protokollieren (default)
- 1 = Den Namen des aktuellen Benutzers von externen Kommandos protokollieren



Anmerkung

Diese Option ist verfügbar ab Icinga 1.0.3.

Option passive Prüfungen protokollieren (Passive Check Logging Option)

Format: **log_passive_checks=<0/1>**

Beispiel: **log_passive_checks=1**

Diese Variable legt fest, ob Icinga [passive Host- und Service-Prüfungen](#) protokolliert, die es von der [externen Befehlsdatei](#) bekommt. Wenn Sie eine [verteilte Überwachungsumgebung](#) aufbauen oder planen, eine große Zahl von passiven Prüfungen auf einer regelmäßigen Basis zu behandeln, dann können Sie diese Option deaktivieren, damit Ihre Protokolldatei nicht zu groß

wird.

- 0 = keine passiven Prüfungen protokollieren
- 1 = passive Prüfungen protokollieren (Default)

Option Aktuelle Zustände protokollieren

Format: **log_current_states=<0/1>**

Beispiel: **log_current_states=1**

Diese Variable legt fest, ob Icinga die aktuellen Host- und Service-Zustände nach einer Log-Datei-Rotation protokolliert. Wenn Sie den Wert von log_current_states auf 0 setzen, werden die aktuellen Zustände nach einer Rotation der Log-Datei nicht protokolliert.

- 0 = die aktuellen Host- und Service-Zustände nicht protokollieren
- 1 = die aktuellen Host- und Service-Zustände protokollieren (Default)



Anmerkung

Diese Option ist verfügbar ab Icinga 1.0.3.

Option langen Plugin-Output protokollieren

Format: **log_long_plugin_output=<0/1>**

Beispiel: **log_long_plugin_output=1**

Diese Variable legt fest, ob Icinga die komplette Ausgabe von Plugin (und nicht nur die erste Zeile) protokolliert. Wenn Sie den Wert von log_long_plugin_output auf 1 setzen, werden die kompletten Ausgaben von Plugins protokolliert.

- 0 = nur die erste Zeile von Plugin-Ausgaben protokollieren (Default)
- 1 = die kompletten Plugin-Ausgaben protokollieren



Anmerkung

Diese Option ist verfügbar ab Icinga 1.0.3.

globale Host-Eventhandler Option (Global Host Event Handler Option)

Format: **global_host_event_handler=<command>**

Beispiel: **global_host_event_handler=log-host-event-to-db**

Diese Option erlaubt Ihnen, einen Host-Eventhandler-Befehl anzugeben, der für jeden Host-Zustandswechsel ausgeführt wird. Der globale Eventhandler wird direkt vor dem Eventhandler ausgeführt, den Sie optional in jeder Host-Definition angeben können. Das *Befehls*-Argument ist der Kurzname eines Befehls, den Sie in Ihrer [Objektkonfigurationsdatei](#) angeben. Die maximale Ausführungszeit dieses Befehls kann durch die [event_handler_timeout](#)-Option angegeben werden. Mehr Informationen zu EventHandlers finden Sie [hier](#).

Globale Service-Eventhandler-Option (Global Service Event Handler Option)

Format: **global_service_event_handler=<command>**

Beispiel: **global_service_event_handler=log-service-event-to-db**

Diese Option erlaubt Ihnen, einen Service-Eventhandler-Befehl anzugeben, der für jeden Service-Zustandswechsel ausgeführt wird. Der globale Eventhandler wird direkt vor dem Eventhandler ausgeführt, den Sie optional in jeder Service-Definition angeben können. Das **Befehls**-Argument ist der Kurzname eines Befehls, den Sie in Ihrer [Objektkonfigurationsdatei](#) angeben. Die maximale Ausführungszeit dieses Befehls kann durch die **event_handler_timeout**-Option angegeben werden. Mehr Informationen zu Eventhandlern finden Sie [hier](#).

Eventhandler für "verfolgte" Hosts (Event handlers for stalked hosts)

Eventhandler für "verfolgte" Services (Event handlers for stalked services)

Format: **stalking_event_handlers_for_hosts=<0|1>**

Format: **stalking_event_handlers_for_services=<0|1>**

Example: **stalking_event_handlers_for_hosts=1**

Diese Optionen erlauben Ihnen festzulegen, ob Icinga Eventhandler für "stalked" Hosts oder Services ausführt. Auf diese Weise können Statusinformationsänderungen an externe Systeme weitergeleitet werden.

- 0 = Eventhandler deaktivieren (Default)
- 1 = Eventhandler aktivieren



Anmerkung

Diese Option ist verfügbar ab Icinga 1.0.3.

Benachrichtigungen für "verfolgte" Hosts (Notifications for stalked hosts)

Benachrichtigungen für "verfolgte" Services (Notifications for stalked services)

Format: **stalking_notifications_for_hosts=<0|1>**

Format: **stalking_notifications_for_services=<0|1>**

Example: **stalking_notifications_for_hosts=1**

Diese Optionen legen global für alle Kontakte fest, ob Icinga Benachrichtigungen für "stalked" Hosts oder Services ausführt. Das \$NOTIFICATION_TYPE\$-Makro wird um den Typ "STALKING" erweitert.



Anmerkung

Solange keine stalking_options beim Host/Service definiert sind, haben diese Optionen keine Auswirkungen.

- 0 = Benachrichtigungen deaktivieren (Default)
- 1 = Benachrichtigungen aktivieren



Anmerkung

Diese Option ist verfügbar ab Icinga 1.6.

Ruhezeit zwischen Prüfungen (Inter-Check Sleep Time)

Format: **sleep_time=<seconds>**

Beispiel: **sleep_time=1**

Dies ist die Anzahl von Sekunden, die Icinga "schlafen" wird, bevor es in der Planungswarteschlange (scheduling queue) nach weiteren auszuführenden Host- oder Service-Prüfungen schaut. Beachten Sie, dass Icinga nur schlafen wird, nachdem es anstehende Service-Prüfungen erledigt hat, die in Verzug geraten waren.

Verzögerungsmethode für Service-Prüfungen (Service Inter-Check Delay Method)

Format: **service_inter_check_delay_method=<n/d/s/x.xx>**

Beispiel: **service_inter_check_delay_method=s**

Diese Option erlaubt Ihnen die Kontrolle darüber, wie Service-Prüfungen anfänglich in der Planungswarteschlange "ausgebreitet" werden. Die Verwendung einer "schlauen" Verzögerungsberechnung (der Standard) veranlasst Icinga, ein durchschnittliches Prüfintervall zu berechnen und die anfänglichen Prüfungen aller Services über dieses Intervall zu verteilen, um dadurch CPU-Lastspitzen zu eliminieren. Keine Verzögerung zu benutzen wird *nicht* empfohlen, weil es dafür sorgt, dass die Ausführung aller Service-Prüfungen zur gleichen Zeit geplant wird. Das bedeutet, dass Sie generell hohe CPU-Spitzen haben werden, wenn die Services alle parallel ausgeführt werden. Mehr Informationen dazu, wie die Verzögerung von Service-Prüfungen die Planung dieser Prüfungen beeinflusst, finden Sie [hier](#). Die Werte sind wie folgt:

- n = keine (none) Verzögerung benutzen - planen, dass alle Service-Prüfungen sofort ausgeführt werden (d.h. zur gleichen Zeit!)
- d = eine "dumme" (dumb) Verzögerung von einer Sekunde zwischen Service-Prüfungen benutzen
- s = eine "schlaue" (smart) Verzögerungsberechnung verwenden, um die Service-Prüfungen gleichmäßig zu verteilen (Default)
- x.xx = eine benutzerdefinierte Verzögerung von x.xx Sekunden zwischen den Prüfungen benutzen

maximale Service-Prüfungsverteilung (Maximum Service Check Spread)

Format: **max_service_check_spread=<minutes>**

Beispiel: **max_service_check_spread=30**

Diese Option legt die maximale Anzahl in Minuten fest vom Icinga-Start bis zur Ausführung aller (regelmäßig geplanten) Service-Prüfungen. Diese Option wird automatisch die [Verzögerungsmethode für Service-Prüfungen](#) anpassen (falls notwendig), um sicherzustellen, dass die anfänglichen Prüfungen aller Services in dem Zeitrahmen stattfinden, den Sie angeben. Generell wird diese Optionen keine Auswirkung auf die Planung von Service-Prüfungen haben, wenn die Planungsinformationen mit Hilfe der [use_retained_scheduling_info](#)-Option aufbewahrt werden. Standardwert ist **30** (Minuten).

Service-Verschachtelungsfaktor (Service Interleave Factor)

Format: **service_interleave_factor=<s | x>**

Beispiel: **service_interleave_factor=s**

Diese Variable legt fest, wie Service-Prüfungen verschachtelt werden. Verschachtelung erlaubt eine gleichmäßige Verteilung von Service-Prüfungen, reduzierte Last auf entfernten Hosts und schnellere Erkennung von Host-Problemen. Das Setzen des Wertes auf 1 ist gleichbedeutend mit keiner Verschachtelung der Service-Prüfungen (so arbeiteten die Icinga-Version bis 0.0.5). Setzen Sie diesen Wert auf s (schlau/SMART) für die automatische Berechnung, solange Sie keinen bestimmten Grund für die Änderung haben. Der beste Weg zum Verständnis, wie Verschachtelung funktioniert, ist der Blick auf das [status CGI](#) (detail view), während Icinga startet. Sie sollten sehen, dass die Service-Prüfergebnisse verteilt werden, während sie auftauchen. Mehr Informationen dazu, wie Verschachtelung funktioniert, finden Sie [hier](#).

- $x =$ eine Zahl gleich oder größer 1, die den zu benutzenden Verschachtelungsfaktor angibt. Ein Verschachtelungsfaktor von 1 bedeutet keine Verschachtelung von Service-Prüfungen
- $s =$ eine "schlaue" (smart) Verschachtelungsberechnung benutzen (Default)

maximale Anzahl gleichzeitiger Service-Prüfungen (Maximum Concurrent Service Checks)

Format: **max_concurrent_checks=<max_checks>**

Beispiel: **max_concurrent_checks=20**

Diese Option erlaubt Ihnen die Angabe der maximalen Anzahl von Service-Prüfungen, die zu irgendeiner Zeit gleichzeitig ausgeführt werden. Das Angeben eines Wertes von 1 verhindert grundsätzlich die Ausführung von parallelen Service-Prüfungen. Der Wert 0 (der Standard) sorgt dafür, dass es keine Beschränkung der Anzahl von gleichzeitig ausgeführten Service-Prüfungen gibt. Sie müssen den Wert auf der Basis der Systemressourcen anpassen, die Ihr Icinga-Rechner zur Verfügung stellt, da er direkt die maximale Last des Systems beeinflusst (Prozessorauslastung, Speicher, usw.). Mehr Informationen dazu, wie viele parallele Prüfungen Sie zulassen sollten, finden Sie [hier](#).

Prüfergebnis-Erntefrequenz (Check Result Reaper Frequency)

Format: **check_result_reaper_frequency=<frequency_in_seconds>**

Beispiel: **check_result_reaper_frequency=5**

Diese Option erlaubt Ihnen die Kontrolle über die Frequenz *in Sekunden* der Prüfergebnis-"Ernte"-Ereignisse. "Ernte"-Ereignisse verarbeiten die Ergebnisse von Host- und Service-Prüfungen, die beendet wurden. Diese Ereignisse bilden den Kern der Überwachungslogik von Icinga.

maximale Prüfergebnis-Erntezeit (Maximum Check Result Reaper Time)

Format: **max_check_result_reaper_time=<seconds>**

Beispiel: **max_check_result_reaper_time=30**

Diese Option erlaubt Ihnen die Kontrolle der maximalen Zeit *in Sekunden*, die Host- und Service-Prüfergebnis-"Ernte"-Ereignisse laufen dürfen. "Ernte"-Ereignisse verarbeiten die Ergebnisse von Host- und Service-Prüfungen, die beendet sind. Wenn es eine Menge von Ergebnissen zu verarbeiten gibt, können Ernte-Ereignisse lange brauchen, um zu enden, was die pünktliche Ausführung von neuen Host- und Service-Prüfungen verzögern könnte. Diese Variable erlaubt Ihnen die Begrenzung der Zeit, die ein einzelnes Ernte-Ereignis laufen darf, bevor es die Kontrolle an andere Teile der Icinga-Überwachungslogik zurückgibt.

Prüfergebnis-Pfad (Check Result Path)

Format: **check_result_path=<path>**

Beispiel: **check_result_path=/var/spool/nagios/checkresults**

Diese Option legt fest, welches Verzeichnis Icinga benutzen wird, um temporär Host- und Service-Prüfergebnisse zu speichern, bevor sie verarbeitet werden. Diese Verzeichnis sollte nicht benutzt werden, um andere Dateien dort zu speichern, weil Icinga dieses Verzeichnis periodisch von alten Dateien säubern wird (mehr Informationen dazu finden Sie bei der [max_check_result_file_age](#)-Option).

Anmerkung: stellen Sie sicher, dass nur eine einzelne Icinga-Instanz Zugriff auf den Prüfergebnispfad hat. Wenn mehrere Icinga-Instanzen Zugriff auf das gleiche Verzeichnis haben, werden Sie Probleme bekommen, weil Prüfergebnisse von der falschen Icinga-Instanz verarbeitet wurden!

maximales Alter der Prüfergebnisdatei (Max Check Result File Age)

Format: **max_check_result_file_age=<seconds>**

Beispiel: **max_check_result_file_age=3600**

Diese Option legt das maximale Alter in Sekunden fest, die Daten aus den Prüfergebnisdateien im [check_result_path](#)-Verzeichnis als gültig angesehen werden. Prüfergebnisdateien, die älter als dieser Schwellwert sind, werden von Icinga gelöscht und die darin enthaltenen Daten werden nicht verarbeitet. Durch die Angabe eines Wertes von Null (0) bei dieser Option wird Icinga alle Prüfergebnisdateien verarbeiten - selbst wenn sie älter als Ihre Hardware sind :-).

Verzögerungsmethode für Host-Prüfungen (Host Inter-Check Delay Method)

Format: **host_inter_check_delay_method=<n/d/s/x.xx>**

Beispiel: **host_inter_check_delay_method=s**

Diese Option erlaubt Ihnen die Kontrolle darüber, wie Host-Prüfungen, *die für eine regelmäßige Prüfung geplant sind*, anfänglich in der Planungswarteschlange "ausgebreitet" werden. Die Verwendung einer "schlauen" Verzögerungsberechnung (der Standard) veranlasst Icinga, ein durchschnittliches Prüfintervall zu berechnen und die anfänglichen Prüfungen aller Hosts über dieses Intervall zu verteilen, um dadurch CPU-Lastspitzen zu eliminieren. Keine Verzögerung zu benutzen wird generell *nicht* empfohlen, weil es dafür sorgt, dass die Ausführung aller

Host-Prüfungen zur gleichen Zeit geplant wird. Mehr Informationen dazu, wie die Verzögerung von Host-Prüfungen die Planung dieser Prüfungen beeinflusst, finden Sie [hier](#). Die Werte sind wie folgt:

- n = keine (none) Verzögerung benutzen - planen, dass alle Host-Prüfungen sofort ausgeführt werden (d.h. zur gleichen Zeit!)
- d = eine "dumme" (dumb) Verzögerung von einer Sekunde zwischen Host-Prüfungen benutzen
- s = eine "schlaue" (smart) Verzögerungsberechnung verwenden, um die Host-Prüfungen gleichmäßig zu verteilen (Default)
- x.xx = eine benutzerdefinierte Verzögerung von x.xx Sekunden zwischen den Prüfungen benutzen

maximale Host-Prüfungsverteilung (Maximum Host Check Spread)

Format: **max_host_check_spread=<minutes>**

Beispiel: **max_host_check_spread=30**

Diese Option legt die maximale Anzahl in Minuten fest vom Icinga-Start bis zur Ausführung aller (regelmäßig geplanten) Host-Prüfungen. Diese Option wird automatisch die [Verzögerungsmethode für Host-Prüfungen](#) anpassen (falls notwendig), um sicherzustellen, dass die anfänglichen Prüfungen aller Hosts in dem Zeitrahmen stattfinden, den Sie angeben. Generell wird diese Optionen keine Auswirkung auf die Planung von Host-Prüfungen haben, wenn die Planungsinformationen mit Hilfe der [use_retained_scheduling_info](#)-Option aufbewahrt werden. Standardwert ist **30** (Minuten).

Zeitintervalllänge (Timing Interval Length)

Format: **interval_length=<seconds>**

Beispiel: **interval_length=60**

Dies ist die Zahl von Sekunden pro "Zeitintervall" (unit interval), das für die Steuerung in der Planungswarteschlange, bei erneuten Benachrichtigungen usw. verwendet wird. "Zeitintervalle" werden in den Objektkonfigurationsdateien benutzt, um festzulegen, wie oft eine Service-Prüfung ausgeführt, ein Kontakt erneut informiert wird usw.

Wichtig: Der Standardwert hierfür ist auf 60 eingestellt, was bedeutet, dass ein "Zeitwert" von eins (1) in der Objektkonfigurationsdatei 60 Sekunden bedeutet (eine Minute). Wir haben keine anderen Werte für diese Variable ausprobiert, also machen Sie Änderungen auf Ihr eigenes Risiko, wenn Sie etwas anderes einstellen!

automatische Wiedereinplanungs-Option (Auto-Rescheduling Option)

Format: **auto_reschedule_checks=<0/1>**

Beispiel: **auto_reschedule_checks=1**

Diese Option legt fest, ob Icinga versucht, aktive Host- und Service-Prüfungen automatisch erneut einzuplanen, um sie über die Zeit "auszugleichen" (smooth out). Dies kann helfen, die Last des Überwachungsrechners auszubalancieren, da es versucht, die Zeit zwischen

aufeinander folgenden Prüfungen einheitlich zu halten, auf Kosten der Ausführung von Prüfungen mit einer rigideren Planung.

WARNUNG: DIES IST EIN EXPERIMENTELLES FEATURE UND KÖNNTE IN DER ZUKUNFT ENTFERNT WERDEN. DIE AKTIVIERUNG DIESER OPTION KANN DIE LEISTUNG REDUZIEREN - STATT SIE ZU ERHÖHEN - WENN SIE UNGEEIGNET BENUTZT WIRD!

automatisches Wiedereinplanungs-Intervall (Auto-Rescheduling Interval)

Format: **auto_rescheduling_interval=<seconds>**

Beispiel: **auto_rescheduling_interval=30**

Diese Option legt fest, wie oft (in Sekunden) Icinga versuchen wird, automatisch Prüfungen erneut einzuplanen. Diese Option ist nur wirksam, wenn die [auto_reschedule_checks](#)-Option aktiviert ist. Standard ist 30 Sekunden.

WARNUNG: DIES IST EIN EXPERIMENTELLES FEATURE UND KÖNNTE IN DER ZUKUNFT ENTFERNT WERDEN. DIE AKTIVIERUNG DIESER OPTION KANN DIE LEISTUNG REDUZIEREN - STATT SIE ZU ERHÖHEN - WENN SIE UNGEEIGNET BENUTZT WIRD!

automatisches Wiedereinplanungsfenster (Auto-Rescheduling Window)

Format: **auto_rescheduling_window=<seconds>**

Beispiel: **auto_rescheduling_window=180**

Diese Option legt das Zeit-"Fenster" fest (in Sekunden), auf das Icinga blickt, wenn es automatisch Prüfungen erneut einplant. Nur Host- und Service-Prüfungen, die in den nächsten X Sekunden (festgelegt durch diese Variable) stattfinden, werden erneut eingeplant. Diese Option ist nur wirksam, wenn die [auto_reschedule_checks](#)-Option aktiviert ist. Standard ist 180 Sekunden (3 Minuten).

WARNING: DIES IST EIN EXPERIMENTELLES FEATURE UND KÖNNTE IN DER ZUKUNFT ENTFERNT WERDEN. DIE AKTIVIERUNG DIESER OPTION KANN DIE LEISTUNG REDUZIEREN - STATT SIE ZU ERHÖHEN - WENN SIE UNGEEIGNET GENUTZT WIRD!

Option aggressive Host-Prüfung (Aggressive Host Checking Option)

Format: **use_aggressive_host_checking=<0/1>**

Beispiel: **use_aggressive_host_checking=0**

Icinga versucht, schlau zu sein, wie und wann es den Status von Hosts prüft. Im Allgemeinen wird die Deaktivierung dieser Option Icinga dazu veranlassen, einige schlauere Entscheidungen zu treffen und Hosts ein bisschen schneller zu prüfen. Die Aktivierung dieser Option wird den Zeitaufwand zur Prüfung von Hosts erhöhen, aber es mag die Zuverlässigkeit ein wenig steigern. Solange Sie keine Probleme damit haben, dass Icinga die Erholung eines Hosts nicht korrekt erkennt, würden wir empfehlen, diese Option **nicht** zu aktivieren.

- 0 = keine aggressive Host-Prüfung benutzen (Default)
- 1 = aggressive Host-Prüfung benutzen

Option passive Host-Prüfung übersetzen (Translate Passive Host Checks Option)

Format: **translate_passive_host_checks=<0/1>**

Beispiel: **translate_passive_host_checks=1**

Diese Option legt fest, ob Icinga DOWN/UNREACHABLE-Ergebnisse von passiven Host-Prüfungen in ihre "korrekten" Zustände vom Standpunkt der lokalen Icinga-Instanz aus übersetzt. Dies kann sehr nützlich in verteilten und Failover-Umgebungen sein. Mehr Informationen zur Übersetzung von passiven Prüfergebnissen finden Sie [hier](#).

- 0 = Prüfübersetzung deaktivieren (Default)
- 1 = Prüfübersetzung aktivieren

Option passive Host-Prüfungen sind SOFT (Passive Host Checks Are SOFT Option)

Format: **passive_host_checks_are_soft=<0/1>**

Beispiel: **passive_host_checks_are_soft=1**

Diese Option legt fest, ob Icinga [passive Host-Prüfungen](#) als HARD- oder SOFT-Zustände behandelt. Per Default wird ein passives Prüfergebnis einen Host in einen [HARD-Status](#) setzen. Sie können dieses Verhalten durch aktivieren dieser Option verändern.

- 0 = passive Host-Prüfungen sind HARD (Default)
- 1 = passive Host-Prüfungen sind SOFT

Option vorausschauende Host-Abhängigkeitsprüfung (Predictive Host Dependency Checks Option)

Format: **enable_predictive_host_dependency_checks=<0/1>**

Beispiel: **enable_predictive_host_dependency_checks=1**

Diese Option legt fest, ob Icinga vorausschauende Prüfungen von Hosts ausführen soll, von denen andere abhängig sind (wie in [Host-Abhängigkeiten](#) definiert) für einen bestimmten Host, dessen Zustand wechselt. Vorausschauende Prüfungen helfen dabei, die Abhängigkeitslogik so genau wie möglich zu machen. Mehr Informationen darüber, wie vorausschauende Prüfungen arbeiten, finden Sie [hier](#).

- 0 = vorausschauende Prüfungen deaktivieren
- 1 = vorausschauende Prüfungen aktivieren (Default)

Option vorausschauende Service-Abhängigkeitsprüfung (Predictive Service Dependency Checks Option)

Format: **enable_predictive_service_dependency_checks=<0/1>**

Beispiel: **enable_predictive_service_dependency_checks=1**

Diese Option legt fest, ob Icinga vorausschauende Prüfungen von Services ausführen soll, von denen andere abhängig sind (wie in [Service-Abhängigkeiten](#) definiert) für einen bestimmten Service, dessen Zustand wechselt. Vorausschauende Prüfungen helfen dabei, die Abhängigkeitslogik so genau wie möglich zu machen. Mehr Informationen darüber, wie vorausschauende Prüfungen arbeiten, finden Sie [hier](#).

- 0 = vorausschauende Prüfungen deaktivieren
- 1 = vorausschauende Prüfungen aktivieren (Default)

Horizont für zwischengespeicherte Host-Prüfungen (Cached Host Check Horizon)

Format: **cached_host_check_horizon=<seconds>**

Beispiel: **cached_host_check_horizon=15**

Diese Option legt die maximale Zeit (in Sekunden) fest, die der Zustand einer vorherigen Host-Prüfung als aktuell angesehen wird. Zwischengespeicherte Host-Zustände (von Host-Prüfungen, die aktueller sind als die in diesem Wert angegebene Zeit) können die Leistung von Host-Prüfungen immens steigern. Ein zu hoher Wert für diese Option kann in (vorübergehend) ungenauen Host-Zuständen resultieren, während ein niedriger Wert zu einem Leistungseinbruch bei Host-Prüfungen führen kann. Benutzen Sie einen Wert von 0, wenn Sie die Zwischenspeicherung von Host-Prüfungen deaktivieren wollen. Mehr Informationen zu zwischengespeicherten Prüfungen finden Sie [hier](#).

Horizont für zwischengespeicherte Service-Prüfungen (Cached Service Check Horizon)

Format: **cached_service_check_horizon=<seconds>**

Beispiel: **cached_service_check_horizon=15**

Diese Option legt die maximale Zeit (in Sekunden) fest, die der Zustand einer vorherigen Service-Prüfung als aktuell angesehen wird. Zwischengespeicherte Service-Zustände (von Service-Prüfungen, die aktueller sind als die in diesem Wert angegebene Zeit) können die Leistung von Service-Prüfungen steigern, wenn eine Menge von [Service-Abhängigkeiten](#) benutzt werden. Ein zu hoher Wert für diese Option kann in (vorübergehend) ungenauen Service-Zuständen resultieren, während ein niedriger Wert zu einem Leistungseinbruch bei Service-Prüfungen führen kann. Benutzen Sie einen Wert von 0, wenn Sie die Zwischenspeicherung von Service-Prüfungen deaktivieren wollen. Mehr Informationen zu zwischengespeicherten Prüfungen finden Sie [hier](#).

Option Verbesserungen für große Installationen (Large Installation Tweaks Option)

Format: **use_large_installation_tweaks=<0/1>**

Beispiel: **use_large_installation_tweaks=0**

Diese Option legt fest, ob der Icinga-Daemon verschiedene Abkürzungen nimmt, um die Leistung zu steigern. Diese Abkürzungen resultieren im Verlust einiger Features, aber große Installationen werden wahrscheinlich einen hohen Nutzen davon haben. Mehr Informationen, welche Optimierungen vorgenommen werden, wenn Sie diese Option aktivieren, finden Sie [hier](#).

- 0 = keine Verbesserungen verwenden (Default)
- 1 = Verbesserungen verwenden

Kindprozess-Speicher-Option (Child Process Memory Option)

Format: **free_child_process_memory=<0/1>**

Beispiel: **free_child_process_memory=0**

Diese Option legt fest, ob Icinga Speicher in Kindprozessen freigibt, wenn sie vom Hauptprozess ge"fork()"ed werden. Per Default gibt Icinga den Speicher frei. Wenn allerdings die [use_large_installation_tweaks](#)-Option aktiviert ist, wird der Speicher nicht freigegeben. Durch Definition dieser Option in Ihrer Konfigurationsdatei sind Sie in der Lage, das Verhalten einzustellen, das Sie möchten.

- 0 = Speicher nicht freigeben
- 1 = Speicher freigeben

Kindprozesse zweimal "fork()"en

Format: **child_processes_fork_twice=<0/1>**

Beispiel: **child_processes_fork_twice=0**

Diese Option legt fest, ob Icinga Kindprozesse zweimal "fork()"ed, wenn es Host- und Service-Prüfungen ausführt. Per Default "fork()"ed Icinga zweimal. Wenn allerdings die [use_large_installation_tweaks](#)-Option aktiviert ist, "fork()"ed Icinga nur einmal. Durch Definition dieser Option in Ihrer Konfigurationsdatei sind Sie in der Lage, das Verhalten einzustellen, das Sie möchten.

- 0 = nur einmal "fork()"en
- 1 = zweimal "fork()"en

Umgebungsmakros-Option

Format: **enable_environment_macros=<0/1>**

Beispiel: **enable_environment_macros=0**

Diese Option legt fest, ob Icinga alle Standard-Makros als Umgebungsvariablen für Prüfungen, Benachrichtigungen, Eventhandler usw. zur Verfügung stellt. In großen Installationen kann dies problematisch sein, weil es zusätzlichen Speicher (und wichtiger) mehr CPU benötigt, um die Werte aller Makros zu berechnen und sie der Umgebung zur Verfügung zu stellen.

- 0 = Makros nicht als Umgebungsvariablen verfügbar machen (Default)
- 1 = Makros als Umgebungsvariablen verfügbar machen



Anmerkung

Ab Icinga 1.5 hat sich der Default geändert. Vorher war diese Direktive aktiviert, nun ist sie *deaktiviert*.

Flattererkennungsoption

Format: **enable_flap_detection=<0/1>**

Beispiel: **enable_flap_detection=0**

Diese Option legt fest, ob Icinga versucht festzustellen, ob Hosts und Services "flattern". Flattern tritt auf, wenn ein Host oder Service zu schnell zwischen verschiedenen Zuständen wechselt, was in einem Bombardement von Benachrichtigungen resultiert. Wenn Icinga erkennt, dass ein Host oder Service flattert, wird es vorübergehend Benachrichtigungen für diesen Host/Service unterdrücken, bis das Flattern endet. Flattererkennung ist sehr experimentell zu diesem Zeitpunkt, also benutzen Sie diese Option mit Vorsicht! Mehr Informationen dazu, wie Flattererkennung und Behandlung funktionieren, finden Sie [hier](#). Anmerkung: Wenn Sie [Statusaufbewahrung](#) aktiviert haben, wird Icinga diese Einstellung ignorieren, wenn es (erneut) startet und die letzte bekannte Einstellung dieser Option nutzen (wie sie in der [Statusaufbewahrungsdatei](#) abgelegt ist), *es sei denn*, Sie haben die [use_retained_program_state](#)-Option deaktiviert. Wenn Sie diese Option ändern möchten, während die Statusaufbewahrung aktiviert ist (und die Option [use_retained_program_state](#) aktiviert ist), müssen Sie den entsprechenden [externen Befehl](#) benutzen oder sie über das Web-Interface ändern.

- 0 = Flattererkennung deaktivieren (Default)
- 1 = Flattererkennung aktivieren

niedriger Service-Flatterschwellwert (Low Service Flap Threshold)

Format: **low_service_flap_threshold=<percent>**

Beispiel: **low_service_flap_threshold=25.0**

Diese Option wird benutzt, um den niedrigen Schwellwert für die Erkennung von Service-Flattern zu setzen. Mehr Informationen dazu, wie Flattererkennung und Behandlung funktionieren (und wie diese Option Dinge beeinflusst), finden Sie [hier](#).

hoher Service-Flatterschwellwert (High Service Flap Threshold)

Format: **high_service_flap_threshold=<percent>**

Beispiel: **high_service_flap_threshold=50.0**

Diese Option wird benutzt, um den hohen Schwellwert für die Erkennung von Service-Flattern zu setzen. Mehr Informationen dazu, wie Flattererkennung und Behandlung funktionieren (und wie diese Option Dinge beeinflusst), finden Sie [hier](#).

niedriger Host-Flatterschwellwert (Low Host Flap Threshold)

Format: **low_host_flap_threshold=<percent>**

Beispiel: **low_host_flap_threshold=25.0**

Diese Option wird benutzt, um den niedrigen Schwellwert für die Erkennung von Host-Flattern zu setzen. Mehr Informationen dazu, wie Flattererkennung und Behandlung funktionieren (und wie diese Option Dinge beeinflusst), finden Sie [hier](#).

hoher Host-Flatterschwellwert (High Host Flap Threshold)

Format: **high_host_flap_threshold=<percent>**

Beispiel: **high_host_flap_threshold=50.0**

Diese Option wird benutzt, um den hohen Schwellwert für die Erkennung von Host-Flattern zu setzen. Mehr Informationen dazu, wie Flattererkennung und Behandlung funktionieren (und wie diese Option Dinge beeinflusst), finden Sie [hier](#).

Soft-Statusabhängigkeitsoption (Soft State Dependencies Option)

Format: **soft_state_dependencies=<0/1>**

Beispiel: **soft_state_dependencies=0**

Diese Option legt fest, ob Icinga Soft-Statusinformationen benutzen soll, um [Host- und Serviceabhängigkeiten](#) zu prüfen. Normalerweise wird Icinga nur den letzten Hard-Status des Hosts oder Service verwenden, wenn Abhängigkeiten geprüft werden. Wenn Sie möchten, dass der letzte Zustand (unabhängig davon, ob es ein Soft- oder Hard-[Zustandstyp](#) ist), dann aktivieren Sie diese Option.

- 0 = keine Soft-Status-Abhängigkeiten benutzen (Default)
- 1 = Soft-Status-Abhängigkeiten benutzen

Service-Prüfungs-Zeitüberschreitung (Service Check Timeout)

Format: **service_check_timeout=<seconds>**

Beispiel: **service_check_timeout=60**

Dies ist die maximale Zahl von Sekunden, die Service-Prüfungen laufen dürfen. Wenn Prüfungen diese Grenze überschreiten, werden sie abgebrochen (killed) und ein CRITICAL-Zustand wird zurückgeliefert. Außerdem wird ein Fehler protokolliert.

Es gibt oft eine weitverbreitete Verwirrung, was diese Option wirklich tut. Es ist als allerletzter Versuch gedacht, wenn Plugins sich "daneben benehmen" und nicht innerhalb einer bestimmten Zeit enden. Sie sollte auf einen hohen Wert (z.B. 60 Sekunden oder mehr) gesetzt werden, so dass jede Service-Prüfung normalerweise innerhalb dieser Zeit beendet ist. Wenn eine Service-Prüfung länger läuft, dann wird Icinga diese Prüfung abbrechen, weil es denkt, dass es sich um einen außer Kontrolle geratenen Prozess handelt.

Service-Prüfungs-Zeitüberschreitungs-Status

Format: **service_check_timeout_state=<c/w/u/o>**

Example: **service_check_timeout_state=u**

Diese Option legt den Status fest den ein Service erhält, wenn er in einen Timeout läuft. Er also nicht in der, mit service_check_timeout definierten Zeit, eine Rückmeldung bekommt. Das kann sehr nützlich sein, wenn eine Maschine gerade einen sehr hohen Load hat und der Service-Check fehlschlägt und dann kein Critical-Alarm generiert werden soll. Der Default-Wert wurde von service_check_timeout_state=c zu service_check_timeout_state=u in Icinga 1.0.1 geändert.

Host-Prüfungs-Zeitüberschreitung (Host Check Timeout)

Format: **host_check_timeout=<seconds>**

Beispiel: **host_check_timeout=60**

Dies ist die maximale Zahl von Sekunden, die Host-Prüfungen laufen dürfen. Wenn Prüfungen diese Grenze überschreiten, werden sie abgebrochen (killed), ein CRITICAL-Zustand wird zurückgeliefert und der Host als "DOWN" angesehen. Außerdem wird ein Fehler protokolliert.

Es gibt oft eine weitverbreitete Verwirrung, was diese Option wirklich tut. Es ist als allerletzter Versuch gedacht, wenn Plugins sich "daneben benehmen" und nicht innerhalb einer bestimmten Zeit enden. Sie sollte auf einen hohen Wert (z.B. 60 Sekunden oder mehr) gesetzt werden, so dass jede Host-Prüfung normalerweise innerhalb dieser Zeit beendet ist. Wenn eine Host-Prüfung länger läuft, dann wird Icinga diese Prüfung abbrechen, weil es denkt, dass es sich um einen außer Kontrolle geratenen Prozess handelt.

Eventhandler-Zeitüberschreitung

Format: **event_handler_timeout=<seconds>**

Beispiel: **event_handler_timeout=60**

Dies ist die maximale Zahl von Sekunden, die Eventhandler laufen dürfen. Wenn ein Eventhandler diese Grenze überschreitet, wird er abgebrochen (killed) und eine Warnung protokolliert.

Es gibt oft eine weitverbreitete Verwirrung, was diese Option wirklich tut. Es ist als allerletzter Versuch gedacht, wenn Befehle sich "daneben benehmen" und nicht innerhalb einer bestimmten Zeit enden. Sie sollte auf einen hohen Wert (z.B. 60 Sekunden oder mehr) gesetzt werden, so dass jeder Eventhandler normalerweise innerhalb dieser Zeit beendet ist. Wenn ein Eventhandler länger läuft, dann wird Icinga diesen Eventhandler abbrechen, weil es denkt, dass es sich um einen außer Kontrolle geratenen Prozess handelt.

Benachrichtigungs-Zeitüberschreitung

Format: **notification_timeout=<seconds>**

Beispiel: **notification_timeout=60**

Dies ist die maximale Zahl von Sekunden, die Benachrichtigungsbefehle laufen dürfen. Wenn ein Benachrichtigungsbefehl diese Grenze überschreitet, wird er abgebrochen (killed) und eine Warnung protokolliert.

Es gibt oft eine weitverbreitete Verwirrung, was diese Option wirklich tut. Es ist als allerletzter Versuch gedacht, wenn Befehle sich "daneben benehmen" und nicht innerhalb einer bestimmten Zeit enden. Sie sollte auf einen hohen Wert (z.B. 60 Sekunden oder mehr) gesetzt werden, so dass jeder Benachrichtigungsbefehl normalerweise innerhalb dieser Zeit beendet ist. Wenn ein Benachrichtigungsbefehl länger läuft, dann wird Icinga diesen Benachrichtigungsbefehl abbrechen, weil es denkt, dass es sich um einen außer Kontrolle geratenen Prozess handelt.

Zwangsvorfolgungs-Service-Prozessor-Zeitüberschreitung (Obsessive Compulsive Service Processor Timeout)

Format: **ocsp_timeout=<seconds>**

Beispiel: **ocsp_timeout=5**

Dies ist die maximale Zahl von Sekunden, die ein [Zwangsvorfolgungs-Service-Prozessor-Befehl](#) (obsessive compulsive service processor command) laufen darf. Wenn ein Befehl diese Grenze überschreitet, wird er abgebrochen (killed) und eine Warnung protokolliert.

Zwangsvorfolgungs-Host-Prozessor-Zeitüberschreitung (Obsessive Compulsive Host Processor Timeout)

Format: **ochp_timeout=<seconds>**

Beispiel: **ochp_timeout=5**

Dies ist die maximale Zahl von Sekunden, die ein [Zwangsvorfolgungs-Host-Prozessor-Befehl](#) (obsessive compulsive host processor command) laufen darf. Wenn ein Befehl diese Grenze überschreitet, wird er abgebrochen (killed) und eine Warnung protokolliert.

Performance-Daten-Prozessorbefehls-Zeitüberschreitung (Performance Data Processor Command Timeout)

Format: **perfdata_timeout=<seconds>**

Beispiel: **perfdata_timeout=5**

Dies ist die maximale Zahl von Sekunden, die ein [Host-Performance-Daten-Prozessorbefehl](#) (host performance data processor command) oder [Service-Performance-Daten-Prozessorbefehl](#) (service performance data processor command) laufen darf. Wenn ein Befehl diese Grenze überschreitet, wird er abgebrochen (killed) und eine Warnung protokolliert.

Verfolgung-von-Services-Option (Obsess Over Services Option)

Format: **obsess_over_services=<0/1>**

Beispiel: **obsess_over_services=1**

Dieser Wert legt fest, ob Icinga Service-Prüfergebnisse "verfolgt" (obsess) und den [Zwangsvorfolgungs-Service-Prozessorbefehl](#) ausführt, den Sie angeben. Nun ja - ein komischer Name, aber das ist alles, was Ethan Galstad eingefallen ist. Diese Option ist nützlich, um [verteilte Überwachung](#) durchzuführen. Wenn Sie keine verteilte Überwachung machen, dann aktivieren Sie diese Option nicht.

- 0 = Services nicht verfolgen (Default)
- 1 = Services verfolgen

Zwangsvorfolgungs-Service-Prozessorbefehl (Obsessive Compulsive Service Processor Command)

Format: **ocsp_command=<command>**

Beispiel: **ocsp_command=obsessive_service_handler**

Diese Option erlaubt Ihnen einen Befehl anzugeben, der nach *jeder* Service-Prüfung ausgeführt wird, was bei [verteilter Überwachung](#) nützlich sein kann. Dieser Befehl wird nach [Eventhandler](#)- oder [Benachrichtigungs](#)-Befehlen ausgeführt. Das *command*-Argument ist der Kurzname einer [command-Definition](#), die Sie in Ihrer Objektkonfigurationsdatei angeben. Die maximale Zeit, die dieser Befehl laufen darf, wird mit der [ocsp_timeout](#)-Option kontrolliert. Mehr Informationen zu verteilter Überwachung finden Sie [hier](#). Dieser Befehl wird nur ausgeführt, wenn die [obsess_over_services](#)-Option global aktiviert ist und wenn die [obsess_over_service](#)-Direktive in der [Service-Definition](#) aktiviert ist.

Verfolgung-von-Hosts-Option (Obsess Over Hosts Option)

Format: **obsess_over_hosts=<0/1>**

Beispiel: **obsess_over_hosts=1**

Dieser Wert legt fest, ob Icinga Host-Prüfergebnisse "verfolgt" (obsess) und den [Zwangsvorfolgungs-Host-Prozessorbefehl](#) ausführt, den Sie angeben. Nun ja - ein komischer Name, aber das ist alles, was Ethan Galstad eingefallen ist. Diese Option ist nützlich, um [verteilte Überwachung](#) durchzuführen. Wenn Sie keine verteilte Überwachung machen, dann aktivieren Sie diese Option nicht.

- 0 = Hosts nicht verfolgen (Default)
- 1 = Hosts verfolgen

Zwangsvorfolgungs-Host-Prozessorbefehl (Obsessive Compulsive Host Processor Command)

Format: **ochp_command=<command>**

Beispiel: **ochp_command=obsessive_host_handler**

Diese Option erlaubt Ihnen einen Befehl anzugeben, der nach *jeder* Host-Prüfung ausgeführt wird, was bei [verteilter Überwachung](#) nützlich sein kann. Dieser Befehl wird nach [Eventhandler](#)- oder [Benachrichtigungs](#)-Befehlen ausgeführt. Das *command*-Argument ist der Kurzname einer [command-Definition](#), die Sie in Ihrer Objektkonfigurationsdatei angeben. Die maximale Zeit, die dieser Befehl laufen darf, wird mit der [ochp_timeout](#)-Option kontrolliert. Mehr Informationen zu verteilter Überwachung finden Sie [hier](#). Dieser Befehl wird nur ausgeführt, wenn die [obsess_over_hosts](#)-Option global aktiviert ist und wenn die [obsess_over_hosts](#)-Direktive in der [Host-Definition](#) aktiviert ist.

Performance-Daten-Verarbeitungsoption (Performance Data Processing Option)

Format: **process_performance_data=<0/1>**

Beispiel: **process_performance_data=1**

Dieser Wert legt fest, ob Icinga [Performance-Daten](#) von Host- und Service-Prüfungen verarbeitet.

- 0 = keine Performance-Daten verarbeiten (Default)
- 1 = Performance-Daten verarbeiten

Host-Performance-Daten-Verarbeitungsbefehl (Host Performance Data Processing Command)

Format: **host_perfdata_command=<command>**

Beispiel: **host_perfdata_command=process-host-perfdata**

Diese Option erlaubt es Ihnen, einen Befehl anzugeben, der nach *jeder* Host-Prüfung ausgeführt wird, um Host-Performance-Daten zu verarbeiten, die von der Prüfung zurückgeliefert worden sein könnten. Das *command*-Argument ist der Kurzname einer [command-Definition](#), die Sie in Ihrer Objektkonfigurationsdatei angeben. Dieser Befehl wird nur ausgeführt, wenn die [process_performance_data](#)-Option global aktiviert ist und wenn die [process_perf_data](#)-Direktive in der [Host-Definition](#) aktiviert ist.

Service-Performance-Daten-Verarbeitungsbefehl (Service Performance Data Processing Command)

Format: **service_perfdata_command=<command>**

Beispiel: **service_perfdata_command=process-service-perfdata**

Diese Option erlaubt es Ihnen, einen Befehl anzugeben, der nach *jeder* Service-Prüfung ausgeführt wird, um Service-Performance-Daten zu verarbeiten, die von der Prüfung zurückgeliefert worden sein könnten. Das *command*-Argument ist der Kurzname einer [command-Definition](#), die Sie in Ihrer Objektkonfigurationsdatei angeben. Dieser Befehl wird nur ausgeführt, wenn die [process_performance_data](#)-Option global aktiviert ist und wenn die [process_perf_data](#)-Direktive in der [Service-Definition](#) aktiviert ist.

Host-Performance-Daten-Datei (Host Performance Data File)

Format: **host_perfdata_file=<file_name>**

Beispiel: **host_perfdata_file=/usr/local/icinga/var/host-perfdata.dat**

Diese Option erlaubt es Ihnen, eine Datei anzugeben, in die nach jeder Host-Prüfung Performance-Daten geschrieben werden. Daten werden in die Performance-Datei geschrieben, wie es in der [host_perfdata_file_template](#)-Option angegeben ist. Performance-Daten werden nur in diese Datei geschrieben, wenn die [process_performance_data](#)-Option global aktiviert ist und wenn die [process_perf_data](#)-Direktive in der [Host-Definition](#) aktiviert ist.

Service-Performance-Daten-Datei (Service Performance Data File)

Format: **service_perfdata_file=<file_name>**

Beispiel: **service_perfdata_file=/usr/local/icinga/var/service-perfdata.dat**

Diese Option erlaubt es Ihnen, eine Datei anzugeben, in die nach jeder Service-Prüfung Performance-Daten geschrieben werden. Daten werden in die Performance-Datei geschrieben, wie es in der [service_perfdata_file_template](#)-Option angegeben ist. Performance-Daten werden nur in diese Datei geschrieben, wenn die [process_performance_data](#)-Option global aktiviert ist und wenn die [process_perf_data](#)-Direktive in der [Service-Definition](#) aktiviert ist.

Host-Performance-Daten-Dateivorlage (Host Performance Data File Template)

Format: **host_perfdata_file_template=<template>**

Beispiel: **host_perfdata_file_template=[HOSTPERFDATA]\t\$TIMET\$\t\$HOSTNAME\$\\t\$HOSTEXECUTIONTIME\$\t\$HOSTOUTPUT\$\t\$HOSTPERFDATA\$**

Diese Option legt fest, welche (und wie) Daten in die **Host-Performance-Daten-Datei** geschrieben werden. Diese Vorlage kann **Makros**, Sonderzeichen (\t für Tabulator, \r für Wagenrücklauf (carriage return), \n für Zeilenvorschub (newline)) und reinen Text enthalten. Nach jedem Schreibvorgang wird ein Zeilenvorschub an die Performance-Daten-Datei angehängt.

Service-Performance-Daten-Dateivorlage (Service Performance Data File Template)

Format: **service_perfdata_file_template=<template>**

Beispiel: **service_perfdata_file_template=[SERVICEPERFDATA]\t\$TIMET\$\t\$HOSTNAME\$ \\t\$SERVICEDESC\$\t\$SERVICEEXECUTIONTIME\$\t\$SERVICELATENCY\$\t\$SERVICEOUTPUT\$\t\$SERVICEPERFDATA\$**

Diese Option legt fest, welche (und wie) Daten in die **Service-Performance-Daten-Datei** geschrieben werden. Diese Vorlage kann **Makros**, Sonderzeichen (\t für Tabulator, \r für Wagenrücklauf (carriage return), \n für Zeilenvorschub (newline)) und reinen Text enthalten. Nach jedem Schreibvorgang wird ein Zeilenvorschub an die Performance-Daten-Datei angehängt.

Host-Performance-Daten-Dateimodus (Host Performance Data File Mode)

Format: **host_perfdata_file_mode=<mode>**

Beispiel: **host_perfdata_file_mode=a**

Diese Option legt fest, wie die **Host-Performance-Datendatei** geöffnet wird. Solange die Datei keine "named pipe" ist, werden Sie diese wahrscheinlich im append-Modus (anhängen) öffnen wollen.

- a = Datei im append-Modus öffnen (Default)
- w = Datei im Write-Modus öffnen
- p = Datei im nicht-blockierenden Schreib-/Lesemodus öffnen (nützlich, wenn man in Pipes schreibt)

Service-Performance-Daten-Dateimodus (Service Performance Data File Mode)

Format: **service_perfdata_file_mode=<mode>**

Beispiel: **service_perfdata_file_mode=a**

Diese Option legt fest, wie die **Service-Performance-Datendatei** geöffnet wird. Solange die Datei keine "named pipe" ist, werden Sie diese wahrscheinlich im append-Modus (anhängen) öffnen wollen.

- a = Datei im append-Modus öffnen (Default)
- w = Datei im Write-Modus öffnen
- p = Datei im nicht-blockierenden Schreib-/Lesemodus öffnen (nützlich, wenn man in Pipes schreibt)

Host-Performance-Daten-Dateiverarbeitungsintervall (Host Performance Data File Processing Interval)

Format: **host_perfdata_file_processing_interval=<seconds>**

Beispiel: **host_perfdata_file_processing_interval=0**

Diese Option erlaubt es Ihnen, das Intervall (in Sekunden) anzugeben, in dem die [Host-Performance-Daten-Datei](#) mit dem [Host-Performance-Daten-Dateiverarbeitungsbefehl](#) verarbeitet wird. Ein Wert von 0 gibt an, dass die Performance-Daten-Datei nicht in regelmäßigen Intervallen verarbeiten werden soll.

Service-Performance-Daten-Dateiverarbeitungsintervall (Service Performance Data File Processing Interval)

Format: **service_perfdata_file_processing_interval=<seconds>**

Beispiel: **service_perfdata_file_processing_interval=0**

Diese Option erlaubt es Ihnen, das Intervall (in Sekunden) anzugeben, in dem die [Service-Performance-Daten-Datei](#) mit dem [Service-Performance-Daten-Dateiverarbeitungsbefehl](#) verarbeitet wird. Ein Wert von 0 gibt an, dass die Performance-Daten-Datei nicht in regelmäßigen Intervallen verarbeiten werden soll.

Host-Performance-Daten-Dateiverarbeitungsbefehl (Host Performance Data File Processing Command)

Format: **host_perfdata_file_processing_command=<command>**

Beispiel: **host_perfdata_file_processing_command=process-host-perfdata-file**

Diese Option erlaubt es Ihnen, den Befehl anzugeben, der ausgeführt werden soll, um die [Host-Performance-Daten-Datei](#) zu verarbeiten. Das *command*-Argument ist der Kurzname einer [command-Definition](#), die Sie in Ihrer Objektkonfigurationsdatei angeben. Das Intervall, in dem dieser Befehl ausgeführt wird, ist durch die [host_perfdata_file_processing_interval](#)-Direktive festgelegt.

Service-Performance-Daten-Dateiverarbeitungsbefehl (Service Performance Data File Processing Command)

Format: **service_perfdata_file_processing_command=<command>**

Beispiel: **service_perfdata_file_processing_command=process-service-perfdata-file**

Diese Option erlaubt es Ihnen, den Befehl anzugeben, der ausgeführt werden soll, um die [Service-Performance-Daten-Datei](#) zu verarbeiten. Das *command*-Argument ist der Kurzname einer [command-Definition](#), die Sie in Ihrer Objektkonfigurationsdatei angeben. Das Intervall, in dem dieser Befehl ausgeführt wird, ist durch die [service_perfdata_file_processing_interval](#)-Direktive festgelegt.

verwaiste Service-Prüfungsoption (Orphaned Service Check Option)

Format: **check_for_orphaned_services=<0/1>**

Beispiel: **check_for_orphaned_services=1**

Diese Option erlaubt es Ihnen, Prüfungen auf verwaiste Service-Prüfungen zu aktivieren oder zu deaktivieren. Verwaiste Service-Prüfungen sind Prüfungen, die ausgeführt und aus der Ereigniswarteschlange entfernt wurden, aber während langer Zeit keine Ergebnisse geliefert haben. Weil keine Ergebnisse für den Service zurückgeliefert wurden, wird er nicht erneut in der Ereigniswarteschlange eingeplant. Das kann dazu führen, dass Service-Prüfungen nicht mehr ausgeführt werden. Normalerweise passiert das sehr selten - es kann dann auftreten, wenn ein externer Benutzer oder Prozess den Prozess abgebrochen (killed) hat, der benutzt wurde, um eine Service-Prüfung auszuführen. Wenn diese Option aktiviert ist und Icinga feststellt, dass eine bestimmte Service-Prüfung kein Ergebnis geliefert hat, dann wird es einen Fehler protokollieren und die Service-Prüfung erneut einplanen. Wenn Sie feststellen, dass Service-Prüfungen anscheinend nie erneut eingeplant werden, dann aktivieren Sie diese Option und schauen Sie nach Protokollmeldungen zu verwaisten Services.

- 0 = nicht auf verwaiste Service-Prüfungen prüfen
- 1 = auf verwaiste Service-Prüfungen prüfen (Default)

verwaiste Host-Prüfungsoption (Orphaned Host Check Option)

Format: **check_for_orphaned_hosts=<0/1>**

Beispiel: **check_for_orphaned_hosts=1**

Diese Option erlaubt es Ihnen, Prüfungen auf verwaiste Host-Prüfungen zu aktivieren oder zu deaktivieren. Verwaiste Host-Prüfungen sind Prüfungen, die ausgeführt und aus der Ereigniswarteschlange entfernt wurden, aber während langer Zeit keine Ergebnisse geliefert haben. Weil keine Ergebnisse für den Host zurückgeliefert wurden, wird er nicht erneut in der Ereigniswarteschlange eingeplant. Das kann dazu führen, dass Host-Prüfungen nicht mehr ausgeführt werden. Normalerweise passiert das sehr selten - es kann dann auftreten, wenn ein externer Benutzer oder Prozess den Prozess abgebrochen (killed) hat, der benutzt wurde, um eine Host-Prüfung auszuführen. Wenn diese Option aktiviert ist und Icinga feststellt, dass eine bestimmte Host-Prüfung kein Ergebnis geliefert hat, dann wird es einen Fehler protokollieren und die Host-Prüfung erneut einplanen. Wenn Sie feststellen, dass Host-Prüfungen anscheinend nie erneut eingeplant werden, dann aktivieren Sie diese Option und schauen Sie nach Protokollmeldungen zu verwaisten Hosts.

- 0 = nicht auf verwaiste Host-Prüfungen prüfen
- 1 = auf verwaiste Host-Prüfungen prüfen (Default)

Service-Frischeprüfungsoption (Service Freshness Checking Option)

Format: **check_service_freshness=<0/1>**

Beispiel: **check_service_freshness=0**

Diese Option legt fest, ob Icinga regelmäßig die "Frische" (freshness) von Service-Prüfungen prüft. Das Aktivieren dieser Option ist nützlich, um sicherzustellen, dass **passive Service-Prüfungen** rechtzeitig empfangen werden. Mehr Informationen zur Frische-Prüfung finden Sie [hier](#).

- 0 = Service-Frische nicht prüfen
- 1 = Service-Frische prüfen (Default)

Service-Frische-Prüfintervall (Service Freshness Check Interval)

Format: **service_freshness_check_interval=<seconds>**

Beispiel: **service_freshness_check_interval=60**

Diese Einstellung legt fest, wie oft (in Sekunden) Icinga regelmäßig die "Frische" (freshness) von Service-Prüfergebnissen prüfen wird. Wenn Sie die Service-Frische-Prüfung (mit der `check_service_freshness`-Option) deaktiviert haben, dann hat diese Option keine Auswirkungen. Mehr Informationen zur Frische-Prüfung finden Sie [hier](#).

Host-Frischeprüfungsoption (Host Freshness Checking Option)

Format: **check_host_freshness=<0/1>**

Beispiel: **check_host_freshness=0**

Diese Option legt fest, ob Icinga regelmäßig die "Frische" (freshness) von Host-Prüfungen prüft. Das Aktivieren dieser Option ist nützlich, um sicherzustellen, dass [passive Host-Prüfungen](#) rechtzeitig empfangen werden. Mehr Informationen zur Frische-Prüfung finden Sie [hier](#).

- 0 = Host-Frische nicht prüfen (Default)
- 1 = Host-Frische prüfen

Host-Frische-Prüfintervall (Host Freshness Check Interval)

Format: **host_freshness_check_interval=<seconds>**

Beispiel: **host_freshness_check_interval=60**

Diese Einstellung legt fest, wie oft (in Sekunden) Icinga regelmäßig die "Frische" (freshness) von Host-Prüfergebnissen prüfen wird. Wenn Sie die Host-Frische-Prüfung (mit der `check_host_freshness`-Option) deaktiviert haben, dann hat diese Option keine Auswirkungen. Mehr Informationen zur Frische-Prüfung finden Sie [hier](#).

zusätzliche Frische-Latenzschwellwert-Option (Additional Freshness Threshold Latency Option)

Format: **additional_freshness_latency=<#>**

Beispiel: **additional_freshness_latency=15**

Diese Option legt die Anzahl von Sekunden fest, die Icinga zu jedem Host- oder Service-Frischeschwellwert hinzurechnet, den es automatisch berechnet (d.h., die nicht explizit durch den Benutzer angegeben wurden). Mehr Informationen zur Frische-Prüfung finden Sie [hier](#).

eingebetteter-Perl-Interpreter-Option (Embedded Perl Interpreter Option)

Format: **enable_embedded_perl=<0/1>**

Beispiel: **enable_embedded_perl=1**

Diese Einstellung legt fest, ob der eingebettete Perl-Interpreter auf programmweiter Basis aktiviert ist. Icinga muss mit Unterstützung für eingebettetes Perl kompiliert sein, damit diese Option eine Auswirkung hat. Mehr Informationen zum eingebauten Perl-Interpreter finden Sie [hier](#).



Anmerkung

Ab Icinga 1.5 hat sich der Default geändert. Vorher war diese Direktive aktiviert, nun ist sie *deaktiviert*.

Option implizite Nutzung des eingebetteten Perl-Interpreters (Embedded Perl Implicit Use Option)

Format: **use_embedded_perl_implicitly=<0/1>**

Beispiel: **use_embedded_perl_implicitly=1**

Diese Einstellung legt fest, ob der eingebettete Perl-Interpreter für Perl-Plugins/Scripte benutzt werden soll, die ihn nicht explizit aktiviert/deaktiviert haben. Icinga muss mit Unterstützung für eingebettetes Perl kompiliert sein, damit diese Option eine Auswirkung hat. Mehr Informationen zum eingebauten Perl-Interpreter finden Sie [hier](#).

Datumformat (Date Format)

Format: **date_format=<option>**

Beispiel: **date_format=us**

Diese Option erlaubt es Ihnen anzugeben, welche Art von Datums-/Zeitformat Icinga im Web-Interface und in den Datums-/Zeit-Makros benutzen soll. Mögliche Optionen (zusammen mit einer Beispieldarstellung) umfassen u.a.:

Option	Ausgabeformat	Beispieldarstellung
us	MM-DD-YYYY HH:MM:SS	06-30-2002 03:15:00
euro	DD-MM-YYYY HH:MM:SS	30-06-2002 03:15:00
iso8601	YYYY-MM-DD HH:MM:SS	2002-06-30 03:15:00
strict-iso8601	YYYY-MM-DDTHH:MM:SS	2002-06-30T03:15:00

Zeitzonen-Option (Timezone Option)

Format: **use_timezone=<tz>**

Beispiel: **use_timezone=US/Mountain**

Diese Option erlaubt es Ihnen, die Standard-Zeitzone zu überschreiben, in der die Icinga-Instanz läuft. Das ist nützlich, wenn Sie mehrere Icinga-Instanzen haben, die auf dem gleichen Server laufen, aber mit verschiedenen lokalen Zeiten verbunden sind. Wenn nichts angegeben ist, wird Icinga die Zeitzone des Rechners benutzen.



Anmerkung: wenn Sie diese Option nutzen, um eine eigene Zeitzone anzugeben, müssen Sie auch die Apache-Konfigurationsdirektiven für die CGIs auf die Zeitzone ändern, die Sie haben möchten. Beispiel:

```
<Directory "/usr/local/icinga/sbin/">
SetEnv TZ "US/Mountain"
...
</Directory>
```

Illegal Zeichen für Objektnamen (Illegal Object Name Characters)

Format: **illegal_object_name_chars=<chars...>**

Beispiel: **illegal_object_name_chars='~!\$%^&*!"|'<>?,()=**

Diese Option erlaubt es Ihnen, illegale Zeichen anzugeben, die nicht in Host-Namen, Service-Beschreibungen oder Namen anderer Objektarten benutzt werden können. Icinga gestattet Ihnen, die meisten Zeichen in Objektdefinitionen zu benutzen, aber wir raten Ihnen, die im Beispiel gezeigten Zeichen nicht zu nutzen. Wenn Sie es dennoch tun, können Sie Probleme im Web-Interface, in Benachrichtigungsbefehlen usw. bekommen.

illegal Zeichen für Makroausgaben (Illegal Macro Output Characters)

Format: **illegal_macro_output_chars=<chars...>**

Beispiel: **illegal_macro_output_chars='~\$^&"|'<>**

Diese Option erlaubt es Ihnen, illegale Zeichen anzugeben, die aus [Makros](#) entfernt werden, bevor diese in Benachrichtigungen, Eventhandlern und anderen Befehlen benutzt werden. Dies betrifft AUCH Makros, die in Service- oder Host-Prüfbefehlen benutzt werden. Sie können sich entscheiden, die Zeichen im Beispiel nicht zu entfernen, aber wir raten Ihnen, das nicht zu tun. Einige dieser Zeichen werden von der Shell interpretiert (z.B. der "Backtick") und können zu Sicherheitsproblemen führen. Die folgenden Makros werden von den Zeichen gereinigt, die Sie angeben:

**\$HOSTOUTPUT\$, \$HOSTPERFDATA\$, \$HOSTACKAUTHOR\$, \$HOSTACKCOMMENT\$,
\$SERVICEOUTPUT\$, \$SERVICEPERFDATA\$, \$SERVICEACKAUTHOR\$, und
\$SERVICEACKCOMMENT\$**

Option Anpassung regulärer Ausdrücke (Regular Expression Matching Option)

Format: **use_regex_matching=<0/1>**

Beispiel: **use_regex_matching=0**

Diese Option legt fest, ob verschiedene Direktiven in Ihren [Objektdefinitionen](#) als reguläre Ausdrücke verarbeitet werden. Mehr Informationen dazu, wie das funktioniert, finden Sie [hier](#).

- 0 = keine angepassten regulären Ausdrücke benutzen (Default)
- 1 = angepasste reguläre Ausdrücke benutzen

Option wahre reguläre Ausdrucksanpassung (True Regular Expression Matching Option)

Format: **use_true_regex_matching=<0/1>**

Beispiel: **use_true_regex_matching=0**

Wenn Sie reguläre Ausdrücke von verschiedenen Objektdirektiven mit der [use_regex_matching](#)-Option aktiviert haben, dann wird diese Option festlegen, wann Objektdirektiven als reguläre Ausdrücke behandelt werden. Wenn diese Option deaktiviert ist (der Standard), dann werden Direktiven nur dann als reguläre Ausdrücke behandelt, wenn sie *, ?, + oder \. enthalten. Wenn diese Option aktiviert ist, werden alle entsprechenden Direktiven als reguläre Ausdrücke behandelt - seien Sie vorsichtig bei der Aktivierung! Mehr Informationen darüber, wie das funktioniert, finden Sie [hier](#).

- 0 = keine Anpassung wahrer regulärer Ausdrücke benutzen (Default)
- 1 = Anpassung wahrer regulärer Ausdrücke benutzen

Administrator-e-Mail-Adresse (Administrator Email Address)

Format: **admin_email=<email_address>**

Beispiel: **admin_email=root@localhost.localdomain**

Dies ist die e-Mail-Adresse des Administrators der lokalen Maschine (d.h. die, auf der Icinga läuft). Dieser Wert kann mit Hilfe des [\\$ADMINEMAIL\\$-Makros](#) in Benachrichtigungsbefehlen benutzt werden.

Administrator-Pager (Administrator Pager)

Format: **admin_pager=<pager_number_or_pager_email_gateway>**

Beispiel: **admin_pager=pageroot@localhost.localdomain**

Dies ist die Pager-Nummer (oder die des Pager-e-Mail-Gateways) des Administrators der lokalen Maschine (d.h. die, auf der Icinga läuft). Die Pager-Nummer/Adresse kann mit Hilfe des [\\$ADMINPAGER\\$-Makros](#) in Benachrichtigungsbefehlen benutzt werden.

Ereignisvermittler-Optionen (Event Broker Options)

Format: **event_broker_options=<#>**

Beispiel: **event_broker_options=-1**

Diese Option kontrolliert, welche Daten an den Ereignisvermittler gesandt werden und damit an jedes geladene Ereignisvermittler-Modul. Dies ist ein fortgeschrittenes Feature. Falls Sie im Zweifel sind, vermitteln Sie entweder gar nichts (wenn Sie keine Ereignisvermittler-Module benutzen) oder alles (wenn Sie Ereignisvermittler-Module benutzen). Mögliche Werte sehen Sie

nachfolgend.

- 0 = nichts vermitteln
- -1 = alles vermitteln
- # = sehen Sie sich die BROKER_*-Definitionen im Quellcode an (include/broker.h), um andere Werte zu ermitteln

Ereignisvermittler-Module (Event Broker Modules)

Format: **broker_module=<modulepath> [moduleargs]**

Beispiel: **broker_module=/usr/local/icinga/bin/ndomod.o
cfg_file=/usr/local/icinga/etc/ndomod.cfg**

Diese Option wird benutzt, um ein Ereignisvermittler-Modul anzugeben, das beim Icinga-Start geladen werden soll. Benutzen Sie mehrere Direktiven, wenn Sie mehrere Module laden wollen. An das Modul zu übergebende Argumente werden durch ein Leerzeichen vom Pfad des Moduls getrennt.

!!! WARNUNG !!!

Überschreiben Sie KEINE Module, während sie von Icinga genutzt werden, oder Icinga wird mit einem SEGFAULT-Feuerwerk abstürzen. Dies ist ein Fehler/eine Begrenzung entweder in dlopen(), dem Kernel, und/oder dem Filesystem. Und vielleicht Icinga...

Der korrekte/sichere Weg, ein Modul zu aktualisieren, ist eine der folgenden Methoden:

1. stoppen Sie Icinga, ersetzen Sie das Modul und starten Sie Icinga erneut
2. während Icinga läuft... löschen Sie die originale Moduldatei, schieben Sie die neue Moduldatei an den richtigen Platz und starten Sie Icinga erneut

Debug-Datei (Debug File)

Format: **debug_file=<file_name>**

Beispiel: **debug_file=/usr/local/icinga/var/nagios.debug**

Diese Option legt fest, wohin Icinga Debugging-Informationen schreiben soll. Welche Informationen (falls überhaupt) geschrieben werden, wird durch die **debug_level**- und **debug_verbose**-Optionen festgelegt. Sie können die Debug-Datei automatisch rotieren lassen, wenn sie eine bestimmte Größe überschreitet, die Sie über die **max_debug_file_size**-Option festlegen können.

Debug-Stufe (Debug Level)

Format: **debug_level=<#>**

Beispiel: **debug_level=24**

Diese Option legt fest, welche Arten von Informationen Icinga in das **debug_file** schreiben soll. Dieser Wert ist ein logisches ODER der nachfolgenden Werte:

- -1 = alles protokollieren
- 0 = nichts protokollieren (Default)
- 1 = Informationen zu Funktionsbeginn/Ende
- 2 = Konfigurationsinformationen
- 4 = Prozessinformationen
- 8 = geplante Ereignisinformationen
- 16 = Host-/Service-Prüfinformationen
- 32 = Benachrichtigungsinformationen
- 64 = Ereignisvermittlerinformationen

Debug-Ausführlichkeit (Debug Verbosity)

Format: **debug_verbosity=<#>**

Beispiel: **debug_verbosity=1**

Diese Option legt fest, wie viel Debugging-Informationen Icinga in das [debug_file](#) schreiben soll.

- 0 = grundlegende Informationen
- 1 = detailliertere Informationen (Default)
- 2 = sehr detaillierte Informationen

maximale Debug-Dateigröße (Maximum Debug File Size)

Format: **max_debug_file_size=<#>**

Beispiel: **max_debug_file_size=1000000**

Diese Option legt die maximale Größe (in Bytes) der [Debug-Datei](#) fest. Wenn die Datei die Größe überschreitet, dann wird ".old" als Erweiterung angehängt. Wenn bereits eine Datei mit der ".old"-Erweiterung existiert, wird diese gelöscht. Das stellt sicher, dass Ihr Plattenplatz nicht außer Kontrolle gerät, wenn Sie Icinga debuggen.

Leere Hostgruppenzuordnung erlauben

Format: **allow_empty_hostgroup_assignment=<0|1>**

Beispiel: **allow_empty_hostgroup_assignment=1**

Diese boolesche Option legt fest, ob Services, die leeren Hostgruppen zuordnet sind (Hostgruppen ohne Host-Member), Icinga beim Start (oder bei der Konfigurationsprüfung) mit einem Fehler abbrechen lassen oder nicht. Das Default-Verhalten, wenn die Option nicht in der Hauptkonfigurationsdatei vorhanden ist (oder auf "0" gesetzt ist), besteht darin, dass Icinga mit einem Fehler abbricht, wenn Services mit Hostgruppen verbunden sind, denen in der hostgroup-Definition keine Host-Member zugeordnet sind.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.3.

Leere Performance-Ergebnisse verarbeiten

Format:

```
host_perfdata_process_empty_results=<0|1>
service_perfdata_process_empty_results=<0|1>
```

Beispiel:

```
host_perfdata_process_empty_results=1
service_perfdata_process_empty_results=1
```

Diese Optionen legen fest, ob der Core leere Performance-Daten-Ergebnisse verarbeiten soll oder nicht. Dies wird für verteilte Überwachung benötigt und ist per Default aktiviert. Wenn Sie keine leeren Performance-Daten benötigen - und ein paar CPU-Zyklen bei der Berechnung von Makros sparen wollen - dann können Sie diese Optionen abschalten. Seien Sie vorsichtig! Werte: 1 = aktiviert, 0 = deaktiviert.



Anmerkung

Diese Optionen sind verfügbar ab Icinga 1.4

[Zurück](#)[Nach oben](#)[Weiter](#)[Konfigurationsüberblick](#)[Zum Anfang](#)[Überblick Objektkonfiguration](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Überblick Objektkonfiguration

[Zurück](#)

[Kapitel 3. Icinga konfigurieren](#)

[Weiter](#)

Überblick Objektkonfiguration

Was sind Objekte?

Objekte sind alle Elemente, die an der Überwachungs- und Benachrichtigungslogik beteiligt sind. Objekttypen umfassen:

- Services
- Servicegruppen
- Hosts
- Hostgruppen
- Kontakte
- Kontaktgruppen
- Befehle
- Zeitfenster
- Benachrichtigungseskalationen
- Benachrichtigungs- und Ausführungsabhängigkeiten

Mehr Informationen darüber, was Objekte sind und wie sie in Beziehung zueinander stehen, finden Sie nachstehend.

Wo werden Objekte definiert?

Objekte können in einer oder mehreren Konfigurationsdateien und/oder Verzeichnissen definiert werden, die Sie mit den `cfg_file`- und/oder `cfg_dir`-Direktiven in der Hauptkonfigurationsdatei angeben.

include_file / include_dir

Eine Objektdefinitionsdatei kann andere Objektdefinitionsdateien einschließen mit Hilfe der `include_file=<file_name>`- und `include_dir=<directory_name>`-Direktiven. Die erste schließt nur die einzelne angegebene Datei ein, die zweite wird im angegebenen Verzeichnis alle Dateien mit der Endung `.cfg` einschließen. Diese Direktiven können mehrfach angegeben werden, um mehrere Dateien und/oder Verzeichnisse einzuschließen.

Die Direktiven sind nicht in den eigentlichen Definitionen eines Objektes erlaubt, sondern sollten vor, nach oder zwischen Objektdefinitionen auftreten. Sie sind eng mit den `cfg_file=`- und `cfg_dir=`-Direktiven in der Hauptkonfigurationsdatei verwandt.

Diese Direktiven können verkettet werden, d.h. eine Objektdefinitionsdatei, die in der Hauptkonfigurationsdatei durch eine `cfg_file=`- oder `cfg_dir=`-Direktive eingeschlossen wird, kann `include_file=` oder `include_dir=` enthalten, um eine weitere Objektdefinitionsdatei einzuschließen, die ebenfalls `include_file=` oder `include_dir=` enthält, um eine weitere Datei einzuschließen, und so fort.



Hinweis: Wenn Sie der [Schnellstart-Installationsanleitung](#) folgen, werden verschiedene Beispiel-Objektkonfigurationsdateien in `/usr/local/icinga/etc/objects/` abgelegt. Sie können diese Beispieldateien benutzen, um zu sehen, wie Objektvererbung funktioniert und lernen, wie Sie Ihre eigenen Objektdefinitionen anlegen.

Wie werden Objekte definiert?

Objekte werden in einem flexiblen Vorlagenformat definiert, das es viel einfacher machen kann, Ihre Icinga-Konfiguration auf lange Sicht zu verwalten. Grundlegende Informationen, wie Objekte in Ihren Konfigurationsdateien definiert werden, finden Sie [hier](#).

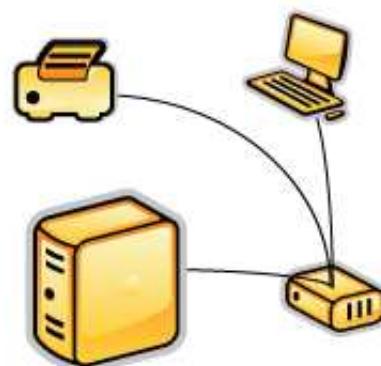
Sobald Sie mit den Grundlagen vertraut sind, wie Objekte zu definieren sind, sollten Sie bei [Objektvererbung](#) weiterlesen, weil es Ihre Konfiguration robuster für die Zukunft macht. Erfahrene Benutzer können einige fortgeschrittene Möglichkeiten der Objektdefinition ausnutzen, die in der Dokumentation zu [Objekt-Tricks](#) beschrieben sind.

Objekte erklärt

Einige der Hauptobjekttypen werden nachfolgend genauer erklärt...

Hosts sind eins der zentralen Objekte in der Überwachungslogik. Wichtige Attribute von Hosts sind:

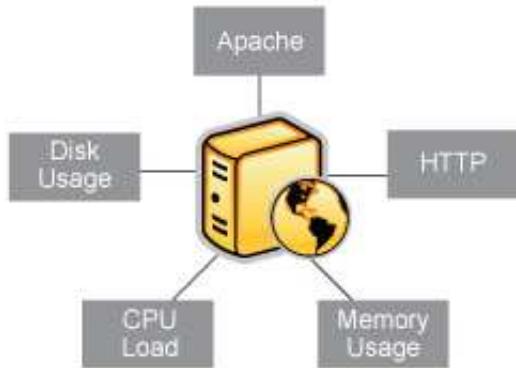
- Hosts sind normaler Weise physikalische Geräte in Ihrem Netzwerk (Server, Workstations, Router, Switches, Drucker usw.).
- Hosts haben eine Adresse irgendeiner Art (z.B. eine IP- oder MAC-Adresse).
- Hosts haben einen oder mehrere Services, die mit ihm verbunden sind.
- Hosts können Eltern/Kind-Beziehungen mit anderen Hosts haben, oftmals dargestellt durch reale Netzwerkverbindungen, die in der [Netzwerk-Erreichbarkeits-Logik](#) benutzt wird.



Hostgruppen sind Gruppen von einem oder mehreren Hosts. Hostgruppen können es einfacher machen, (1) den Status von in Beziehung stehenden Hosts im Icinga-Web-Interface anzusehen und (2) Ihre Konfiguration mit Hilfe von [Objekt-Tricks](#) zu vereinfachen.

Services sind eins der zentralen Objekte in der Überwachungslogik. Services sind mit Hosts verbunden und können:

- Attribute eines Hosts sein (CPU-Auslastung, Plattenbelegung, Laufzeit, usw.)
- Services sein, die durch den Host zur Verfügung gestellt werden (HTTP, POP3, FTP, SSH, usw.)
- andere Dinge sein, die mit dem Host verbunden sind (DNS records, usw.)



Servicegruppen sind Gruppen von einem oder mehreren Services. Servicegruppen können es einfacher machen, (1) den Status von in Beziehung stehenden Services im Icinga-Web-Interface anzusehen und (2) Ihre Konfiguration mit Hilfe von [Objekt-Tricks](#) zu vereinfachen.

Kontakte sind Leute, die am Benachrichtigungsprozess beteiligt sind:

- Kontakte haben eine oder mehrere Benachrichtigungsmethoden (Handy, Pager, e-Mail, Sofortnachrichten, usw.)
- Kontakte erhalten Benachrichtigungen zu Hosts und Services, für die sie verantwortlich sind



Kontaktgruppen sind Gruppen von einem oder mehreren Kontakten. Kontaktgruppen können es einfacher machen, alle Leute zu definieren, die informiert werden, wenn bestimmte Host- oder Serviceprobleme auftreten.

Zeitfenster werden benutzt, um zu kontrollieren:

- wann Hosts und Services überwacht werden
- wann Kontakte Benachrichtigungen erhalten



Informationen darüber, wie Zeitfenster arbeiten, finden Sie [hier](#).

Befehle werden benutzt, um Icinga mitzuteilen, welche Programme, Scripte usw. es ausführen soll:



- Host- und Service-Prüfungen
- Benachrichtigungen
- Eventhandler
- und mehr...

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Optionen der
Hauptkonfigurationsdatei

[Zum Anfang](#)

Objektdefinitionen

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Objektdefinitionen

[Zurück](#)
[Kapitel 3. Icinga konfigurieren](#)
[Weiter](#)

Objektdefinitionen

Einführung

Eine der Möglichkeiten des Objektkonfigurationsformats von Icinga besteht darin, dass Sie Objektkonfigurationsdefinitionen erstellen können, die Eigenschaften von anderen Objektdefinitionen erben. Eine Erklärung, wie Objektvererbung funktioniert, finden Sie [hier](#). Wir empfehlen Ihnen dringend, dass Sie sich mit Objektvererbung beschäftigen, nachdem Sie die folgende Dokumentation überflogen haben, weil sie Ihnen die Arbeit bei der Erstellung und Wartung der Objektdefinitionen viel einfacher macht. Lesen Sie außerdem die [Objekt-Tricks](#), die Ihnen Abkürzungsmöglichkeiten für andernfalls langwierige Konfigurationsaufgaben bieten.

 Wenn Sie Konfigurationsdateien anlegen und/oder editieren, dann behalten Sie das Folgende im Hinterkopf:

1. Zeilen, die mit einem '#' -Zeichen beginnen, werden als Kommentare angesehen und nicht verarbeitet
2. Direktivennamen sind "case-sensitive", die Beachtung von Groß- und Kleinschreibung ist wichtig!
3. Zeichen nach einem Semikolon (;) in Konfigurationszeilen werden als Kommentar angesehen und deshalb nicht verarbeitet



Anmerkung

Bitte vermeiden Sie die Verwendung von Sonderzeichen in den Objektdefinitionen. Im Moment (Icinga <= 1.5.0) gibt es einen Fehler, der die Ausführung von "forced reschedules" im Classic UI verhindert, wobei als Meldung ein "not authorized" angezeigt wird.

Anmerkungen zur Aufbewahrung (Retention Notes)

Es ist wichtig anzumerken, dass einige Direktiven in Host-, Service- und Kontaktdefinitionen möglicherweise keine Wirkung zeigen, wenn Sie diese in den Konfigurationsdateien ändern. Objektdirektiven, die dieses Verhalten zeigen, sind mit einem Stern gekennzeichnet (*). Der Grund für dieses Verhalten liegt darin, dass Icinga Werte in der [Statusaufbewahrungsdatei](#) denen aus den Konfigurationsdateien vorzieht, wenn Sie [Statusaufbewahrung](#) auf programmweiter Basis aktiviert haben *und* den Wert der Direktive während der Laufzeit mit einem [externen Befehl](#) geändert haben.

Ein Weg, um dieses Problem zu umgehen, ist das Deaktivieren der Aufbewahrung von nicht-Statusinformationen mit Hilfe der `retain_nonstatus_information`-Direktive in den Host-, Service- und Kontaktdefinitionen. Durch das Deaktivieren dieser Direktive wird Icinga dazu veranlasst, beim (Neu-)Start die initialen Werte für diese Direktiven aus Ihren Konfigurationsdateien zu nehmen, statt die aus der Statusaufbewahrungsdatei zu verwenden.

Beispielkonfigurationsdatei

 Anmerkung: Beispielobjektkonfigurationsdateien werden im `/usr/local/icinga/etc/-Verzeichnis` installiert, wenn Sie der [Schnellstart-Anleitung](#) gefolgt sind.

Objekttypen

[Host-Definitionen](#)

[Hostgruppen-Definitionen](#)

[Service-Definitionen](#)

[Servicegruppen-Definitionen](#)

[Kontakt-Definitionen](#)

[Kontaktgruppen-Definitionen](#)

[Zeitfenster-Definitionen](#)

[Befehlsdefinitionen](#)

[Service-Abhängigkeitsdefinitionen](#)

[Service-Eskalationsdefinitionen](#)

[Host-Abhängigkeitsdefinitionen](#)

[Host-Eskalationsdefinitionen](#)

[erweiterte Host-Informationsdefinitionen](#)

[erweiterte Service-Informationsdefinitionen](#)

Host-Definition

Beschreibung:

Eine Host-Definition wird benutzt, um einen Server, eine Workstation, ein Gerät usw. zu definieren, die sich in Ihrem Netzwerk befinden.

Definition:

Anmerkung: "Direktiven" werden benötigt, die anderen sind optional.

```
define host{
    host_name          host_name
    alias              alias
```

display_name	display_name
address	address
address6	address6
parents	host_names
hostgroups	hostgroup_names
check_command	command_name
initial_state	[o,d,u]
max_check_attempts	#
check_interval	#
retry_interval	#
active_checks_enabled	[0/1]
passive_checks_enabled	[0/1]
check_period	timeperiod_name
obsess_over_host	[0/1]
check_freshness	[0/1]
freshness_threshold	#
event_handler	command_name
event_handler_enabled	[0/1]
low_flap_threshold	#
high_flap_threshold	#
flap_detection_enabled	[0/1]
flap_detection_options	[o,d,u]
process_perf_data	[0/1]
retain_status_information	[0/1]
retain_nonstatus_information	[0/1]
contacts	contacts
contact_groups	contact_groups
notification_interval	#
first_notification_delay	#
notification_period	timeperiod_name
notification_options	[d,u,r,f,s]
notifications_enabled	[0/1]
stalking_options	[o,d,u]

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
}	

Beispieldefinition:

```
define host{
    host_name                bogus-router
    alias                     Bogus Router #1
    address                   192.168.1.254
    parents                   server-backbone
    check_command              check-host-alive
    check_interval             5
    retry_interval            1
    max_check_attempts         5
    check_period               24x7
    process_perf_data          0
    retain_nonstatus_information 0
    contact_groups             router-admins
    notification_interval      30
    notification_period        24x7
    notification_options       d,u,r
}
```

*Beschreibung der Direktiven:***host_name:**

Diese Direktive wird benutzt, um einen Kurznamen zu definieren, der den Host identifiziert. Er wird in Hostgruppen- und Service-Definitionen benutzt, um auf diesen bestimmten Host zu verweisen. Hosts können mehrere Services haben (die überwacht werden), die mit ihm verbunden sind.

alias:

Diese Direktive wird benutzt, um einen längeren Namen oder eine Beschreibung zu definieren, der/die den Host identifiziert. Er/sie wird angeboten, damit Sie den Host einfacher identifizieren können. Bei korrekter Anwendung wird das \$HOSTALIAS\$-Makro diesen Alias/diese Beschreibung enthalten.

address:

Diese Direktive wird benutzt, um die Adresse des Hosts zu definieren. Normalerweise ist dies die IP-Adresse des Hosts, obwohl es eigentlich alles sein kann, was Sie wollen (solange es genutzt werden kann, um den Status des Hosts zu prüfen). Sie können einen vollqualifizierten Domänennamen (FQDN) statt einer IP-Adresse benutzen, um den Host zu identifizieren, aber wenn keine DNS-Dienste verfügbar sind, kann dies zu Problemen führen. Bei korrekter Anwendung wird das \$HOSTADDRESS\$-[Makro](#) diese Adresse enthalten. **Anmerkung:** Wenn Sie keine Adress-Direktive in einer Host-Definition benutzen, wird der Name des Hosts statt der Adresse benutzt. Trotzdem ein Wort der Warnung: wenn DNS ausfällt, werden die meisten Ihrer Service-Prüfungen fehlschlagen, weil die Plugins nicht in der Lage sind, den Host-Namen aufzulösen.

address6:

Diese Direktive wird benutzt, um eine zweite Adresse des Hosts zu definieren. Normalerweise ist dies die IPv6-Adresse des Hosts, obwohl es eigentlich alles sein kann, was Sie wollen (solange es genutzt werden kann, um den Status des Hosts zu prüfen). Sie können einen vollqualifizierten Domänennamen (FQDN) statt einer IP-Adresse benutzen, um den Host zu identifizieren, aber wenn keine DNS-Dienste verfügbar sind, kann dies zu Problemen führen. Bei korrekter Anwendung wird das \$HOSTADDRESS6\$-[Makro](#) diese Adresse enthalten. **Anmerkung:** Wenn Sie keine Address6-Direktive in einer Host-Definition benutzen, wird der Name des Hosts statt der Adresse benutzt. Trotzdem ein Wort der Warnung: wenn DNS ausfällt, werden die meisten Ihrer Service-Prüfungen fehlschlagen, weil die Plugins nicht in der Lage sind, den Host-Namen aufzulösen (verfügbar ab Icinga 1.3).

display_name:

Diese Direktive wird benutzt, um einen alternativen Namen zu definieren, der im Web-Interface für den Host angezeigt wird. Wenn nicht angegeben, wird statt dessen der Wert der `host_name`-Direktive benutzt. Anmerkung: Die CGIs bis einschließlich Icinga 1.0.1 nutzen diese Option nicht.

parents:

Diese Direktive wird benutzt, um eine Komma-separierte Liste von Kurznamen der "Eltern"-Hosts dieses bestimmten Hosts zu definieren. Eltern-Hosts sind typischerweise Router, Switches, Firewalls usw. Ein Router, Switch usw., der am nächsten zum entfernten Host ist, wird als "Eltern" dieses Hosts angesehen. Lesen Sie weitere Informationen im Dokument "Festlegen des Zustands und der Erreichbarkeit von Netzwerk-Hosts", das Sie [hier](#) finden. Wenn dieser Host im gleichen Netzwerksegment wie der überwachende Host ist (ohne dazwischen liegende Router usw.), wird der Host als im lokalen Netzwerk befindlich angesehen und hat deshalb keinen Eltern-Host. Lassen Sie diesen Wert leer, wenn der Host keinen Eltern-Host hat (d.h. wenn er im gleichen Segment wie der Icinga-Host ist). Die Reihenfolge, in der Sie Eltern-Hosts angeben, hat keinen Einfluss darauf, wie Dinge überwacht werden.

hostgroups:

Diese Direktive wird benutzt, um den/die Kurznamen der [Hostgruppe\(n\)](#) anzugeben, zu dem/denen der Host gehört. Mehrere Hostgruppen werden durch Kommata von einander getrennt. Diese Direktive kann als Alternative (oder zusätzlich) zur `members`-Direktive in den [hostgroup](#)-Definitionen genutzt werden.

check_command:	Diese Direktive wird benutzt, um den <i>Kurznamen des Befehls</i> anzugeben, mit dem geprüft wird, ob der Host funktioniert oder nicht. Typischerweise wird dieser Befehl versuchen, den Host per "ping" zu prüfen, ob er "lebt". Der Befehl muss den Status OK (0) zurückliefern, denn sonst wird Icinga annehmen, dass der Host "down" ist. Wenn Sie diesen Wert leer lassen, wird der Host <i>nicht</i> aktiv geprüft. Dadurch wird Icinga höchstwahrscheinlich annehmen, dass der Host "up" ist (und ihn ggf. als "PENDING" im Web-Interface anzeigen). Das ist nützlich, wenn Sie Drucker oder andere Geräte überwachen, die regelmäßig ausgeschaltet werden. Die maximale Zeit, die der Prüfbefehl laufen darf, wird durch die host_check_timeout -Option kontrolliert.
initial_state:	Als Default nimmt Icinga an, dass sich alle Hosts im UP-Zustand befinden, wenn es startet. Sie können mit dieser Direktive den initialen Zustand eines Hosts übersteuern. Gültige Optionen sind: o = UP, d = DOWN und u = UNREACHABLE.
max_check_attempts:	Diese Direktive wird benutzt, um zu definieren, wie oft Icinga den Host-Prüfbefehl wiederholt, wenn er einen anderen als einen OK-Zustand zurückliefert. Bei einem Wert von 1 wird Icinga einen Alarm generieren, ohne den Host erneut zu prüfen. Anmerkung: wenn Sie den Zustand des Hosts nicht prüfen wollen, müssen Sie den Wert trotzdem mindestens auf 1 setzen. Lassen Sie die <i>check_command</i> -Option leer, um die Host-Prüfung zu umgehen.
check_interval:	Diese Direktive wird benutzt, um die Anzahl von "Zeiteinheiten" zwischen regelmäßig geplanten Prüfungen zu definieren. Solange Sie die interval_length -Direktive mit einem Default-Wert von 60 nicht verändert haben, wird diese Zahl Minuten bedeuten. Mehr Informationen zu diesem Wert finden Sie in der check scheduling -Dokumentation.
retry_interval:	Diese Direktive wird benutzt, um die Anzahl von "Zeiteinheiten" zu definieren, die zwischen erneuten Überprüfungen gewartet werden sollen. Erneute Überprüfungen für den Host werden mit dem Wiederholungsintervall eingeplant, wenn dieser in einen nicht-UP-Zustand gewechselt ist. Sobald der Host max_check_attempts -Mal ohne eine Zustandsänderung geprüft wurde, wird die Planung zum "normalen" Wert zurückkehren, der durch den check_interval -Wert angegeben wird. Solange Sie die interval_length -Direktive mit einem Default-Wert von 60 nicht verändert haben, wird diese Zahl Minuten bedeuten. Mehr Informationen zu diesem Wert finden Sie in der check scheduling -Dokumentation.
active_checks_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob aktive Prüfungen (entweder regelmäßig geplant oder nach Bedarf) für diesen Host aktiviert sind oder nicht. Werte: 0 = keine aktiven Host-Prüfungen, 1 = aktive Host-Prüfungen.
passive_checks_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob passive Prüfungen für diesen Host aktiviert sind oder nicht. Werte: 0 = passive Host-Prüfungen deaktivieren, 1 = passive Host-Prüfungen aktivieren
check_period:	Diese Direktive wird benutzt, um den Kurznamen des Zeitfensters anzugeben, in dem aktive Prüfungen für diesen Host ausgeführt werden.
obsess_over_host *:	Diese Direktive legt fest, ob Prüfungen für den Host über ochp_command "verfolgt" werden sollen.
check_freshness *:	Diese Direktive wird benutzt, um festzulegen, ob Frische-Prüfungen (freshness checks) für diesen Host aktiviert sind oder nicht. Werte: 0 = Frische-Prüfungen deaktivieren, 1 = Frische-Prüfungen aktivieren.

freshness_threshold:	Diese Direktive wird benutzt, um den Frische-Schwellwert (freshness threshold) (in Sekunden) für diesen Host festzulegen. Wenn Sie einen Wert von Null für diese Direktive setzen, wird Icinga automatisch einen Frische-Schwellwert festlegen.
event_handler:	Diese Direktive wird benutzt, um den <i>Kurznamen des Befehls</i> anzugeben, der jedes Mal ausgeführt werden soll, sobald ein Statuswechsel für den Host erkannt wird (d.h. er "down" geht oder sich wieder erholt). Lesen Sie die Dokumentation zu Eventhandler für eine detailliertere Erklärung, wie Scripte zur Behandlung von Ereignissen geschrieben werden. Die maximale Zeit, die ein Eventhandler-Befehl dauern darf, wird durch die event_handler_timeout -Option kontrolliert.
event_handler_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob der Eventhandler für diesen Host aktiviert ist oder nicht. Werte: 0 = Host-Eventhandler deaktivieren, 1 = Host-Eventhandler aktivieren
low_flap_threshold:	Diese Direktive wird benutzt, um den unteren Zustandsänderungsschwellwert zu definieren, der in der Flattererkennung für diesen Host benutzt wird. Mehr Informationen zur Flattererkennung finden Sie hier . Wenn Sie diese Direktive auf 0 setzen, wird der programmweite Wert aus der low_host_flap_threshold -Direktive benutzt.
high_flap_threshold:	Diese Direktive wird benutzt, um den oberen Zustandsänderungsschwellwert zu definieren, der in der Flattererkennung für diesen Host benutzt wird. Mehr Informationen zur Flattererkennung finden Sie hier . Wenn Sie diese Direktive auf 0 setzen, wird der programmweite Wert aus der high_host_flap_threshold -Direktive benutzt.
flap_detection_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob Flattererkennung für diesen Host aktiviert ist. Mehr Informationen zur Flattererkennung finden Sie hier . Werte: 0 = Host-Flattererkennung deaktivieren, 1 = Host-Flattererkennung aktivieren.
flap_detection_options:	Diese Direktive wird benutzt, um festzulegen, welche Host-Zustände die Flattererkennungslogik für diesen Host benutzen wird. Gültige Optionen sind Kombinationen von einem oder mehreren folgender Werten: o = UP-Zustände, d = DOWN-Zustände, u = UNREACHABLE-Zustände.
process_perf_data *:	Diese Direktive wird benutzt, um festzulegen, ob die Verarbeitung von Performance-Daten für diesen Host aktiviert ist. Werte: 0 = Verarbeitung von Performance-Daten deaktiviert, 1 = Verarbeitung von Performance-Daten aktiviert.
retain_status_information:	Diese Direktive wird benutzt, um festzulegen, ob zustandsbezogene Informationen zu diesem Host über Programmneustarts hinweg aufbewahrt wird. Das ist nur sinnvoll, wenn Sie Statusaufbewahrung über die retain_state_information -Direktive aktiviert haben. Werte: 0 = Aufbewahrung von Statusinformationen deaktivieren, 1 = Aufbewahrung von Statusinformationen aktivieren.
retain_nonstatus_information:	Diese Direktive wird benutzt, um festzulegen, ob nicht-zustandsbezogene Informationen zu diesem Host über Programmneustarts hinweg aufbewahrt wird. Das ist nur sinnvoll, wenn Sie Statusaufbewahrung über die retain_state_information -Direktive aktiviert haben. Werte: 0 = Aufbewahrung von nicht-Statusinformationen deaktivieren, 1 = Aufbewahrung von nicht-Statusinformationen aktivieren.

contacts:	Dies ist eine Liste der <i>Kurznamen</i> der Kontakte , die über Probleme (oder Erholungen) dieses Hosts informiert werden sollen. Mehrere Kontakte werden jeweils durch ein Komma voneinander getrennt. Nützlich, wenn Benachrichtigungen nur an ein paar Leute gehen sollen und Sie dafür keine Kontaktgruppen definieren wollen. Sie müssen mindestens einen Kontakt oder eine Kontaktgruppe in jeder Host-Definition angeben.
contact_groups:	Dies ist eine Liste der <i>Kurznamen</i> der Kontaktgruppen , die über Probleme (oder Erholungen) dieses Hosts informiert werden sollen. Mehrere Kontaktgruppen werden durch ein Komma voneinander getrennt. Sie müssen mindestens einen Kontakt oder eine Kontaktgruppe in jeder Host-Definition angeben.
notification_interval:	Diese Direktive wird benutzt, um die Anzahl von "Zeiteinheiten" anzugeben, die gewartet werden soll, bevor ein Kontakt erneut darüber informiert werden soll, dass dieser Host <i>immer noch</i> "down" oder unerreichbar ist. Solange Sie nicht die interval_length -Direktive auf einen anderen als den Standardwert von 60 verändert haben, bedeutet diese Zahl Minuten. Wenn Sie diesen Wert auf 0 setzen, wird Icinga die Kontakte <i>nicht</i> erneut über Probleme dieses Hosts informieren - nur eine Problembenachrichtigung wird versandt.
first_notification_delay:	Diese Direktive wird benutzt, um die Anzahl von "Zeiteinheiten" anzugeben, die gewartet werden soll, bevor die erste Problembenachrichtigung versandt wird, wenn dieser Host in einen nicht-UP-Zustand wechselt. Solange Sie nicht die interval_length -Direktive auf einen anderen als den Standardwert von 60 verändert haben, bedeutet diese Zahl Minuten. Wenn Sie diesen Wert auf 0 setzen, wird Icinga sofort Benachrichtigungen versenden.
notification_period:	Diese Direktive wird benutzt, um den Kurznamen des Zeitfensters anzugeben, in dem Benachrichtigungen zu Ereignissen dieses Hosts an Kontakte versandt werden. Wenn ein Host zu einer Zeit "down" geht, unerreichbar wird oder sich wieder erholt, die nicht in diesem Zeitfenster liegt, werden keine Benachrichtigungen versandt.
notification_options:	Diese Direktive wird benutzt, um festzulegen, wann Benachrichtigungen für diesen Host versandt werden. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: d = Benachrichtigungen bei einem DOWN-Zustand versenden, u = Benachrichtigungen bei einem UNREACHABLE-Zustand versenden, r = Benachrichtigungen bei Erholungen (OK-Zustand) versenden, f = Benachrichtigungen versenden, wenn der Host mit Flattern anfängt bzw. aufhört und s = Benachrichtigungen versenden, wenn eine geplante Ausfallzeit anfängt oder aufhört. Wenn Sie n (none) als Option angeben, werden keine Host-Benachrichtigungen versandt. Wenn Sie keine Benachrichtigungsoptionen angeben, geht Icinga davon aus, dass Sie Benachrichtigungen zu allen möglichen Zuständen haben möchten. Beispiel: wenn Sie d,r in diesem Feld angeben, werden Benachrichtigungen nur dann versandt, wenn der Host in einen DOWN-Zustand geht und sich wieder von einem DOWN-Zustand erholt.
notifications_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob Benachrichtigungen für diesen Host aktiviert sind oder nicht. Werte: 0 = Host-Benachrichtigungen deaktivieren, 1 = Host-Benachrichtigungen aktivieren.

stalking_options:

Diese Direktive legt fest, für welche Host-Zustände "Verfolgung" (stalking) aktiviert ist. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: **o** = verfolgen von UP-Zuständen, **d** = verfolgen von DOWN-Zuständen und **u** = verfolgen von UNREACHABLE-Zuständen. Mehr Informationen zur Statusverfolgung finden Sie [hier](#).

notes:

Diese Direktive wird benutzt, um optional einen Text mit Informationen zu diesem Host anzugeben. Wenn Sie hier Anmerkungen angeben, werden Sie diese in der [extended information](#)-CGI sehen (wenn Sie Informationen zu dem entsprechenden Host ansehen).

notes_url:

Diese Variable wird benutzt, um einen optionalen URL anzugeben, der verwendet werden kann, um weitere Informationen zu diesem Host zu liefern. Wenn Sie einen URL angeben, werden Sie ein rotes Verzeichnis-Icon in den CGIs sehen (wenn Sie Host-Informationen betrachten), das auf den URL verweist, den Sie hier angeben. Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. `/cgi-bin/icinga/`). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu diesem Host, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.

action_url:

Diese Direktive wird benutzt, um einen optionalen URL anzugeben, der verwendet werden kann, um weitere Aktionen für diesen Host zu ermöglichen. Wenn Sie einen URL angeben, werden Sie einen roten "Klecks" in den CGIs sehen (wenn Sie Host-Informationen betrachten). Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. `/cgi-bin/icinga/`).

**Anmerkung**

Seit Icinga 1.0.2 ist es möglich, mehrere URLs für `action` | `notes_url` bei Host- und Service-Objektdefinitionen zu anzugeben. Die Syntax ist wie folgt:

```
notes_url|action_url 'ersteURL' 'zweiteURL' 'dritteURL'  
notes_url|action_url nureineURL
```

Bitte achten Sie darauf, dass mehrere URLs auch gleichzeitig mehrere Icon-Bilder bedeuten. Diese sind hartkodiert, so dass z.B. `action` | `notes.gif` zu `1-action` | `1-notes.gif` wird. Stellen Sie sicher, dass diese vorhanden sind. Die letzte URL kann ohne singlequotes angegeben werden und wird dann wie eine einzelne URL betrachtet und verweist auf das normale Icon (`action.gif`).

icon_image:

Diese Variable wird benutzt, um den Namen eines GIF-, PNG oder JPG-Images anzugeben, das mit diesem Host verbunden werden soll. Dieses Bild wird an verschiedenen Stellen in den CGIs angezeigt. Das Bild wird am besten aussehen, wenn es 40x40 Pixel groß ist. Bilder für Hosts werden im `logos/`-Unterverzeichnis Ihres HTML-Images-Verzeichnisses gesucht (d.h. `/usr/local/icinga/share/images/logos`).

icon_image_alt:

Diese Variable wird benutzt, um eine optionale Zeichenkette anzugeben, die für den ALT-Tag des Bildes benutzt wird, das durch das `<icon_image>`-Argument angegeben wurde.

statusmap_image:

Diese Variable wird benutzt, um den Namen eines Bildes anzugeben, das mit diesem Host im [statusmap](#)-CGI verbunden werden soll. Sie können ein JPG-, PNG- oder GIF-Bild angeben, aber wir würden zu einem Bild im GD2-Format raten, weil andere Bildformate zu hohen CPU-Belastungen führen können, wenn die Statusmap generiert wird. PNG-Bilder können mit Hilfe des [pngtogd2](#)-Utilitys (das in Thomas Boutell's [gd library](#) enthalten ist) ins GD2-Format umgewandelt werden. Die GD2-Bilder werden im *unkomprimierten* Format erstellt, um die CPU-Belastung zu minimieren, während das Statusmap-CGI das Netzwerkkartenbild erstellt. Das Bild wird am besten aussehen, wenn es 40x40 Pixel groß ist. Sie können diese Option leer lassen, wenn Sie das Statusmap-CGI nicht nutzen. Bilder für Hosts werden im **logos**-Unterverzeichnis Ihres HTML-Images-Verzeichnisses gesucht (d.h. `/usr/local/icinga/share/images/logos`).

2d_coords:

Diese Variable wird benutzt, um Koordinaten anzugeben, wenn der Host im [statusmap](#)-CGI gezeichnet wird. Koordinaten sollen in positiven Ganzzahlen angegeben werden, weil sie physischen Pixeln im generierten Bild entsprechen. Der Ursprung (0,0) für die Zeichnung ist die linke, obere Ecke des Bildes, das sich in die positive X-Richtung (nach rechts) und in die positive Y-Richtung (nach unten) erstreckt. Die Größe der Icons ist normalerweise etwa 40x40 Pixel (Text benötigt etwas mehr Platz). Die Koordinaten, die Sie angeben, beziehen sich auf die linke, obere Ecke des Icons. Anmerkung: Machen Sie sich keine Sorgen über die maximalen X- und Y-Koordinaten, die Sie benutzen können. Das CGI wird automatisch die maximale Größe des zu erstellenden Bildes aufgrund der größten X- und Y-Koordinaten festlegen, die Sie angegeben haben.

Hostgruppen-Definition

Beschreibung:

Eine Hostgruppen-Definition wird benutzt, um einen oder mehrere Hosts zu gruppieren, um die Konfiguration mit [Objekt-Tricks](#) zu vereinfachen oder für Anzeigezwecke in den [CGIs](#).

Definition:

Anmerkung: "[Direktiven](#)" werden benötigt, die anderen sind optional.

```
define hostgroup{
    hostgroup_name      hostgroup_name
    alias               alias
    members             hosts
    hostgroup_members   hostgroups
    notes               note_string
    notes_url           url
    action_url          url
}
```

Beispieldefinition:

```
define hostgroup{
    hostgroup_name          novell-servers
    alias                   Novell Servers
    members                 netware1,netware2,netware3,netware4
}
```

Beschreibung der Direktiven:

hostgroup_name: Diese Direktive wird benutzt, um einen Kurznamen zu definieren, der die Hostgruppe identifiziert.

alias: Diese Direktive wird benutzt, um einen längeren Namen oder eine Beschreibung zu definieren, der die Hostgruppen identifiziert. Er/sie wird angeboten, damit Sie eine bestimmte Hostgruppe einfacher identifizieren können.

members: Dies ist eine Liste von *Kurznamen* der [Hosts](#), die in dieser Gruppe enthalten sein sollen. Mehrere Hostnamen sollten jeweils durch Komma von einander getrennt werden. Diese Direktive kann als Alternative (oder als Zusatz) zu der *hostgroups*-Direktive in den [Host-Definitionen](#) verwendet werden.

hostgroup_members: Diese optionale Direktive kann genutzt werden, um Hosts aus anderen "sub"-Hostgruppen in diese Hostgruppe aufzunehmen. Geben Sie eine komma-separierte Liste von Kurznamen anderer Hostgruppen an, deren Mitglieder in diese Gruppe aufgenommen werden sollen.

notes: Diese Direktive wird benutzt, um optional einen Text mit Informationen zu dieser Hostgruppe anzugeben. Wenn Sie hier Anmerkungen angeben, werden Sie diese in der [extended information](#)-CGI sehen (wenn Sie Informationen zu der entsprechenden Hostgruppe ansehen).

notes_url: Diese Variable wird benutzt, um einen optionalen URL anzugeben, der verwendet werden kann, um weitere Informationen zu dieser Hostgruppe zu liefern. Wenn Sie einen URL angeben, werden Sie ein rotes Verzeichnis-Icon in den CGIs sehen (wenn Sie Hostgruppen-Informationen betrachten), das auf den URL verweist, den Sie hier angeben. Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. /cgi-bin/icinga/). Dies kann sehr nützlich sein, wenn Sie detaillierte Infomationen zu dieser Hostgruppe, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.

action_url: Diese Direktive wird benutzt, um einen optionalen URL anzugeben, der verwendet werden kann, um weitere Aktionen für diese Hostgruppe zu ermöglichen. Wenn Sie einen URL angeben, werden Sie einen roten "Klecks" in den CGIs sehen (wenn Sie Hostgruppen-Informationen betrachten). Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. /cgi-bin/icinga/).

Service-Definition

Beschreibung:

Eine Service-Definition wird benutzt, um einen "Service" zu identifizieren, der auf einem Host läuft. Der Begriff "Service" wird sehr locker benutzt. Es kann sich um einen realen Service auf einem Host handeln (POP, SMTP, HTTP, etc.) oder eine andere Art von Metrik, die mit dem Host verbunden ist (Antwort auf einen Ping, Anzahl der angemeldeten Benutzer, freier Plattenplatz usw.). Die verschiedenen Parameter einer Service-Definition sind nachfolgend dargestellt.

Definition:

Anmerkung: "**Direktiven**" werden benötigt, die anderen sind optional.

```
define service{
    host_name                      host_name
    hostgroup_name                  hostgroup_name
    service_description             service_description
    display_name                    display_name
    servicegroups                   servicegroup_names
    is_volatile                     [0|1|2]
    check_command                   command_name
    initial_state                   [o,w,u,c]
    max_check_attempts              #
    check_interval                  #
    retry_interval                 #
    active_checks_enabled          [0/1]
    passive_checks_enabled         [0/1]
    check_period                    timeperiod_name
    obsess_over_service            [0/1]
    check_freshness                [0/1]
    freshness_threshold             #
    event_handler                  command_name
    event_handler_enabled          [0/1]
    low_flap_threshold              #
    high_flap_threshold             #
    flap_detection_enabled         [0/1]
    flap_detection_options          [o,w,c,u]
```

```

process_perf_data           [0/1]
retain_status_information   [0/1]
retain_nonstatus_information [0/1]
notification_interval      #
first_notification_delay    #
notification_period         timeperiod_name
notification_options         [w,u,c,r,f,s]
notifications_enabled        [0/1]
contacts                   contacts
contact_groups             contact_groups
stalking_options             [o,w,u,c]
notes                        note_string
notes_url                    url
action_url                   url
icon_image                   image_file
icon_image_alt                alt_string
}

```

Beispieldefinition:

```

define service{
    host_name          linux-server
    service_description check-disk-sda1
    check_command       check-disk! /dev/sda1
    max_check_attempts 5
    check_interval      5
    retry_interval      3
    check_period        24x7
    notification_interval 30
    notification_period 24x7
    notification_options w,c,r
    contact_groups      linux-admins
}

```

Beschreibung der Direktiven:

host_name:

Diese Direktive wird benutzt, um den *Kurznamen* des/der **Hosts** anzugeben, auf denen der Service "läuft" bzw. mit dem/denen er verbunden ist. Mehrere Hosts sind jeweils durch Komma von einander zu trennen.

hostgroup_name:

Diese Direktive wird benutzt, um den/die *Kurznamen* der **Hostgruppe(n)** anzugeben, auf der/denen der Service "läuft" bzw. mit der/denen er verbunden ist. Mehrere Hostgruppen werden durch Kommata von einander getrennt. Der hostgroup_name kann anstatt oder zusätzlich zur host_name-Direktive benutzt werden.

service_description:	Diese Direktive wird benutzt, um eine Beschreibung des Service zu definieren, die Leerzeichen, Bindestriche und Doppelpunkte enthalten kann (Semikolon, Apostroph und Fragezeichen sollten vermieden werden). Keine zwei Services des gleichen Hosts können die gleiche Beschreibung haben. Services werden eindeutig durch die <i>host_name</i> und <i>service_description</i> -Direktiven identifiziert.
display_name:	Diese Direktive wird benutzt, um einen alternativen Namen zu definieren, der im Web-Interface für diesen Service angezeigt wird. Falls nicht angegeben, wird der Wert aus der <i>service_description</i> -Direktive benutzt. Anmerkung: Die CGIs bis einschließlich Icinga 1.0.1 nutzen diese Option nicht.
servicegroups:	Diese Direktive wird benutzt, um den/die Kurznamen der Servicegruppe(n) anzugeben, zu der/denen der Service gehört. Mehrere Servicegruppen werden durch Kommata von einander getrennt. Diese Direktive kann als Alternative (oder zusätzlich) zur <i>members</i> -Direktive in den <i>servicegroup</i> -Definitionen genutzt werden.
is_volatile:	Diese Direktive wird benutzt, um zu kennzeichnen, dass der Service "sprunghaft" (<i>volatile</i>) ist. Services sind normalerweise <i>nicht</i> sprunghaft. Mehr Informationen zu sprunghaften Services und wie sie sich von normalen Services unterscheiden, finden Sie hier . Werte: 0 = Service ist nicht sprunghaft, 1 = Service ist sprunghaft, 2 = Service ist sprunghaft, respektiert aber die Einstellung der Direktive <i>notification_interval</i> bei erneuten Benachrichtigungen (Option "2" gibt es seit Icinga 1.0.2).
check_command:	Diese Direktive wird benutzt, um den <i>Kurznamen</i> des Befehls anzugeben, mit dem der Zustand des Service geprüft wird. Die maximale Zeit, die der Prüfbefehl laufen darf, wird durch die <i>service_check_timeout</i> -Option kontrolliert.
initial_state:	Als Default nimmt Icinga an, dass sich alle Services im OK-Zustand befinden, wenn es startet. Sie können mit dieser Direktive den initialen Zustand eines Service übersteuern. Gültige Optionen sind: o = OK, w = WARNING, u = UNKNOWN und c = CRITICAL.
max_check_attempts:	Diese Direktive wird benutzt, um zu definieren, wie oft Icinga den Service-Prüfbefehl wiederholt, wenn er einen anderen als einen OK-Zustand zurückliefert. Bei einem Wert von 1 wird Icinga einen Alarm generieren, ohne den Service erneut zu prüfen.
check_interval:	Diese Direktive wird benutzt, um die Anzahl von "Zeiteinheiten" zwischen "regulär" geplanten Prüfungen zu definieren. "Reguläre" Prüfungen sind solche, die stattfinden, wenn sich der Service in einem OK-Zustand befindet oder wenn sich der Service in einem nicht-OK-Zustand befindet, aber mehr als max_check_attempts -mal erneut geprüft wurde. Solange Sie die <i>interval_length</i> -Direktive mit einem Default-Wert von 60 nicht verändert haben, wird diese Zahl Minuten bedeuten. Mehr Informationen zu diesem Wert finden Sie in der check scheduling -Dokumentation.

retry_interval:	Diese Direktive wird benutzt, um die Anzahl von "Zeiteinheiten" zu definieren, die zwischen erneuten Überprüfungen des Service gewartet werden sollen. Erneute Überprüfungen für Services werden mit dem Wiederholungsintervall eingeplant, wenn diese in einen nicht-OK-Zustand gewechselt sind. Sobald der Service max_check_attempts -Mal ohne eine Zustandsänderung geprüft wurde, wird die Planung zum "normalen" Wert zurückkehren, der durch den check_interval -Wert angegeben wird. Solange Sie die interval_length -Direktive mit einem Default-Wert von 60 nicht verändert haben, wird diese Zahl Minuten bedeuten. Mehr Informationen zu diesem Wert finden Sie in der check scheduling -Dokumentation.
active_checks_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob aktive Prüfungen (entweder regelmäßig geplant oder nach Bedarf) für diesen Service aktiviert sind oder nicht. Werte: 0 = keine aktiven Service-Prüfungen, 1 = aktive Service-Prüfungen.
passive_checks_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob passive Prüfungen für diesen Service aktiviert sind oder nicht. Werte: 0 = passive Service-Prüfungen deaktivieren, 1 = passive Service-Prüfungen aktivieren
check_period:	Diese Direktive wird benutzt, um den Kurznamen des Zeitfensters anzugeben, in dem aktive Prüfungen für diesen Service ausgeführt werden.
obsess_over_service *:	Diese Direktive legt fest, ob Prüfungen für den Service über ocsp_command "verfolgt" werden sollen.
check_freshness *:	Diese Direktive wird benutzt, um festzulegen, ob Frische-Prüfungen (freshness checks) für diesen Service aktiviert sind oder nicht. Werte: 0 = Frische-Prüfungen deaktivieren, 1 = Frische-Prüfungen aktivieren.
freshness_threshold:	Diese Direktive wird benutzt, um den Frische-Schwellwert (freshness threshold) (in Sekunden) für diesen Service festzulegen. Wenn Sie einen Wert von Null für diese Direktive setzen, wird Icinga automatisch einen Frische-Schwellwert festlegen.
event_handler:	Diese Direktive wird benutzt, um den Kurznamen des Befehls anzugeben, der jedes Mal ausgeführt werden soll, sobald ein Statuswechsel für den Service erkannt wird (d.h. er in einen nicht-OK-Zustand geht oder sich wieder erholt). Lesen Sie die Dokumentation zu Eventhandlern für eine detailliertere Erklärung, wie Scripte zur Behandlung von Ereignissen geschrieben werden. Die maximale Zeit, die ein Eventhandler-Befehl dauern darf, wird durch die event_handler_timeout -Option kontrolliert.
event_handler_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob der Eventhandler für diesen Service aktiviert ist oder nicht. Werte: 0 = Service-Eventhandler deaktivieren, 1 = Service-Eventhandler aktivieren
low_flap_threshold:	Diese Direktive wird benutzt, um den unteren Zustandsänderungsschwellwert zu definieren, der in der Flattererkennung für diesen Service benutzt wird. Mehr Informationen zur Flattererkennung finden Sie hier . Wenn Sie diese Direktive auf 0 setzen, wird der programmweite Wert aus der low_service_flap_threshold -Direktive benutzt.
high_flap_threshold:	Diese Direktive wird benutzt, um den oberen Zustandsänderungsschwellwert zu definieren, der in der Flattererkennung für diesen Service benutzt wird. Mehr Informationen zur Flattererkennung finden Sie hier . Wenn Sie diese Direktive auf 0 setzen, wird der programmweite Wert aus der high_service_flap_threshold -Direktive benutzt.

flap_detection_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob Flattererkennung für diesen Service aktiviert ist. Mehr Informationen zur Flattererkennung finden Sie hier . Werte: 0 = Service-Flattererkennung deaktivieren, 1 = Service-Flattererkennung aktivieren.
flap_detection_options :	Diese Direktive wird benutzt, um festzulegen, welche Service-Zustände die Flattererkennungslogik für diesen Service benutzen wird. Gültige Optionen sind Kombinationen von einem oder mehreren folgender Werte: o = OK-Zustände, w = WARNING-Zustände, c = CRITICAL-Zustände, u = UNKNOWN-Zustände.
process_perf_data *:	Diese Direktive wird benutzt, um festzulegen, ob die Verarbeitung von Performance-Daten für diesen Service aktiviert ist. Werte: 0 = Verarbeitung von Performance-Daten deaktiviert, 1 = Verarbeitung von Performance-Daten aktiviert.
retain_status_information :	Diese Direktive wird benutzt, um festzulegen, ob zustandsbezogene Informationen zu diesem Service über Programmneustarts hinweg aufbewahrt werden. Das ist nur sinnvoll, wenn Sie Statusaufbewahrung über die retain_state_information -Direktive aktiviert haben. Werte: 0 = Aufbewahrung von Statusinformationen deaktivieren, 1 = Aufbewahrung von Statusinformationen aktivieren.
retain_nonstatus_information :	Diese Direktive wird benutzt, um festzulegen, ob nicht-zustandsbezogene Informationen zu diesem Service über Programmneustarts hinweg aufbewahrt werden. Das ist nur sinnvoll, wenn Sie Status-Beibehaltung über die retain_state_information -Direktive aktiviert haben. Werte: 0 = Aufbewahrung von nicht-Statusinformationen deaktivieren, 1 = Aufbewahrung von nicht-Statusinformationen aktivieren.
notification_interval :	Diese Direktive wird benutzt, um die Anzahl von "Zeiteinheiten" anzugeben, die gewartet werden soll, bevor ein Kontakt erneut darüber informiert werden soll, dass dieser Service <i>immer noch</i> in einem nicht-OK-Zustand ist. Solange Sie nicht die interval_length -Direktive auf einen anderen als den Standardwert von 60 verändert haben, bedeutet diese Zahl Minuten. Wenn Sie diesen Wert auf 0 setzen, wird Icinga die Kontakte <i>nicht</i> erneut über Probleme dieses Service informieren - nur eine Problembenachrichtigung wird versandt.
first_notification_delay :	Diese Direktive wird benutzt, um die Anzahl von "Zeiteinheiten" anzugeben, die gewartet werden soll, bevor die erste Problembenachrichtigung versandt wird, wenn dieser Service in einen nicht-OK-Zustand wechselt. Solange Sie nicht die interval_length -Direktive auf einen anderen als den Standardwert von 60 verändert haben, bedeutet diese Zahl Minuten. Wenn Sie diesen Wert auf 0 setzen, wird Icinga sofort Benachrichtigungen versenden.
notification_period :	Diese Direktive wird benutzt, um den Kurznamen des Zeitfensters anzugeben, in dem Benachrichtigungen zu Ereignissen dieses Service an Kontakte versandt werden. Zu Zeiten, die nicht in diesem Zeitfenster liegen, werden keine Benachrichtigungen versandt.

notification_options:	Diese Direktive wird benutzt, um festzulegen, wann Benachrichtigungen für diesen Service versandt werden. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: w = Benachrichtigungen bei einem WARNING-Zustand versenden, u = Benachrichtigungen bei einem UNKNOWN-Zustand versenden, r = Benachrichtigungen bei Erholungen (OK-Zustand) versenden, f = Benachrichtigungen versenden, wenn der Service mit Flattern anfängt bzw. aufhört und s = Benachrichtigungen versenden, wenn eine geplante Ausfallzeit anfängt oder aufhört. Wenn Sie n (none) als Option angeben, werden keine Service-Benachrichtigungen versandt. Wenn Sie keine Benachrichtigungsoptionen angeben, geht Icinga davon aus, dass Sie Benachrichtigungen zu allen möglichen Zuständen haben möchten. Beispiel: wenn Sie w,r in diesem Feld angeben, werden Benachrichtigungen nur dann versandt, wenn der Service in einen WARNING-Zustand geht und sich wieder von einem WARNING-Zustand erholt.
notifications_enabled *:	Diese Direktive wird benutzt, um festzulegen, ob Benachrichtigungen für diesen Service aktiviert sind oder nicht. Werte: 0 = Service-Benachrichtigungen deaktivieren, 1 = Service-Benachrichtigungen aktivieren.
contacts:	Dies ist eine Liste der <i>Kurznamen</i> der Kontakte , die über Probleme (oder Erholungen) dieses Service informiert werden sollen. Mehrere Kontakte werden jeweils durch Kommata voneinander getrennt. Nützlich, wenn Benachrichtigungen nur an ein paar Leute gehen sollen und Sie dafür keine Kontaktgruppen definieren wollen. Sie müssen mindestens einen Kontakt oder eine Kontaktgruppe in jeder Service-Definition angeben.
contact_groups:	Dies ist eine Liste der <i>Kurznamen</i> der Kontaktgruppen , die über Probleme (oder Erholungen) dieses Service informiert werden sollen. Mehrere Kontaktgruppen werden durch Kommata voneinander getrennt. Sie müssen mindestens einen Kontakt oder eine Kontaktgruppe in jeder Service-Definition angeben.
stalking_options:	Diese Direktive legt fest, für welche Service-Zustände "Verfolgung" (stalking) aktiviert ist. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: o = verfolgen von OK-Zuständen, w = verfolgen von WARNING-Zuständen, c = verfolgen von CRITICAL-Zuständen und u = verfolgen von UNKNOWN-Zuständen. Mehr Informationen zur Statusverfolgung finden Sie hier .
notes:	Diese Direktive wird benutzt, um optional einen Text mit Informationen zu diesem Service anzugeben. Wenn Sie hier Anmerkungen angeben, werden Sie diese in der extended information -CGI sehen (wenn Sie Informationen zu dem entsprechenden Service ansehen).
notes_url:	Diese Variable wird benutzt, um einen optionalen URL anzugeben, der benutzt werden kann, um weitere Informationen zu diesem Service zu liefern. Wenn Sie einen URL angeben, werden Sie ein rotes Verzeichnis-Icon in den CGIs sehen (wenn Sie Service-Informationen betrachten), das auf den URL verweist, den Sie hier angeben. Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. <code>/cgi-bin/icinga/</code>). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu diesem Service, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.

action_url:

Diese Direktive wird benutzt, um einen optionalen URL anzugeben, der benutzt werden kann, um weitere Aktionen für diesen Service zu ermöglichen. Wenn Sie einen URL angeben, werden Sie einen roten "Klecks" in den CGIs sehen (wenn Sie Host-Informationen betrachten). Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. `/cgi-bin/icinga/`).

**Anmerkung**

Seit Icinga 1.0.2 ist es möglich, mehrere URLs für `action|notes_url` bei Host- und Service-Objektdefinitionen anzugeben. Die Syntax ist wie folgt:

```
notes_url|action_url 'ersteURL' 'zweiteURL' 'dritteURL'  
notes_url|action_url nureineURL
```

Bitte achten Sie darauf, dass mehrere URLs auch gleichzeitig mehrere Icon-Bilder bedeuten. Diese sind hartkodiert, so dass z.B. `action|notes.gif` zu `1-action|1-notes.gif` wird. Stellen Sie sicher, dass diese vorhanden sind. Die letzte URL kann ohne singlequotes angegeben werden und wird dann wie eine einzelne URL betrachtet und verweist auf das normale Icon (`action.gif`).

icon_image:

Diese Variable wird benutzt, um den Namen eines GIF-, PNG oder JPG-Images anzugeben, das mit diesem Service verbunden werden soll. Dieses Bild wird an verschiedenen Stellen in den CGIs angezeigt. Das Bild wird am besten aussehen, wenn es 40x40 Pixel groß ist. Bilder für Services werden im `logos/-Unterverzeichnis` Ihres HTML-Images-Verzeichnisses gesucht (d.h. `/usr/local/icinga/share/images/logos/`).

icon_image_alt:

Diese Variable wird benutzt, um eine optionale Zeichenkette anzugeben, die für den ALT-Tag des Bildes benutzt wird, das durch das `<icon_image>`-Argument angegeben wurde.

Servicegruppen-Definition

Beschreibung:

Eine Servicegruppen-Definition wird benutzt, um einen oder mehrere Services zu gruppieren, um die Konfiguration mit [Objekt-Tricks](#) zu vereinfachen oder für Anzeigezwecke in den [CGIs](#).

Definition:

Anmerkung: "[Direktiven](#)" werden benötigt, die anderen sind optional.

```
define servicegroup{
    servicegroup_name      servicegroup_name
    alias                  alias
    members                services
    servicegroup_members   servicegroups
    notes                 note_string
    notes_url              url
    action_url             url
}
```

Beispieldefinition:

```
define servicegroup{
    servicegroup_name      dbservices
    alias                  Database Services
    members                ms1,SQL Server,ms1,SQL Server Agent,ms1,SQL DTC
}
```

Beschreibung der Direktiven:

servicegroup_name:	Diese Direktive wird benutzt, um einen Kurznamen zu definieren, der die Servicegruppe identifiziert.
alias:	Diese Direktive wird benutzt, um einen längeren Namen oder eine Beschreibung zu definieren, der die Servicegruppen identifiziert. Er/sie wird angeboten, damit Sie ein bestimmte Servicegruppe einfacher identifizieren können.
members:	Dies ist eine Liste von <i>Kurznamen</i> der Services (und der Namen der entsprechenden Hosts), die in dieser Gruppe enthalten sein sollen. Host- und Service-Namen sollten jeweils durch Komma von einander getrennt werden. Diese Direktive kann als Alternative (oder als Zusatz) zu der <i>servicegroups</i> -Direktive in den Service-Definitionen verwendet werden. Das Format der member-Direktive ist wie folgt (beachten Sie, dass ein Host-Name einem Service-Namen/einer Service-Beschreibung vorangestellt werden muss):
	<code>members=<host1>,<service1>,<host2>,<service2>,...,<hostn>,<servicen></code>
	Seit Icinga 1.2 ist es möglich, "*" als Wildcard für alle Hosts zu benutzen. Bitte beachten Sie, dass es NICHT möglich ist, Hosts oder Service über ein vorangestelltes "!" auszuschließen.
servicegroup_members:	Diese optionale Direktive kann genutzt werden, um Services aus anderen "sub"-Servicegruppen in diese Servicegruppe aufzunehmen. Geben Sie eine komma-separierte Liste von Kurznamen anderer Servicegruppen an, deren Mitglieder in diese Gruppe aufgenommen werden sollen.
notes:	Diese Direktive wird benutzt, um optional einen Text mit Informationen zu dieser Servicegruppe anzugeben. Wenn Sie hier Anmerkungen angeben, werden Sie diese in der extended information -CGI sehen (wenn Sie Informationen zu der entsprechenden Servicegruppe ansehen).
notes_url:	Diese Variable wird benutzt, um einen optionalen URL anzugeben, der benutzt werden kann, um weitere Informationen zu dieser Servicegruppe zu liefern. Wenn Sie einen URL angeben, werden Sie ein rotes Verzeichnis-Icon in den CGIs sehen (wenn Sie Servicegruppen-Informationen betrachten), das auf den URL verweist, den Sie hier angeben. Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. /cgi-bin/icinga/). Dies kann sehr nützlich sein, wenn Sie detaillierte Infomationen zu dieser Servicegruppe, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.
action_url:	Diese Direktive wird benutzt, um einen optionalen URL anzugeben, der benutzt werden kann, um weitere Aktionen für diese Servicegruppe zu ermöglichen. Wenn Sie einen URL angeben, werden Sie einen roten "Klecks" in den CGIs sehen (wenn Sie Servicegruppen-Informationen betrachten). Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. /cgi-bin/icinga/).

Kontakt-Definition

Beschreibung:

Eine Kontakt-Definition wird benutzt, um jemanden zu identifizieren, der im Falle eines Problems in Ihrem Netzwerk informiert werden soll. Die verschiedenen Parameter einer Kontakt-Definition werden nachfolgend beschrieben.

Definition:

Anmerkung: "Direktiven" werden benötigt, die anderen sind optional.

```
define contact{
```

contact_name	contact_name
alias	alias
contactgroups	contactgroup_names
host_notifications_enabled	[0/1]
service_notifications_enabled	[0/1]
host_notification_period	timeperiod_name
service_notification_period	timeperiod_name
host_notification_options	[d,u,r,f,s,n]
service_notification_options	[w,u,c,r,f,s,n]
host_notification_commands	command_name
service_notification_commands	command_name
email	email_address
pager	pager_number or pager_email_gateway
addressx	additional_contact_address
can_submit_commands	[0/1]
retain_status_information	[0/1]
retain_nonstatus_information	[0/1]
}	

Beispieldefinition:

```
define contact{
    contact_name          jdoe
    alias                 John Doe
    host_notifications_enabled 1
    service_notifications_enabled 1
    service_notification_period 24x7
    host_notification_period 24x7
    service_notification_options w,u,c,r
    host_notification_options d,u,r
    service_notification_commands notify-by-email
    host_notification_commands host-notify-by-email
    email                 jdoe@localhost.localdomain
    pager                555-5555@pagergateway.localhost.localdomain
```

```

address1           xxxxx.xyyy@icq.com
address2           555-555-5555
can_submit_commands 1
}

```

Beschreibung der Direktiven:

contact_name:

Diese Direktive wird benutzt, um einen Kurznamen zu definieren, der den Kontakt identifiziert. Er wird in [Kontaktgruppen](#)-Definitionen benutzt. Bei korrekter Anwendung wird das \$CONTACTNAME\$-[Makro](#) diesen Wert enthalten.

alias:

Diese Direktive wird benutzt, um einen längeren Namen oder eine Beschreibung zu definieren, der/die den Kontakt identifiziert. Bei korrekter Anwendung wird das \$CONTACTALIAS\$-[Makro](#) diesen Alias/diese Beschreibung enthalten. Falls nicht angegeben, wird der *contact_name* als Alias verwendet.

contactgroups:

Diese Direktive wird benutzt, um den/die *Kurznamen* der [Kontaktgruppe\(n\)](#) anzugeben, zu dem/denen der Kontakt gehört. Mehrere Kontaktgruppen werden durch Kommata von einander getrennt. Diese Direktive kann als Alternative (oder zusätzlich) zur *members*-Direktive in den [contactgroup](#)-Definitionen genutzt werden.

host_notifications_enabled:

Diese Direktive wird benutzt, um festzulegen, ob der Kontakt Benachrichtigungen über Host-Probleme und Erholungen bekommt. Werte: 0 = keine Benachrichtigungen versenden, 1 = Benachrichtigungen versenden

service_notifications_enabled:

Diese Direktive wird benutzt, um festzulegen, ob der Kontakt Benachrichtigungen über Service-Probleme und Erholungen bekommt. Werte: 0 = keine Benachrichtigungen versenden, 1 = Benachrichtigungen versenden

host_notification_period:

Diese Direktive wird benutzt, um den Kurznamen des [Zeitfensters](#) anzugeben, in dem der Kontakt über Host-Probleme oder Erholungen informiert wird. Sie können dies als "Bereitschafts"-Zeiten dieses Kontakts für Host-Benachrichtigungen ansehen. Lesen Sie die Dokumentation zu [Zeitfenstern](#), um mehr Informationen darüber zu erhalten, wie diese funktionieren und welche potenziellen Probleme durch unsachgemäßen Gebrauch entstehen können.

service_notification_period:

Diese Direktive wird benutzt, um den Kurznamen des [Zeitfensters](#) anzugeben, in dem der Kontakt über Service-Probleme oder Erholungen informiert wird. Sie können dies als "Bereitschafts"-Zeiten dieses Kontakts für Service-Benachrichtigungen ansehen. Lesen Sie die Dokumentation zu [Zeitfenstern](#), um mehr Informationen darüber zu erhalten, wie diese funktionieren und welche potenziellen Probleme durch unsachgemäßen Gebrauch entstehen können.

host_notification_commands:

Diese Direktive wird benutzt, um *Kurznamen* von [Befehlen](#) anzugeben, die zur Benachrichtigung von Kontakten über *Host*-Probleme oder Erholungen benutzt werden. Mehrere Benachrichtigungsbefehle sollten durch Kommata von einander getrennt werden. Alle Benachrichtigungsbefehle werden ausgeführt, wenn der Kontakt informiert werden muss. Die maximale Zeit, die der Benachrichtigungsbefehl laufen darf, wird durch die [notification_timeout](#)-Option kontrolliert.

host_notification_options:

Diese Direktive wird benutzt, um die Host-Zustände festzulegen, bei denen Benachrichtigungen an den Kontakt versandt werden. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: **d** = Benachrichtigungen bei einem DOWN-Zustand versenden, **u** = Benachrichtigungen bei einem UNREACHABLE-Zustand versenden, **r** = Benachrichtigungen bei Erholungen (UP-Zustand) versenden, **f** = Benachrichtigungen versenden, wenn der Host mit [Flattern](#) anfängt bzw. aufhört und **s** = Benachrichtigungen versenden, wenn eine [geplante Ausfallzeit](#) anfängt oder aufhört. Wenn Sie **n** (none) als Option angeben, werden keine Host-Benachrichtigungen versandt.

service_notification_options:

Diese Direktive wird benutzt, um die Service-Zustände festzulegen, bei denen Benachrichtigungen an den Kontakt versandt werden. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: **w** = Benachrichtigungen bei einem WARNING-Zustand versenden, **c** = Benachrichtigungen bei einem CRITICAL-Zustand versenden, **u** = Benachrichtigungen bei einem UNKNOWN-Zustand versenden, **r** = Benachrichtigungen bei Erholungen (OK-Zustand) versenden, **f** = Benachrichtigungen versenden, wenn der Service mit [Flattern](#) anfängt bzw. aufhört und **s** = Benachrichtigungen versenden, wenn eine [geplante Ausfallzeit](#) anfängt oder aufhört. Wenn Sie **n** (none) als Option angeben, werden keine Service-Benachrichtigungen versandt.

service_notification_commands:	Diese Direktive wird benutzt, um <i>Kurznamen</i> von Befehlen anzugeben, die zur Benachrichtigung von Kontakten über <i>Service</i> -Probleme oder Erholungen benutzt werden. Mehrere Benachrichtigungsbefehle sollten durch Kommata von einander getrennt werden. Alle Benachrichtigungsbefehle werden ausgeführt, wenn der Kontakt informiert werden muss. Die maximale Zeit, die der Benachrichtigungsbefehl laufen darf, wird durch die notification_timeout -Option kontrolliert.
email:	Diese Direktive wird benutzt, um ein e-Mail-Adresse für den Kontakt zu definieren. Abhängig von Ihren Benachrichtigungsbefehlen kann sie benutzt werden, um eine Alarm-Mail an den Kontakt zu versenden. Bei korrekter Anwendung wird das \$CONTACTEMAIL\$ -Makro diesen Wert enthalten.
pager:	Diese Direktive wird benutzt, um eine Pager-Nummer für den Kontakt zu definieren. Sie kann auch eine e-Mail-Adresse eines Pager-Gateways (z.B. pagejoe@pagenet.com) sein. Abhängig von Ihren Benachrichtigungsbefehlen kann sie benutzt werden, um eine Alarm-Page an den Kontakt zu versenden. Bei korrekter Anwendung wird das \$CONTACTPAGER\$ -Makro diesen Wert enthalten.
addressx:	Adress-Direktiven werden benutzt, um zusätzliche "Adressen" für den Kontakt zu definieren. Diese Adressen können alles sein - Mobiltelefonnummern, Instant-Messaging-Adressen usw. Abhängig von Ihren Benachrichtigungsbefehlen kann sie benutzt werden, um einen Alarm an den Kontakt zu versenden. Bis zu sechs Adressen können mit Hilfe dieser Direktiven definiert werden (<i>address1</i> bis <i>address6</i>). Das \$CONTACTADDRESSx\$ -Makro wird diesen Wert enthalten.
can_submit_commands:	Diese Direktive wird benutzt, um festzulegen, ob dieser Kontakt über die CGIs externe Befehle an Icinga erteilen kann. Werte: 0 = dem Kontakt die Erteilung von Befehlen verweigern, 1 = dem Kontakt die Erteilung von Befehlen erlauben.
retain_status_information:	Diese Direktive wird benutzt, um festzulegen, ob zustandsbezogene Informationen zu diesem Kontakt über Programmneustarts hinweg aufbewahrt wird. Das ist nur sinnvoll, wenn Sie Statusaufbewahrung über die retain_state_information -Direktive aktiviert haben. Werte: 0 = Aufbewahrung von Statusinformationen deaktivieren, 1 = Aufbewahrung von Statusinformationen aktivieren.

retain_nonstatus_information: Diese Direktive wird benutzt, um festzulegen, ob nicht-zustandsbezogene Informationen zu diesem Kontakt über Programmneustarts hinweg aufbewahrt wird. Das ist nur sinnvoll, wenn Sie Statusaufbewahrung über die **retain_state_information**-Direktive aktiviert haben. Werte: 0 = Aufbewahrung von nicht-Statusinformationen deaktivieren, 1 = Aufbewahrung von nicht-Statusinformationen aktivieren.

Kontaktgruppen-Definition

Beschreibung:

Eine Kontaktgruppen-Definition wird benutzt, um einen oder mehrere **Kontakte** zu gruppieren, um Alarm-/Erholungs-**Benachrichtigungen** zu versenden.

Definition:

Anmerkung: "**Direktiven**" werden benötigt, die anderen sind optional.

```
define contactgroup{
    contactgroup_name      contactgroup_name
    alias                  alias
    members                contacts
    contactgroup_members   contactgroups
}
```

Beispieldefinition:

```
define contactgroup{
    contactgroup_name      novell-admins
    alias                  Novell Administrators
    members                jdoe,rtobert,tzach
}
```

Beschreibung der Direktiven:

contactgroup_name:	Diese Direktive wird benutzt, um einen Kurznamen zu definieren, der die Kontaktgruppe identifiziert.
alias:	Diese Direktive wird benutzt, um einen längeren Namen oder eine Beschreibung zu definieren, der die Kontaktgruppe identifiziert.
members:	Dies ist eine Liste von <i>Kurznamen</i> der Kontakte , die in dieser Gruppe enthalten sein sollen. Mehrere Kontaktnamen sollten jeweils durch Komma von einander getrennt werden. Diese Direktive kann als Alternative (oder als Zusatz) zu der <i>contactgroups</i> -Direktive in den Kontakt-Definitionen verwendet werden.
contactgroup_members:	Diese optionale Direktive kann genutzt werden, um Kontakte aus anderen "sub"-Kontaktgruppen in diese Kontaktgruppe aufzunehmen. Geben Sie eine komma-separierte Liste von Kurznamen anderer Kontaktgruppen an, deren Mitglieder in diese Gruppe aufgenommen werden sollen.

Zeitfenster-Definition (timeperiod)

Beschreibung:

Ein Zeitfenster ist eine Liste von Zeiten an verschiedenen Tagen, die als "gültige" Zeiten für Benachrichtigungen und Service-Prüfungen angesehen werden. Es besteht aus Zeitbereichen für jeden Tag der Woche. Verschiedene Ausnahmen zu den normalen wöchentlichen Zeiten werden unterstützt, u.a.: bestimmte Wochentage, bestimmte Tage eines Monats, Tage eines bestimmten Monats und Kalendertage.

Definition:

Anmerkung: "[Direktiven](#)" werden benötigt, die anderen sind optional.

```
define timeperiod{
    timeperiod_name    timeperiod_name
    alias              alias
    [weekday]          timeranges
    [exception]        timeranges
    exclude            [timeperiod1,timeperiod2,...,timeperiodn]
}
```

Beispiel-Definitionen:

```
define timeperiod{
    timeperiod_name      nonworkhours
    alias                Non-Work Hours
    sunday               00:00-24:00                      ; jeder Sonntag jeder Woche
    monday               00:00-09:00,17:00-24:00       ; jeder Montag jeder Woche
    tuesday              00:00-09:00,17:00-24:00       ; jeder Dienstag jeder Woche
    wednesday            00:00-09:00,17:00-24:00       ; jeder Mittwoch jeder Woche
    thursday              00:00-09:00,17:00-24:00       ; jeder Donnerstag jeder Woche
    friday               00:00-09:00,17:00-24:00       ; jeder Freitag jeder Woche
    saturday              00:00-24:00                      ; jeder Samstag jeder Woche
}
define timeperiod{
    timeperiod_name      misc-single-days
    alias                Misc Single Days
```

```

1999-01-28      00:00-24:00      ; 28. Januar 1999
monday 3        00:00-24:00      ; 3. Montag im Monat
day 2           00:00-24:00      ; 2. Tag im Monat
february 10     00:00-24:00      ; 10. Februar im Jahr
february -1     00:00-24:00      ; letzter Tag im Februar
friday -2       00:00-24:00      ; vorletzter Freitag im Monat
thursday -1 november 00:00-24:00 ; letzter Donnerstag im November
}
define timeperiod{
    timeperiod_name      misc-date-ranges
    alias                Misc Date Ranges
    2007-01-01 - 2008-02-01 00:00-24:00      ; 1. Januar 2007 bis zum 1. Februar 2008
    monday 3 - thursday 4  00:00-24:00      ; 3. Montag bis 4. Donnerstag
    day 1 - 15            00:00-24:00      ; 1. bis 15. Tag
    day 20 - -1           00:00-24:00      ; 20. Tag bis Monatsende
    july 10 - 15          00:00-24:00      ; 10. - 15. Juli
    april 10 - may 15    00:00-24:00      ; 10. April bis zum 15. Mai
    tuesday 1 april - friday 2 may 00:00-24:00 ; 1. Dienstag im April
                                                ; bis zum 2. Freitag im Mai
}
define timeperiod{
    timeperiod_name      misc-skip-ranges
    alias                Misc Skip Ranges
    2007-01-01 - 2008-02-01 / 3 00:00-24:00      ; jeder dritte Tag vom 1. Januar 2008 bis zum 1. Februar 2008
    2008-04-01 / 7        00:00-24:00      ; jeder 7. Tag ab dem 1. April 2008 (ohne Endedatum)
    monday 3 - thursday 4 / 2 00:00-24:00      ; jeder zweite Tag vom 3. Montag bis zum 4. Donnerstag des Monats
    day 1 - 15 / 5         00:00-24:00      ; jeder 5. Tag vom 1. bis zum 15. Tag des Monats
    july 10 - 15 / 2       00:00-24:00      ; jeder zweite Tag vom 10. Juli bis zum 15.Juli
    tuesday 1 april - friday 2 may / 6 00:00-24:00 ; jeder sechste Tag vom 1. Dienstag im April
                                                ; bis zum 2. Freitag im Mai
}

```

Beschreibung der Direktiven:

- timeperiod_name:** Diese Direktive ist der Kurzname, der benutzt wird, um das Zeitfenster zu identifizieren.
- alias:** Diese Direktive ist ein längerer Name oder eine Beschreibung zur Identifizierung des Zeitfensters.
- [weekday]:** Die Wochentags-Direktiven ("sunday" bis "saturday") sind komma-separierte Listen von Zeitbereichen, die "gültige" Zeiten für einen bestimmten Tag der Woche sind. Beachten Sie, dass es sieben verschiedene Tage gibt, für die Sie Zeitbereiche angeben können ("Sunday" bis "Saturday"). Jeder Zeitbereich hat die Form **HH:MM-HH:MM**, wobei die Stunden in einem 24-Stunden-Format angegeben werden. Wenn Sie einen kompletten Tag aus dem Zeitfenster ausschließen wollen, dann geben Sie ihn einfach nicht an.
- [exception]:** Sie können verschiedene Arten von Ausnahmen zum Standard-Wochentagsplan angeben. Ausnahmen können eine Reihe von verschiedenen Formen annehmen, u.a. einzelne Tage eines bestimmten oder jeden Monats, einzelne Wochentage eines Monats oder einzelner Kalenderdaten. Sie können auch einen Bereich von Tagen/Daten und sogar bestimmte Intervalle überspringen, um Funktionalitäten wie "alle drei Tage zwischen diesen Daten" zu erreichen. Statt alle möglichen von Ausnahmen anzugeben, zeigen wir Ihnen anhand der o.g. Beispieldefinitionen, was möglich ist.:-) Wochentage und verschiedene Arten von Ausnahmen haben alle verschiedene Vorrangebenen, so dass es wichtig ist zu verstehen, wie sie sich gegenseitig beeinflussen. Mehr Informationen dazu finden Sie in der Dokumentation zu [Zeitfenstern](#).
- exclude:** Diese Direktive wird benutzt, um die Kurznamen von anderen Zeitfenstern abzugeben, deren Zeitbereiche in diesem Zeitfenster ausgeschlossen werden sollen. Mehrere Zeitfensternamen sind durch Kommata von einander zu trennen.

Befehls-Definition (command)

Beschreibung:

Eine Befehls-Definition ist genau das. Sie definiert einen Befehl. Befehle, die definiert werden können, umfassen u.a. Service-Prüfungen, Host-Benachrichtigungen und Host-Eventhandler. Befehls-Definitionen können [Makros](#) enthalten, aber Sie müssen sicherstellen, dass Sie nur solche Makros verwenden, die unter den gegebenen Umständen "gültig" sind. Mehr Informationen dazu, welche Makros verfügbar und wann sie "gültig" sind, finden Sie [hier](#). Die verschiedenen Argumente einer Befehls-Definition sehen Sie nachfolgend.

Definition:

Anmerkung: "[Direktiven](#)" werden benötigt, die anderen sind optional.

```
define command{
    command_name    command_name
    command_line    command_line
}
```

Beispieldefinition:

```
define command{
    command_name    check_pop
    command_line    /usr/local/icinga/libexec/check_pop -H $HOSTADDRESS$}
```

Beschreibung der Direktiven:

- command_name:** Diese Direktive ist der Kurzname, der zur Identifizierung des Befehls benutzt wird. Er wird u.a. in [Kontakt](#)-, [Host](#)- und [Service](#)-Definitionen (in notification-, check-, und event handler-Direktiven) verwendet.
- command_line:** Diese Direktive wird benutzt, um zu definieren, was wirklich durch Icinga ausgeführt wird, wenn der Befehl für Service- oder Host-Prüfungen, Benachrichtigungen oder [Eventhandler](#) benutzt wird. Vor der Ausführung der Kommandozeile werden alle gültigen [Makros](#) durch die entsprechenden Werte ersetzt. Lesen Sie die Dokumentation, um festzustellen, welche verschiedenen Makros Sie benutzen können. Beachten Sie, dass die Kommandozeile *nicht* von Anführungszeichen eingeschlossen wird. Achten Sie auch darauf, dass Sie bei der Übergabe eines Dollarzeichens (\$) ein weiteres Dollarzeichen zur Maskierung benutzen müssen (aus bar\$foo muss bar\$\$foo werden).

ANMERKUNG: Sie dürfen kein **Semikolon** (;) in der *command_line*-Direktive einsetzen, weil alles danach als Kommentar angesehen wird. Sie können diese Begrenzung umgehen, indem Sie eines der **\$USER\$** -Makros in Ihrem [resource file](#) mit einem Semikolon füllen und dann in der *command_line*-Direktive auf das entsprechende \$USER\$-Makro verweisen.

Wenn Sie während der Laufzeit Parameter an Befehle übergeben wollen, können Sie die **\$ARGn\$-Makros** in der *command_line*-Direktive der Befehlsdefinition benutzen und in den Objektdefinitions-Direktiven (Host-Prüfbefehl, Service-Eventhandler, usw.), die auf den Befehl verweisen, einzelne Argumente durch Ausrufezeichen (!) vom Befehlsnamen (und von einander) trennen. Mehr Informationen, wie Argumente in Befehlsdefinitionen während der Laufzeit verarbeitet werden, finden Sie in der Dokumentation zu [Makros](#).

Service-Abhängigkeits-Definition (servicedependency)

Beschreibung:

Service-Abhängigkeiten sind ein fortgeschrittenes Feature von Icinga, das es Ihnen erlaubt, Benachrichtigungen und aktive Prüfungen von Services in Abhängigkeit vom Status eines oder mehrerer Services zu unterdrücken. Service-Abhängigkeiten sind optional und zielen hauptsächlich auf fortgeschrittene Benutzer mit komplizierten Überwachungsumgebungen. Mehr Informationen, wie Service-Abhängigkeiten arbeiten (lesen Sie dies!), finden Sie [hier](#).

Definition:

Anmerkung: "Direktiven" werden benötigt, die anderen sind optional. Trotz allem müssen Sie mindestens ein Kriterium angeben, damit die Definition von Nutzen ist.

```
define servicedependency{
    dependent_host_name          host_name
    dependent_hostgroup_name      hostgroup_name
    dependent_service_description service_description
    host_name                  host_name
    hostgroup_name                hostgroup_name
    service_description          service_description
    inherits_parent                [0/1]
    execution_failure_criteria    [o,w,u,c,p,n]
    notification_failure_criteria [o,w,u,c,p,n]
    dependency_period             timeperiod_name
}
```

Beispieldefinition:

```
define servicedependency{
    host_name          WWW1
    service_description Apache Web Server
    dependent_host_name WWW1
    dependent_service_description Main Web Site
    execution_failure_criteria n
    notification_failure_criteria w,u,c
}
```

Beschreibung der Direktiven:

dependent_host:

Diese Direktive wird benutzt, um den/die *Kurznamen* des/der [Hosts](#) anzugeben, auf dem der *abhängige Service* "läuft" oder mit dem er verbunden ist. Mehrere Hosts werden durch Kommata von einander getrennt. Bleibt die Direktive leer, so kann dadurch eine "[same host](#)"-Abhängigkeit erstellt werden.

dependent_hostgroup:

Diese Direktive wird benutzt, um den/die *Kurznamen* der [Hostgruppe\(n\)](#) anzugeben, auf der/denen der *abhängige Service* "läuft" oder mit dem er verbunden ist. Mehrere Hostgruppen werden durch Kommata von einander getrennt. Die **dependent_hostgroup**-Direktive kann statt der (oder zusätzlich zur) **dependent_host**-Direktive benutzt werden.

dependent_service_description:

Diese Direktive wird benutzt, um die *Beschreibung* des [abhängigen Service](#) anzugeben.

host_name:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> des/der Hosts anzugeben, auf dem/denen der Service "läuft" oder mit dem/denen er verbunden ist, von dem "es" <i>abhängt</i> (auch als Master-Service bezeichnet). Mehrere Hosts werden durch Kommata von einander getrennt.
hostgroup_name:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> der Hostgruppe(n) anzugeben, auf der/denen der Service "läuft" oder mit der/denen er verbunden ist, von dem "es" <i>abhängt</i> (auch als Master-Service bezeichnet). Mehrere Hostgruppen werden durch Kommata von einander getrennt. Der hostgroup_name kann statt oder zusätzlich zur host_name-Direktive benutzt werden.
service_description:	Diese Direktive wird benutzt, um die <i>Beschreibung</i> des Service anzugeben, von dem "es" <i>abhängt</i> (auch als Master-Service bezeichnet).
inherits_parent:	Diese Direktive gibt an, ob die abhängige Definition Abhängigkeiten von dem Service erbt, von dem sie <i>abhängt</i> (auch als Master-Service bezeichnet). In anderen Worten, wenn der Master-Service von anderen Services abhängt und eine der Abhängigkeiten fehlschlägt, dann wird auch diese Abhängigkeit fehlschlagen.
execution_failure_criteria:	Diese Direktive wird benutzt, um die Kriterien festzulegen, wann der abhängige Service <i>nicht</i> aktiv geprüft werden soll. Wenn der Master-Service in einem der Zustände ist, die wir angeben, wird der <i>abhängige</i> Service nicht aktiv geprüft. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte (mehrere Werte werden durch Kommata von einander getrennt): o = fehlschlagen bei einem OK-Zustand, w = fehlschlagen bei einem WARNING-Zustand, u = fehlschlagen bei einem UNKNOWN-Zustand, c = fehlschlagen bei einem CRITICAL-Zustand und p = fehlschlagen bei einem PENDING-Zustand (d.h. der Service wurde bisher noch nie geprüft). Wenn Sie n (none) als Option angeben, wird die Ausführungsabhängigkeit nie fehlschlagen und die Prüfungen des abhängigen Service werden immer erfolgen (falls andere Bedingungen das erlauben). Beispiel: wenn Sie o,c,u in diesem Feld angeben, dann wird der <i>abhängige</i> Service nicht aktiv geprüft, wenn der Master-Service sich in einem OK-, CRITICAL- oder UNKNOWN-Zustand befindet.

notification_failure_criteria:	Diese Direktive wird benutzt, um die Kriterien festzulegen, wann <i>keine</i> Benachrichtigungen für den abhängigen Service versandt werden sollen. Wenn der Master-Service in einem der Fehler-Zustände ist, die wir angeben, werden keine Benachrichtigungen für den <i>abhängigen</i> Service an die Kontakte versandt. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: o = fehlschlagen bei einem OK-Zustand, w = fehlschlagen bei einem WARNING-Zustand, u = fehlschlagen bei einem UNKNOWN-Zustand, c = fehlschlagen bei einem CRITICAL-Zustand und p = fehlschlagen bei einem PENDING-Zustand (d.h. der Service wurde bisher noch nie geprüft). Wenn Sie n (none) als Option angeben, wird die Benachrichtigungsabhängigkeit nie fehlschlagen und die Benachrichtigungen für den abhängigen Service werden immer erfolgen. Beispiel: wenn Sie w in diesem Feld angeben, dann werden die Benachrichtigungen für den <i>abhängigen</i> Service nicht versandt, wenn der <i>Master-Service</i> sich in einem WARNING-Zustand befindet.
dependency_period:	Diese Direktive wird benutzt, um den Kurznamen eines Zeitfensters anzugeben, in welchem diese Abhängigkeit gültig ist. Wenn diese Direktive nicht angegeben wird, ist die Abhängigkeit zu allen Zeiten gültig.

Serviceescalations-Definition

Beschreibung:

Serviceescalationen sind *komplett optional* und werden benutzt, um Benachrichtigungen für einen bestimmten Service zu eskalieren. Mehr Informationen, wie Eskalationen arbeiten, finden Sie [hier](#).

Definition:

Anmerkung: "**Direktiven**" werden benötigt, die anderen sind optional.

```
define serviceescalation{
    host_name                host_name
    hostgroup_name            hostgroup_name
    servicegroup_name          servicegroup_name
    service_description        service_description
    contacts                  contacts
    contact_groups             contactgroup_name
    first_notification         #
    last_notification          #
    notification_interval      #
    escalation_period          timeperiod_name
    escalation_options          [w,u,c,r]
    escalation_condition        <condition> ( [ & / | ] <condition> )*
    first_warning_notification   #
    last_warning_notification    #
    first_critical_notification   #
    last_critical_notification    #
    first_unknown_notification   #
    last_unknown_notification    #
}
```

Beispieldefinition:

```
define serviceescalation{
    host_name                nt-3
    service_description        Processor Load
    first_notification         4
    last_notification          0
    notification_interval      30
    contact_groups              all-nt-admins,themanager
}
```

Beschreibung der Direktiven:

host_name:

Diese Direktive wird benutzt, um den/die *Kurznamen* des/der [Hosts](#) anzugeben, für den die [Service](#)-Eskalation gilt oder mit dem/denen er verbunden ist.

hostgroup_name:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> der Hostgruppen anzugeben, für den die Service -Eskalation gilt oder mit der/denen er verbunden ist. Mehrere Hostgruppen werden durch Kommata von einander getrennt. Der hostgroup_name kann statt oder zusätzlich zur host_name-Direktive benutzt werden.
servicegroup_name:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> der Servicegruppen anzugeben, für den die Service -Eskalation gilt oder mit der/denen er verbunden ist. Mehrere Servicegruppen werden durch Kommata von einander getrennt. Der servicegroup_name kann statt oder zusätzlich zur service_name-Direktive benutzt werden.
service_description:	Diese Direktive wird benutzt, um die <i>Beschreibung</i> des Service zu identifizieren, auf den die Eskalation zutreffen soll.
first_notification:	Diese Direktive ist eine Zahl, die die <i>erste</i> Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 3 setzen, dann wird diese Eskalation nur dann benutzt, wenn der Service lang genug in einem nicht-OK-Zustand ist, damit eine dritte Benachrichtigung versandt wird.
last_notification:	Diese Direktive ist eine Zahl, die die <i>letzte</i> Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 5 setzen, dann wird diese Eskalation nicht benutzt, wenn mehr als fünf Benachrichtigungen für den Service versandt werden. Wenn der Wert auf Null gesetzt wird, wird dieser Eskalationseintrag immer benutzt (unabhängig davon, wie viele Benachrichtigungen versandt werden.)
contacts:	Dies ist eine Liste von <i>Kurznamen</i> der Kontakte , die informiert werden sollen, wenn es Probleme (oder Erholungen) für diesen Service gibt. Mehrere Kontakte werden durch Kommata von einander getrennt. Das ist nützlich, wenn Sie Benachrichtigungen nur an ein paar Leute verschicken wollen und keine Kontaktgruppen definieren wollen. Sie müssen mindestens einen Kontakt oder eine Kontaktgruppe in jeder Serviceescalations-Definition angeben.
contact_groups:	Dies ist eine Liste von <i>Kurznamen</i> der Kontaktgruppen , die informiert werden sollen, wenn die Service-Benachrichtigung eskaliert. Mehrere Kontaktgruppen werden durch Kommata von einander getrennt. Sie müssen mindestens einen Kontakt oder eine Kontaktgruppe in jeder Serviceescalations-Definition angeben.

notification_interval:	Diese Direktive wird benutzt, um das Intervall festzulegen, in dem Benachrichtigungen versandt werden, wenn diese Eskalation gültig ist. Wenn Sie einen Wert von 0 für dieses Intervall angeben, wird Icinga die erste Benachrichtigung versenden, wenn diese Eskalation gültig wird, dann aber verhindern, dass weitere Benachrichtigungen versandt werden. Benachrichtigungen werden wieder versandt, bis sich der Host erholt. Dies ist nützlich, wenn Sie nach einer bestimmten Zeit keine weiteren Benachrichtigungen versenden wollen. Anmerkung: Wenn mehrere Eskalationseinträge eines Hosts für ein oder mehr Benachrichtigungsbereiche überlappen, wird das kürzeste Benachrichtigungsintervall aller Eskalationseinträge benutzt.
escalation_period:	Diese Direktive wird benutzt, um den Kurznamen des Zeitfensters anzugeben, in dem diese Eskalation gültig ist. Wenn diese Direktive nicht angegeben wird, ist diese Eskalation zu allen Zeiten gültig.
escalation_options:	Diese Direktive wird benutzt, um die Kriterien festzulegen, wann diese Service-Eskalation benutzt wird. Diese Eskalation wird nur benutzt, wenn der Service in einem der Zustände ist, die in dieser Direktive angegeben werden. Wenn diese Direktive nicht in einer Service-Eskalation angegeben wird, ist die Eskalation für alle Service-Zustände gültig. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: r = eskalieren bei einem OK-(Erholungs)-Zustand, w = eskalieren bei einem WARNING-Zustand, u = eskalieren bei einem UNKNOWN-Zustand, und c = eskalieren bei einem CRITICAL-Zustand. Beispiel: wenn Sie w in diesem Feld angeben, dann wird die Eskalation nur benutzt, wenn sich der Service in einem WARNING-Zustand befindet.
escalation_condition:	Diese Direktive ist ab Icinga 1.0.1 verfügbar. Nähere Einzelheiten finden Sie hier .
first_warning_notification:	Diese Direktive ist eine Zahl, die die <i>erste</i> WARNING-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 3 setzen, dann wird diese Eskalation nur dann benutzt, wenn der Service lang genug in einem nicht-OK-Zustand ist, damit eine dritte WARNING-Benachrichtigung versandt wird. Diese Direktive ist ab Icinga 1.0.2 verfügbar.
last_warning_notification:	Diese Direktive ist eine Zahl, die die <i>letzte</i> WARNING-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 5 setzen, dann wird diese Eskalation nicht benutzt, wenn mehr als fünf WARNING-Benachrichtigungen für den Service versandt werden. Wenn der Wert auf Null gesetzt wird, wird dieser Eskalationseintrag immer benutzt (unabhängig davon, wie viele Benachrichtigungen versandt werden). Diese Direktive ist ab Icinga 1.0.2 verfügbar.

first_critical_notification:	Diese Direktive ist eine Zahl, die die <i>erste</i> CRITICAL-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 3 setzen, dann wird diese Eskalation nur dann benutzt, wenn der Service lang genug in einem nicht-OK-Zustand ist, damit eine dritte CRITICAL-Benachrichtigung versandt wird. Diese Direktive ist ab Icinga 1.0.2 verfügbar.
last_critical_notification:	Diese Direktive ist eine Zahl, die die <i>letzte</i> CRITICAL-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 5 setzen, dann wird diese Eskalation nicht benutzt, wenn mehr als fünf CRITICAL-Benachrichtigungen für den Service versandt werden. Wenn der Wert auf Null gesetzt wird, wird dieser Eskalationseintrag immer benutzt (unabhängig davon, wie viele Benachrichtigungen versandt werden). Diese Direktive ist ab Icinga 1.0.2 verfügbar.
first_unknown_notification:	Diese Direktive ist eine Zahl, die die <i>erste</i> UNKNOWN-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 3 setzen, dann wird diese Eskalation nur dann benutzt, wenn der Service lang genug in einem nicht-OK-Zustand ist, damit eine dritte UNKNOWN-Benachrichtigung versandt wird. Diese Direktive ist ab Icinga 1.0.2 verfügbar.
last_unknown_notification:	Diese Direktive ist eine Zahl, die die <i>letzte</i> UNKNOWN-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 5 setzen, dann wird diese Eskalation nicht benutzt, wenn mehr als fünf UNKNOWN-Benachrichtigungen für den Service versandt werden. Wenn der Wert auf Null gesetzt wird, wird dieser Eskalationseintrag immer benutzt (unabhängig davon, wie viele Benachrichtigungen versandt werden). Diese Direktive ist ab Icinga 1.0.2 verfügbar.

Host-Abhängigkeits-Definition (hostdependency)

Beschreibung:

Host-Abhängigkeiten sind ein fortgeschrittenes Feature von Icinga, das es Ihnen erlaubt, Benachrichtigungen von Hosts in Abhängigkeit vom Status eines oder mehrerer Hosts zu unterdrücken. Host-Abhängigkeiten sind optional und zielen hauptsächlich auf fortgeschrittene Benutzer mit komplizierten Überwachungsumgebungen. Mehr Informationen, wie Host-Abhängigkeiten arbeiten (lesen Sie dies!), finden Sie [hier](#).

Definition:

Anmerkung: "Direktiven" werden benötigt, die anderen sind optional.

```
define hostdependency{
    dependent_host_name          host_name
    dependent_hostgroup_name      hostgroup_name
    host_name                   host_name
    hostgroup_name                hostgroup_name
    inherits_parent                [0/1]
    execution_failure_criteria    [o,d,u,p,n]
    notification_failure_criteria [o,d,u,p,n]
    dependency_period             timeperiod_name
}
```

Beispieldefinition:

```
define hostdependency{
    host_name                  WWW1
    dependent_host_name        DBASE1
    notification_failure_criteria d,u
}
```

Beschreibung der Direktiven:

- | | |
|----------------------------------|--|
| dependent_host_name: | Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> des/der <i>abhängigen Hosts</i> zu identifizieren. Mehrere Hosts werden durch Kommata von einander getrennt. |
| dependent_hostgroup_name: | Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> der <i>abhängigen Hostgruppe(n)</i> zu identifizieren. Mehrere Hostgruppen werden durch Kommata von einander getrennt. Der dependent_hostgroup_name kann statt oder zusätzlich zur dependent_host_name-Direktive benutzt werden. |
| host_name: | Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> des/der <i>Hosts</i> anzugeben, von dem "es" <i>abhängt</i> (auch als Master-Host bezeichnet). Mehrere Hosts werden durch Kommata von einander getrennt. |
| hostgroup_name: | Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> der <i>Hostgruppe(n)</i> anzugeben, von dem "es" <i>abhängt</i> (auch als Master-Host bezeichnet). Mehrere Hostgruppen werden durch Kommata von einander getrennt. Der hostgroup_name kann statt oder zusätzlich zur host_name-Direktive benutzt werden. |
| inherits_parent: | Diese Direktive gibt an, ob die abhängige Definition Abhängigkeiten von dem Host erbt, von dem sie <i>abhängt</i> (auch als Master-Host bezeichnet). In anderen Worten, wenn der Master-Host von anderen Hosts abhängt und eine der Abhängigkeiten fehlschlägt, dann wird auch diese Abhängigkeit fehlschlagen. |

execution_failure_criteria:	Diese Direktive wird benutzt, um die Kriterien festzulegen, wann der abhängige Host <i>nicht</i> aktiv geprüft werden soll. Wenn der Master-Host in einem der Zustände ist, die wir angeben, wird der <i>abhängige</i> Host nicht aktiv geprüft. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte (mehrere Werte werden durch Kommata von einander getrennt): o = fehlschlagen bei einem UP-Zustand, u = fehlschlagen bei einem UNREACHABLE-Zustand und p = fehlschlagen bei einem PENDING-Zustand (d.h. der Host wurde bisher noch nie geprüft). Wenn Sie n (none) als Option angeben, wird die Ausführungsabhängigkeit nie fehlschlagen und die Prüfungen des abhängigen Host werden immer erfolgen (falls andere Bedingungen das erlauben). Beispiel: wenn Sie u,d in diesem Feld angeben, dann wird der <i>abhängige</i> Host nicht aktiv geprüft, wenn der <i>Master-Service</i> sich in einem UNREACHABLE- oder DOWN-Zustand befindet.
notification_failure_criteria:	Diese Direktive wird benutzt, um die Kriterien festzulegen, wann <i>keine</i> Benachrichtigungen für den abhängigen Host versandt werden sollen. Wenn der Master-Host in einem der Fehler-Zustände ist, die wir angeben, werden keine Benachrichtigungen für den <i>abhängigen</i> Host an die Kontakte versandt. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: o = fehlschlagen bei einem UP-Zustand, d = fehlschlagen bei einem DOWN-Zustand, u = fehlschlagen bei einem UNREACHABLE-Zustand, und p = fehlschlagen bei einem PENDING-Zustand (d.h. der Host wurde bisher noch nie geprüft). Wenn Sie n (none) als Option angeben, wird die Benachrichtigungsabhängigkeit nie fehlschlagen und die Benachrichtigungen für den abhängigen Host werden immer erfolgen. Beispiel: wenn Sie d in diesem Feld angeben, dann werden die Benachrichtigungen für den <i>abhängigen</i> Host nicht versandt, wenn der <i>Master-Host</i> sich in einem DOWN-Zustand befindet.
dependency_period:	Diese Direktive wird benutzt, um den Kurznamen eines Zeitfensters anzugeben, in welchem diese Abhängigkeit gültig ist. Wenn diese Direktive nicht angegeben wird, ist die Abhängigkeit zu allen Zeiten gültig.

Host-Eskalations-Definition

Beschreibung:

Host-Eskalationen sind *komplett optional* und werden benutzt, um Benachrichtigungen für einen bestimmten Host zu eskalieren. Mehr Informationen, wie Eskalationen arbeiten, finden Sie [hier](#).

Definition:

Anmerkung: "**Direktiven**" werden benötigt, die anderen sind optional.

```
define hostescalation{
    host_name
    hostgroup_name
    contacts
    contact_groups
    first_notification
    last_notification
    notification_interval
    escalation_period
    escalation_options
    escalation_condition
    first_down_notification
    last_down_notification
    first_unreachable_notification
    last_unreachable_notification
}
```

Beispieldefinition:

```
define hostescalation{
    host_name          router-34
    first_notification 5
    last_notification   8
    notification_interval 60
    contact_groups      all-router-admins
}
```

Beschreibung der Direktiven:

host_name:

Diese Direktive wird benutzt, um den/die *Kurznamen* des/der [Hosts](#) anzugeben, für den die Eskalation gilt.

hostgroup_name:

Diese Direktive wird benutzt, um den/die *Kurznamen* der [Hostgruppen](#) anzugeben, für den die Eskalation gilt. Mehrere Hostgruppen werden durch Kommata von einander getrennt. Wenn diese Direktive benutzt wird, trifft die Eskalation auf alle Hosts zu, die Mitglied der angegebenen Hostgruppe(n) sind.

first_notification:

Diese Direktive ist eine Zahl, die die *erste* Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 3 setzen, dann wird diese Eskalation nur dann benutzt, wenn der Host lang genug "down" oder unerreichbar ist, damit eine dritte Benachrichtigung versandt wird.

last_notification:	Diese Direktive ist eine Zahl, die die <i>letzte</i> Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 5 setzen, dann wird diese Eskalation nicht benutzt, wenn mehr als fünf Benachrichtigungen für den Host versandt werden. Wenn der Wert auf Null gesetzt wird, wird dieser Eskalationseintrag immer benutzt (unabhängig davon, wie viele Benachrichtigungen versandt werden.)
contacts:	Dies ist eine Liste von <i>Kurznamen</i> der Kontakte , die informiert werden sollen, wenn es Probleme (oder Erholungen) für diesen Host gibt. Mehrere Kontakte werden durch Kommata von einander getrennt. Das ist nützlich, wenn Sie Benachrichtigungen nur an ein paar Leute verschicken wollen und keine Kontaktgruppen definieren wollen. Sie müssen mindestens einen Kontakt oder eine Kontaktgruppe in jeder Hosteskalations-Definition angeben.
contact_groups:	Dies ist eine Liste von <i>Kurznamen</i> der Kontaktgruppen , die informiert werden sollen, wenn die Host-Benachrichtigung eskaliert. Mehrere Kontaktgruppen werden durch Kommata von einander getrennt. Sie müssen mindestens einen Kontakt oder eine Kontaktgruppe in jeder Hosteskalations-Definition angeben.
notification_interval:	Diese Direktive wird benutzt, um das Intervall festzulegen, in dem Benachrichtigungen versandt werden, wenn diese Eskalation gültig ist. Wenn Sie einen Wert von 0 für dieses Intervall angeben, wird Icinga die erste Benachrichtigung versenden, wenn diese Eskalation gültig wird, dann aber verhindern, dass weitere Benachrichtigungen versandt werden. Benachrichtigungen werden wieder versandt, bis sich der Host erholt. Dies ist nützlich, wenn Sie nach einer bestimmten Zeit keine weiteren Benachrichtigungen versenden wollen. Anmerkung: Wenn mehrere Eskalationseinträge eines Hosts für ein oder mehr Benachrichtigungsbereiche überlappen, wird das kürzeste Benachrichtigungsintervall aller Eskalationseinträge benutzt.
escalation_period:	Diese Direktive wird benutzt, um den Kurznamen des Zeitfensters anzugeben, in dem diese Eskalation gültig ist. Wenn diese Direktive nicht angegeben wird, ist diese Eskalation zu allen Zeiten gültig.

escalation_options:	Diese Direktive wird benutzt, um die Kriterien festzulegen, wann diese Host-Eskalation benutzt wird. Diese Eskalation wird nur benutzt, wenn der Host in einem der Zustände ist, die in dieser Direktive angeben werden. Wenn diese Direktive nicht in einer Host-Eskalation angegeben wird, ist die Eskalation für alle Host-Zustände gültig. Gültige Optionen sind eine Kombination von einem oder mehreren folgender Werte: r = eskalieren bei einem UP-(Erholungs)-Zustand, d = eskalieren bei einem DOWN-Zustand und u = eskalieren bei einem UNREACHABLE-Zustand. Beispiel: wenn Sie d in diesem Feld angeben, dann wird die Eskalation nur benutzt, wenn sich der Host in einem DOWN-Zustand befindet.
escalation_condition:	Diese Direktive ist ab Icinga 1.0.1 verfügbar. Nähere Einzelheiten finden Sie hier .
first_down_notification:	Diese Direktive ist eine Zahl, die die <i>erste</i> DOWN-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 3 setzen, dann wird diese Eskalation nur dann benutzt, wenn der Host lang genug "down" ist, damit eine dritte Benachrichtigung versandt wird. Diese Direktive ist ab Icinga 1.0.2 verfügbar.
last_down_notification:	Diese Direktive ist eine Zahl, die die <i>letzte</i> DOWN-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 5 setzen, dann wird diese Eskalation nicht benutzt, wenn mehr als fünf Benachrichtigungen für den Host versandt werden. Wenn der Wert auf Null gesetzt wird, wird dieser Eskalationseintrag immer benutzt (unabhängig davon, wie viele Benachrichtigungen versandt werden). Diese Direktive ist ab Icinga 1.0.2 verfügbar.
first_unreachable_notification:	Diese Direktive ist eine Zahl, die die <i>erste</i> UNREACHABLE-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 3 setzen, dann wird diese Eskalation nur dann benutzt, wenn der Host lang genug unerreichbar ist, damit eine dritte Benachrichtigung versandt wird. Diese Direktive ist ab Icinga 1.0.2 verfügbar.
last_unreachable_notification:	Diese Direktive ist eine Zahl, die die <i>letzte</i> UNREACHABLE-Benachrichtigung angibt, für die diese Eskalation gilt. Wenn Sie beispielsweise den Wert auf 5 setzen, dann wird diese Eskalation nicht benutzt, wenn mehr als fünf Benachrichtigungen für den Host versandt werden. Wenn der Wert auf Null gesetzt wird, wird dieser Eskalationseintrag immer benutzt (unabhängig davon, wie viele Benachrichtigungen versandt werden). Diese Direktive ist ab Icinga 1.0.2 verfügbar.

erweiterte Hostinformations-Definition (hostextinfo)

Beschreibung:

Einträge für erweiterte Hostinformationen sind grundsätzlich dazu gedacht, die Ausgaben der [status](#)-, [statusmap](#)- und [extinfo](#)-CGIs schöner aussehen zu lassen. Sie haben keinen Einfluss auf die Überwachung und sind vollständig optional.



Hinweis: Alle Direktiven der erweiterten Hostinformations-Definition sind auch in den [Host-Definitionen](#) verfügbar. Dadurch können Sie entscheiden, die nachstehenden Direktiven in Ihren Host-Definitionen zu benutzen, wenn es Ihre Konfigurationen vereinfacht. Separate erweiterte Hostinformations-Definitionen werden weiterhin unterstützt, um Rückwärtskompatibilität zu gewährleisten.

Definition:

Anmerkung: "Direktiven" werden benötigt, die anderen sind optional. Trotz allem müssen Sie mindestens ein Kriterium angeben, damit die Definition von Nutzen ist.

```
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
}
```

Beispieldefinition:

```
define hostextinfo{
    host_name          netware1
    notes              This is the primary Netware file server
    notes_url          http://webserver.localhost.localdomain/hostinfo.pl?host=netware1
    icon_image         novell40.png
    icon_image_alt     IntranetWare 4.11
    statusmap_image    novell40.gd2
    2d_coords          100,250
    3d_coords          100.0,50.0,75.0
}
```

Variablenbeschreibungen:

host_name:	Diese Variable wird benutzt, um den/die <i>Kurznamen</i> des/der Hosts zu identifizieren, mit dem/denen diese Daten verbunden sind.
notes:	Diese Direktive wird benutzt, um eine optionale Zeichenkette mit Anmerkungen zu definieren, die den Host betreffen. Wenn Sie hier eine Anmerkung angeben, werden Sie diese im extended information -CGI sehen (wenn Sie Informationen zu dem bestimmten Host ansehen).
notes_url:	Diese Variable wird benutzt, um einen optionalen URL zu definieren, der mehr Informationen zu diesem Host bereitstellt. Wenn Sie einen URL angeben, werden Sie im extended information -CGI einen Link namens "Extra Host Notes" sehen (wenn Sie Informationen zu dem bestimmten Host ansehen). Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. <code>/cgi-bin/icinga/</code>). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu diesem Host, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.
action_url:	Diese Variable wird benutzt, um einen optionalen URL zu definieren, der mehr Aktionen für diesen Host bereitstellt. Wenn Sie einen URL angeben, werden Sie im extended information -CGI einen Link namens "Extra Host Notes" sehen (wenn Sie Informationen zu dem bestimmten Host ansehen). Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. <code>/cgi-bin/icinga/</code>). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu diesem Host, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.



Anmerkung

Seit Icinga 1.0.2 ist es möglich, mehrere URLs für `action|notes_url` bei Host- und Service-Objektdefinitionen anzugeben. Die Syntax ist wie folgt:

```
notes_url|action_url 'ersteURL' 'zweiteURL' 'dritteURL'  
notes_url|action_url nureineURL
```

Bitte achten Sie darauf, dass mehrere URLs auch gleichzeitig mehrere Icon-Bilder bedeuten. Diese sind hartkodiert, so dass z.B. `action|notes.gif` zu `1-action|1-notes.gif` wird. Stellen Sie sicher, dass diese vorhanden sind. Die letzte URL kann ohne singlequotes angegeben werden und wird dann, wie eine einzelne URL betrachtet und verweist auf das normale Icon (`action.gif`).

icon_image:	Diese Variable wird benutzt, um den Namen eines GIF-, PNG- oder JPG-Images anzugeben, das mit diesem Host verbunden werden soll. Dieses Bild wird in den status - und extended information -CGIs angezeigt. Das Bild wird am besten aussehen, wenn es 40x40 Pixel groß ist. Bilder für Hosts werden im logos -Unterverzeichnis Ihres HTML-Images-Verzeichnisses gesucht (d.h. <code>/usr/local/icinga/share/images/logos</code>).
--------------------	---

icon_image_alt:	Diese Variable wird benutzt, um eine optionale Zeichenkette anzugeben, die für den ALT-Tag des Bildes benutzt wird, das durch das <code><icon_image></code> -Argument angegeben wurde. Das ALT-Tag wird in den status -, extended information - und statusmap -CGIs benutzt.
statusmap_image:	Diese Variable wird benutzt, um den Namen eines Bildes anzugeben, das mit diesem Host im statusmap -CGI verbunden werden soll. Sie können ein JPG-, PNG- oder GIF-Bild angeben, aber wir würden zu einem Bild im GD2-Format raten, weil andere Bildformate zu hohen CPU-Belastungen führen können, wenn die Statusmap generiert wird. PNG-Bilder können mit Hilfe des pngtoggd2 -Utilities (das in Thomas Boutell's gd library enthalten ist) ins GD2-Format umgewandelt werden. Die GD2-Bilder werden im <i>unkomprimierten</i> Format erstellt, um die CPU-Belastung zu minimieren, während das Statusmap-CGI das Netzwerkkartenbild erstellt. Das Bild wird am besten aussehen, wenn es 40x40 Pixel groß ist. Sie können diese Option leer lassen, wenn Sie das Statusmap-CGI nicht nutzen. Bilder für Hosts werden im logos -Unterverzeichnis Ihres HTML-Images-Verzeichnisses gesucht (d.h. <code>/usr/local/icinga/share/images/logos</code>).
2d_coords:	Diese Variable wird benutzt, um Koordinaten anzugeben, wenn der Host im statusmap -CGI gezeichnet wird. Koordinaten sollen in positiven Ganzzahlen angegeben werden, weil sie physischen Pixeln im generierten Bild entsprechen. Der Ursprung (0,0) für die Zeichnung ist die linke, obere Ecke des Bildes, das sich in die positive X-Richtung (nach rechts) und in die positive Y-Richtung (nach unten) erstreckt. Die Größe der Icons ist normalerweise etwa 40x40 Pixel (Text benötigt etwas mehr Platz). Die Koordinaten, die Sie angeben, beziehen sich auf die linke, obere Ecke des Icons. Anmerkung: Machen Sie sich keine Sorgen über die maximalen X- und Y-Koordinaten, die Sie benutzen können. Das CGI wird automatisch die maximale Größe des zu erstellenden Bildes aufgrund der größten X- und Y-Koordinaten festlegen, die Sie angegeben haben.

erweiterte Serviceinformations-Definition (serviceextinfo)

Beschreibung:

Einträge für erweiterte Serviceinformationen sind grundsätzlich dazu gedacht, die Ausgaben der [status](#)- und [extinfo](#)-CGIs schöner aussehen zu lassen. Sie haben keinen Einfluss auf die Überwachung und sind vollständig optional.



Hinweis: Alle Direktiven der erweiterten Serviceinformations-Definition sind auch in den [Service-Definitionen](#) verfügbar. Dadurch können Sie entscheiden, die nachstehenden Direktiven in Ihren Service-Definitionen zu benutzen, wenn es Ihre Konfigurationen vereinfacht. Separate erweiterte Serviceinformations-Definitionen werden weiterhin unterstützt, um Rückwärtskompatibilität zu gewährleisten.

Definition:

Anmerkung: "Direktiven" werden benötigt, die anderen sind optional. Trotz allem müssen Sie mindestens ein Kriterium angeben, damit die Definition von Nutzen ist.

```
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
}
```

Beispieldefinition:

```
define serviceextinfo{
    host_name          linux2
    service_description Log Anomalies
    notes              Security-related log anomalies on secondary Linux server
    notes_url          http://webserver.localhost.localdomain/serviceinfo.pl?host=linux2&service=Log+Anomalies
    icon_image         security.png
    icon_image_alt     Security-Related Alerts
}
```

Variablenbeschreibungen:

- host_name:** Diese Variable wird benutzt, um den/die *Kurznamen* des/der [Hosts](#) zu identifizieren, mit dem/denen der Service verbunden sind.
- service_description:** Diese ist die Beschreibung des [Service](#), mit dem/denen diese Daten verbunden sind.
- notes:** Diese Direktive wird benutzt, um eine optionale Zeichenkette mit Anmerkungen zu definieren, die den Service betreffen. Wenn Sie hier eine Anmerkung angeben, werden Sie diese im [extended Information-CGI](#) sehen (wenn Sie Informationen zu dem bestimmten Service ansehen).
- notes_url:** Diese Variable wird benutzt, um einen optionalen URL zu definieren, der mehr Informationen zu diesem Service bereitstellt. Wenn Sie einen URL angeben, werden Sie im [extended information-CGI](#) einen Link namens "Extra Service Notes" sehen (wenn Sie Informationen zu dem bestimmten Service ansehen). Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. `/cgi-bin/icinga/`). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu diesem Host, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.

action_url: Diese Variable wird benutzt, um einen optionalen URL zu definieren, der mehr Aktionen für diesen Service bereitstellt. Wenn Sie einen URL angeben, werden Sie im [extended information](#)-CGI einen Link namens "Extra Service Notes" sehen (wenn Sie Informationen zu dem bestimmten Service ansehen). Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. `/cgi-bin/icinga/`). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu diesem Host, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.



Anmerkung

Seit Icinga 1.0.2 ist es möglich, mehrere URLs für `action | notes_url` bei Host- und Service-Objektdefinitionen anzugeben. Die Syntax ist wie folgt:

```
notes_url|action_url 'ersteURL' 'zweiteURL' 'dritteURL'  
notes_url|action_url nureineURL
```

Bitte achten Sie darauf, dass mehrere URLs auch gleichzeitig mehrere Icon-Bilder bedeuten. Diese sind hartkodiert, so dass z.B. `action | notes.gif` zu `1-action | 1-notes.gif` wird. Stellen Sie sicher, dass diese vorhanden sind. Die letzte URL kann ohne singlequotes angegeben werden und wird dann wie eine einzelne URL betrachtet und verweist auf das normale Icon (`action.gif`).

icon_image: Diese Variable wird benutzt, um den Namen eines GIF-, PNG- oder JPG-Images anzugeben, das mit diesem Service verbunden werden soll. Dieses Bild wird in den [status](#)- und [extended information](#)-CGIs angezeigt. Das Bild wird am besten aussehen, wenn es 40x40 Pixel groß ist. Bilder für Service werden im [logos](#)-Unterverzeichnis Ihres HTML-Images-Verzeichnisses gesucht (d.h. `/usr/local/icinga/share/images/logos`).

icon_image_alt: Diese Variable wird benutzt, um eine optionale Zeichenkette anzugeben, die für den ALT-Tag des Bildes benutzt wird, das durch das `<icon_image>`-Argument angegeben wurde. Der ALT-Tag wird in den [status](#)-, [extended information](#)- und [statusmap](#)-CGIs benutzt.

Module-Definition

Beschreibung:

Eine Module-Definition wird benutzt, um Informationen zu einem Modul anzugeben. Sie kann anstatt eines `broker_module`-Eintrags in der Hauptkonfigurationsdatei verwendet werden und ist deshalb flexibler (Sie können `cfg_file`/`cfg_dir`-Einträge benutzen, um sie einzuschließen).



Anmerkung

Moduldefinitionen sind seit Icinga 1.4 verfügbar.

Definition:

Anmerkung: "Direktiven" werden benötigt, die anderen sind optional.

```
define module{
    module_name    module name
    path          path
    args          arguments
    module_type   neb
}
```

Beispieldefinitionen:

```
define module{
    module_name    ido_mod
    path          /usr/local/icinga/bin/idomod.o
    module_type   neb
    args          config_file=/usr/local/icinga/etc/idomod.cfg
}
```

Basierend auf der [MKLiveStatus](#)-Dokumentation könnte die module-Definition wie folgt aussehen:

```
define module{
    module_name    mklivestatus
    path          /usr/local/lib/mk-livestatus/livestatus.o
    module_type   neb
    args          /var/lib/nagios/rw/live
}
```

Beschreibung der Direktiven:

module_name: Diese Direktive legt den eindeutigen Namen des Moduls fest, so dass Sie den Modulnamen nicht mehrfach vergeben können. Die Direktive ist notwendig, anderenfalls wird die Konfiguration nicht akzeptiert und das Modul nicht geladen.

module_type: Diese optionale Direktive gibt den Typ des Moduls an, z.B. 'neb' für Eventbroker-Module. Diese Direktive gedacht, um weitere Filterung beim Laden des Moduls zu erlauben.

path: Diese notwendige Direktive gibt den kompletten Pfad des zu ladenden Moduls an. Bei Eventbroker-Modulen wie z.B. idomod muss der Benutzer des Icinga-Prozesses berechtigt sein, das Modul lesen und ausführen zu dürfen.

args: Diese Direktive definiert optionale Argumente, die an das Modul übergeben werden. idomod benötigt config_file=.../idomod.cfg während andere Module ihre eigene Syntax haben. Der Wert der Direktive wird als Argument-String an den Eventbroker-Modul-Lader übergeben, wenn es als neb-Modul benutzt wird.



Anmerkung

Die Nutzung von Templates sollte möglich sein, aber das wurde noch nicht ausgiebig mit Icinga 1.4 getestet.

[Zurück](#)[Nach oben](#)[Weiter](#)[Überblick Objektkonfiguration](#)[Zum Anfang](#)[Maßgeschneiderte
Objektvariablen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Maßgeschneiderte Objektvariablen

[Zurück](#)[Kapitel 3. Icinga konfigurieren](#)[Weiter](#)

Maßgeschneiderte Objektvariablen

Einführung

Benutzer fragen oft nach neuen Variablen in Host-, Service- und Kontaktdefinitionen. Dazu gehören Variablen für die SNMP-Community, MAC-Adressen, AIM-Benutzernamen, Skype-Nummern und Straßennamen. Die Liste ist endlos. Das Problem, was wir darin sehen ist, dass Icinga weniger generisch und mehr infrastrukturspezifisch wird. Icinga war dazu gedacht, flexibel zu sein, was bedeutet, dass die Dinge in einer generischen Art und Weise geplant waren. Host-Definitionen in Icinga zum Beispiel haben eine generische "address"-Variable, die alles von einer IP-Adresse bis zu menschlich lesbaren Wegbeschreibungen enthalten kann - was immer für die Umgebung des Benutzers angemessen ist.

Trotzdem muss es eine Methode für Administratoren geben, in ihrer Icinga-Konfiguration Informationen zu ihren Infrastrukturkomponenten zu speichern, ohne anderen einen Satz von speziellen Variablen aufzubürden. Icinga versucht dieses Problem zu lösen, indem es Benutzern erlaubt, maßgeschneiderte Variablen in ihren Objektdefinitionen anzugeben. Maßgeschneiderte Variablen erlauben es Benutzern, zusätzliche Eigenschaften in ihren Host-, Service- und Kontaktdefinitionen anzugeben und ihre Werte in Benachrichtigungen, Eventhandlern sowie Host- und Service-Prüfungen zu benutzen.

Grundlagen zu maßgeschneiderten Variablen

Es gibt ein paar wichtige Dinge, die Sie bei maßgeschneiderten Variablen beachten sollten:

- maßgeschneiderte Variablennamen müssen mit einem Unterstrich (_) beginnen, um einen Namenskonflikt mit Standardvariablen zu verhindern
- maßgeschneiderten Variablennamen werden vor der Benutzung in Großbuchstaben umgewandelt
- maßgeschneiderten Variablen werden von Objektvorlagen wie normale Variablen **geerbt**
- Scripts können sich mit **Makros** und **Umgebungsvariablen** auf die Werte von maßgeschneiderten Variablen beziehen



Wichtig

Maßgeschneiderte Variablen werden erst *zur Laufzeit* ersetzt, weder bei der Überprüfungsphase (icinga -v icinga.cfg) noch während des Icinga-Startprozesses, so dass das Platzieren in Host-Namen, Service-Beschreibungen oder verschiedenen anderen Stellen der Objektdefinition während der Überprüfung zu einer Fehlermeldung ähnlich "<object> contains one or more illegal characters." führt. Das Ändern der Direktive "illegal_object_name_chars" wird *nicht* das Problem lösen.

Beispiele

Hier ein Beispiel, wie maßgeschneiderte Variablen in verschiedenen Arten von Objektdefinitionen definiert werden können:

```
define host{
    host_name      linuxserver
    _mac_address   00:06:5B:A6:AD:AA          ; <-- Custom MAC_ADDRESS variable
    _rack_number   R32                         ; <-- Custom RACK_NUMBER variable
    ...
}
define service{
    host_name      linuxserver
    description     Memory Usage
    _SNMP_community public                     ; <-- Custom SNMP_COMMUNITY variable
    _TechContact    Jane Doe                  ; <-- Custom TECHCONTACT variable
    ...
}
define contact{
    contact_name   john
    _AIM_username  john16                     ; <-- Custom AIM_USERNAME variable
    _YahooID       john32                     ; <-- Custom YAHOOID variable
    ...
}
```

maßgeschneiderte Variablen als Makros

Maßgeschneiderte Variablen können über **Makros** oder Umgebungsvariablen in Scripts und Programmen eingesetzt werden, die Icinga für Prüfungen, Benachrichtigungen usw. ausführt.

Um Namenskonflikte zwischen maßgeschneiderten Variablen aus verschiedenen Objektarten zu verhindern, stellt Icinga "_HOST", "_SERVICE" oder "_CONTACT" an den Anfang von maßgeschneiderten Host-, Service- oder Kontaktvariablennamen in Makros und Umgebungsvariablen. Die folgende Tabelle zeigt die entsprechenden Namen für maßgeschneiderte Variablen, die im obigen Beispiel definiert wurden.

Objekttyp	Variablenname	Makroname	Umgebungsvariable
Host	MAC_ADDRESS	\$_HOSTMAC_ADDRESS\$	ICINGA__HOSTMAC_ADDRESS
Host	RACK_NUMBER	\$_HOSTRACK_NUMBER\$	ICINGA__HOSTRACK_NUMBER
Service	SNMP_COMMUNITY	\$_SERVICESNMP_COMMUNITY\$	ICINGA__SERVICESNMP_COMMUNITY
Service	TECHCONTACT	\$_SERVICETECHCONTACT\$	ICINGA__SERVICETECHCONTACT
Contact	AIM_USERNAME	\$_CONTACTAIM_USERNAME\$	ICINGA__CONTACTAIM_USERNAME
Contact	YAHOOID	\$_CONTACTYAHOOID\$	ICINGA__CONTACTYAHOOID

maßgeschneiderte Variablen und Vererbung

Maßgeschneiderte Objektvariablen werden genau wie Standard-Host-, Service- oder Kontaktvariablen [vererbt](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Objektdefinitionen](#)

[Zum Anfang](#)

[Optionen](#)

[CGI-Konfigurationsdatei](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Optionen CGI-Konfigurationsdatei

[Zurück](#)
[Kapitel 3. Icinga konfigurieren](#)
[Weiter](#)

Optionen CGI-Konfigurationsdatei

Anmerkungen

Wenn Sie Konfigurationsdateien erstellen oder anpassen, beachten Sie bitte folgendes:

1. Zeilen, die mit einem '#' -Zeichen beginnen, werden als Kommentar betrachtet und nicht verarbeitet
2. Variablennamen müssen am Zeilenanfang beginnen - "white space" sind vor dem Namen NICHT erlaubt
3. Variablennamen sind abhängig von Groß- und Kleinschreibung

Beispielkonfiguration



Hinweis: eine Beispiel-CGI-Konfigurationsdatei (`/usr/local/icinga/etc/cgi.cfg`) wird für Sie installiert, wenn Sie der [Schnellstart-Anleitung](#) folgen.

Position der Konfigurationsdatei

Als Standard erwartet Icinga, dass die CGI-Konfigurationsdatei `cgi.cfg` heißt und zusammen mit der [Hauptkonfigurationsdatei](#) im Verzeichnis für die Konfigurationsdateien liegt. Wenn Sie den Namen der Datei oder die Position ändern müssen, dann können Sie Apache so konfigurieren, dass eine Umgebungsvariable namens `ICINGA_CONFIG` übergeben wird (die auf die korrekte Position der CGIs verweist). Lesen Sie in der Apache-Dokumentation nach, wie das zu tun ist.

Variablen der Konfigurationsdatei

Nachfolgend finden Sie Beschreibungen jeder Option der Hauptkonfigurationsdatei...

Position der Hauptkonfigurationsdatei

Format: `main_config_file=<file_name>`

Beispiel: `main_config_file=/usr/local/icinga/etc/icinga.cfg`

Dies gibt die Position Ihrer [Hauptkonfigurationsdatei](#) an. Die CGIs müssen wissen, wo sie zu finden ist, um Informationen zu Konfigurationsinformationen, aktuellen Host- und Service-Zuständen usw. zu bekommen.

vollständiger (physical) HTML-Pfad

Format: **physical_html_path=<Pfad>**

Beispiel: **physical_html_path=/usr/local/icinga/share**

Dies ist der *vollständige* Pfad zu den HTML-Dateien von Icinga auf Ihrer Workstation oder Ihrem Server. Icinga nimmt an, dass die Dokumentation und die Bilddateien (die von den CGIs benutzt werden) in Unterverzeichnissen namens *docs/* und *images/* gespeichert sind.

URL-HTML-Pfad

Format: **url_html_path=<Pfad>**

Beispiel: **url_html_path=/icinga**

Wenn Sie Icinga über einen Web-Browser mit einem URL wie **http://www.myhost.com/icinga**, aufrufen, sollte dieser Wert */icinga* sein. Grundsätzlich ist es der Pfadanteil des URL, der zum Aufruf der Icinga-HTML-Seiten benutzt wird.

URL Stylesheet Path

Format: **url_stylesheet_path=<path>/stylesheets**

Beispiel: **url_stylesheet_path=/icinga/stylesheets**

Diese Option erlaubt Ihnen die Angabe einer alternativen Stylesheet URL. Dies ist nützlich, wenn Sie benutzerdefinierte Stylesheets aus einem anderen Verzeichnis hinzufügen möchten. Wird sie nicht gesetzt, wird der Standard-Speicherort verwendet (`url_stylesheets_path=url_html_path/stylesheets`).

Nutzung der Authentifizierung

Format: **use_authentication=[0/1]**

Beispiel: **use_authentication=1**

Diese Option kontrolliert, ob die CGIs die Authentifizierungs- und Autorisierungsfunktionalität nutzen, um den Zugang von Benutzern zu Informationen und Befehlen zu prüfen oder nicht. Wir möchten dringend raten, dass Sie die Authentifizierungsfunktionalität für die CGIs nutzen. Wenn Sie sich entscheiden, die Authentifizierung nicht zu nutzen, dann stellen Sie sicher, dass das [command CGI](#) entfernt wird, um nicht autorisierte Benutzer an der Ausführung von Icinga-Befehlen zu hindern. Das CGI wird keine Befehle ausführen, wenn Authentifizierung deaktiviert ist, aber wir würden trotzdem dazu raten, das CGI zu entfernen, damit man auf der sicheren Seite ist. Mehr Informationen zur Einstellung der Authentifizierung und der Konfiguration von Autorisierung für die CGIs finden Sie [hier](#).

- 0 = die Authentifizierungsfunktionalität nicht nutzen
- 1 = die Authentifizierungs- und Autorisierungsfunktionalität nutzen (Default)

Standard-Benutzername

Format: **default_user_name=<username>**

Beispiel: **default_user_name=guest**

Das Setzen dieser Variable definiert einen Standard-Benutzernamen, der die CGIs aufrufen kann. Dies erlaubt es Leuten in einer sicheren Domäne (d.h. hinter einer Firewall) die CGIs aufzurufen, ohne dass sie sich am Web-Server authentifizieren müssen. Sie können das benutzen, um die Basis-Authentifizierung zu verhindern, wenn Sie keinen sicheren Server einsetzen, weil Basis-Authentifizierung Passwörter im Klartext über das Internet überträgt.

Wichtig: Definieren Sie *keinen* Standard-Benutzernamen, solange Sie nicht einen sicheren Web-Server haben und sicher sind, dass sich jeder, der die CGIs aufruft, in irgendeiner Weise authentifiziert hat. Wenn Sie diese Variable definieren, dann wird jeder, der sich am Web-Server authentifiziert, alle Rechte dieses Benutzers erben!

Zugang zu System/Prozessinformationen

Format: **authorized_for_system_information=<user1>,<user2>,<user3>,...<usern>**

Beispiel: **authorized_for_system_information=icingaadmin,theboss**

Dies ist eine Komma-separierte Liste von Namen von *authentifizierten Benutzern*, die System/Prozessinformationen im [extended information CGI](#) ansehen können. Benutzer in dieser Liste sind *nicht* automatisch autorisiert, System/Prozessbefehle zu erteilen. Wenn Sie möchten, dass Benutzer auch System/Prozessbefehle erteilen können, dann müssen Sie diese der [authorized_for_system_commands](#)-Variable hinzufügen. Mehr Informationen, wie man Authentifizierung einrichtet und Autorisierung für die CGIs konfiguriert, finden Sie [hier](#).



Anmerkung

Ab Icinga 1.5.0 können Sie alternativ `authorized_contactgroup_for_system_information` angeben, um statt eines contact eine contactgroup zu autorisieren.

Zugang zu System/Prozessbefehlen

Format: **authorized_for_system_commands=<user1>,<user2>,<user3>,...<usern>**

Beispiel: **authorized_for_system_commands=icingaadmin**

Dies ist eine Komma-separierte Liste von Namen von *authentifizierten Benutzern*, die System/Prozessbefehle über das [command CGI](#) erteilen können. Benutzer in dieser Liste sind *nicht* automatisch autorisiert, System/Prozessinformationen anzusehen. Wenn Sie möchten, dass Benutzer auch System/Prozessinformationen ansehen können, dann müssen Sie diese der [authorized_for_system_information](#)-Variable hinzufügen. Mehr Informationen, wie man Authentifizierung einrichtet und Autorisierung für die CGIs konfiguriert, finden Sie [hier](#).



Anmerkung

Ab Icinga 1.5.0 können Sie alternativ `authorized_contactgroup_for_system_commands` angeben, um statt eines contact eine contactgroup zu autorisieren.

Zugang zu Konfigurationsinformationen

Format: **authorized_for_configuration_information=<user1>,<user2>,<user3>,...<usern>**

Beispiel: **authorized_for_configuration_information=icingaadmin**

Dies ist eine Komma-separierte Liste von Namen von *authentifizierten Benutzern*, die Konfigurationsinformationen im **configuration CGI** ansehen können. Benutzer in dieser Liste können Informationen zu allen konfigurierten Hosts, Hostgruppen, Kontakten, Kontaktgruppen, Zeitfenstern und Befehlen ansehen. Mehr Informationen, wie man Authentifizierung einrichtet und Autorisierung für die CGIs konfiguriert, finden Sie [hier](#).



Anmerkung

Ab Icinga 1.5.0 können Sie alternativ **authorized_contactgroup_for_configartion_information** angeben, um statt eines contact eine contactgroup zu autorisieren.

Global Host Information Access

Format: **authorized_for_all_hosts=<user1>,<user2>,<user3>,...<usern>**

Beispiel: **authorized_for_all_hosts=icingaadmin,theboss**

Dies ist eine Komma-separierte Liste von Namen von *authentifizierten Benutzern*, die Status- und Konfigurationsinformationen im für alle Hosts ansehen können. Benutzer in dieser Liste sind automatisch autorisiert, Informationen zu allen Services anzusehen. Benutzer in dieser Liste sind *nicht* automatisch berechtigt, Befehle für alle Hosts oder Services zu erteilen. Wenn Sie möchten, dass Benutzer auch Befehle für alle Hosts oder Services erteilen können, dann müssen Sie diese der **authorized_for_all_host_commands**-Variable hinzufügen. Mehr Informationen, wie man Authentifizierung einrichtet und Autorisierung für die CGIs konfiguriert, finden Sie [hier](#).



Anmerkung

Ab Icinga 1.5.0 können Sie alternativ **authorized_contactgroup_for_all_hosts** angeben, um statt eines contact eine contactgroup zu autorisieren.

Zugang zu globalen Host-Befehlen

Format: **authorized_for_all_host_commands=<user1>,<user2>,<user3>,...<usern>**

Beispiel: **authorized_for_all_host_commands=icingaadmin**

Dies ist eine Komma-separierte Liste von Namen von *authentifizierten Benutzern*, die Befehle für alle Hosts über das **command CGI** erteilen können. Benutzer in dieser Liste sind auch automatisch autorisiert, Befehle für alle Services zu erteilen. Benutzer in dieser Liste sind *nicht* automatisch berechtigt, Status- oder Konfigurationsinformationen für alle Hosts oder Services anzusehen. Wenn Sie möchten, dass Benutzer auch Status- und Konfigurationsinformationen für alle Hosts oder Services ansehen können, dann müssen Sie diese der **authorized_for_all_hosts**-Variable hinzufügen. Mehr Informationen, wie man Authentifizierung einrichtet und Autorisierung für die CGIs konfiguriert, finden Sie [hier](#).



Anmerkung

Ab Icinga 1.5.0 können Sie alternativ `authorized_contactgroup_for_all_host_commands` angeben, um statt eines contact eine contactgroup zu autorisieren.

Zugang zu globalen Service-Informationen

Format: `authorized_for_all_services=<user1>,<user2>,<user3>,...<usern>`

Beispiel: `authorized_for_all_services=icingaadmin,theboss`

Dies ist eine Komma-separierte Liste von Namen von *authentifizierten Benutzern*, die Status- und Konfigurationsinformationen für alle Services ansehen können. Benutzer in dieser Liste sind *nicht* automatisch autorisiert, Informationen zu allen Hosts anzusehen. Benutzer in dieser Liste sind *nicht* automatisch berechtigt, Befehle für alle Services zu erteilen. Wenn Sie möchten, dass Benutzer auch Befehle für alle Services erteilen können, dann müssen Sie diese der `authorized_for_all_service_commands`-Variable hinzufügen. Mehr Informationen, wie man Authentifizierung einrichtet und Autorisierung für die CGIs konfiguriert, finden Sie [hier](#).



Anmerkung

Ab Icinga 1.5.0 können Sie alternativ `authorized_contactgroup_for_all_services` angeben, um statt eines contact eine contactgroup zu autorisieren.

Zugang zu globalen Service-Befehlen

Format: `authorized_for_all_service_commands=<user1>,<user2>,<user3>,...<usern>`

Beispiel: `authorized_for_all_service_commands=icingaadmin`

Dies ist eine Komma-separierte Liste von Namen von *authentifizierten Benutzern*, die Befehle für alle Services über das [command CGI](#) erteilen können. Benutzer in dieser Liste sind *nicht* automatisch autorisiert, Befehle für alle Hosts zu erteilen. Benutzer in dieser Liste sind *nicht* automatisch berechtigt, Status- oder Konfigurationsinformationen für alle Hosts anzusehen. Wenn Sie möchten, dass Benutzer auch Status- und Konfigurationsinformationen für alle Services ansehen können, dann müssen Sie diese der `authorized_for_all_services`-Variable hinzufügen. Mehr Informationen, wie man Authentifizierung einrichtet und Autorisierung für die CGIs konfiguriert, finden Sie [hier](#).



Anmerkung

Ab Icinga 1.5.0 können Sie alternativ `authorized_contactgroup_for_all_service_commands` angeben, um statt eines contact eine contactgroup zu autorisieren.

Zeige alle Services für einen berechtigten Host

Format: `show_all_services_host_is_authorized_for=[0|1]`

Beispiel: `show_all_services_host_is_authorized_for=1`

Per Default kann ein Benutzer alle Services eines Hosts sehen, wenn er als Kontakt für den Host autorisiert ist. Durch deaktivieren dieser Option muss der Benutzer auch autorisierter Kontakt des Service sein, um diesen ansehen zu können. Bitte beachten Sie, dass diese Option keine Auswirkung hat, wenn die Option `authorized_for_all_services` für den Benutzer gesetzt ist (seit Icinga 1.0.2).

Zugang zu Befehlen und Kommentaren verweigern

Format: **`authorized_for_read_only=<user1>,<user2>,<user3>,...<usern>`**

Beispiel: **`authorized_for_read_only=icingalook`**

Dies ist eine Komma-separierte Liste von Namen von *authentifizierten Benutzern*, die Nur-Lesen-Zugriff auf die CGIs haben. Dies blockiert jegliche Host- oder Service-Befehle. Es werden auch keine Kommentare angezeigt. Benutzer in dieser Liste sind *nicht* automatisch autorisiert, Befehle für alle Hosts zu erteilen. Benutzer in dieser Liste sind *nicht* automatisch berechtigt, Status- oder Konfigurationsinformationen für alle Hosts anzusehen. Wenn Sie möchten, dass Benutzer auch Status- und Konfigurationsinformationen für alle Services ansehen können, dann müssen Sie diese der `authorized_for_all_services`-Variable hinzufügen. Mehr Informationen, wie man Authentifizierung einrichtet und Autorisierung für die CGIs konfiguriert, finden Sie [hier](#).



Anmerkung

Ab Icinga 1.5.0 können Sie alternativ `authorized_contactgroup_for_read_only` angeben, um statt eines contact eine contactgroup zu autorisieren.

Autorennamen sperren

Format: **`lock_author_names=[0/1]`**

Beispiel: **`lock_author_names=1`**

Diese Option erlaubt es Ihnen, Benutzer daran zu hindern, den Autorennamen zu ändern, wenn sie Kommentare, Bestätigungen und geplanten Ausfallzeiten über das Web-Interface eingeben. Wenn diese Option aktiviert ist, können Benutzer nicht mit der Befehlsanfrage verbundene Autorennamen ändern.

- 0 = Benutzern erlauben, den Autorennamen bei der Erteilung von Befehlen zu ändern
- 1 = Benutzer daran hindern, den Autorennamen zu ändern (default)

Status Show Long Plugin Output

Format: **`status_show_long_plugin_output=[0/1]`**

Beispiel: **`status_show_long_plugin_output=1`**

Diese Option erlaubt es Ihnen, den Umfang der Statusinformationen von Plugins festzulegen, die in der `status.cgi` angezeigt werden. Wenn Sie den Wert auf 1 setzen, dann wird die komplette Ausgabe angezeigt, sonst nur die erste Zeile.

- 0 = Nur die erste Zeile der Plugin-Ausgabe anzeigen (default)
- 1 = Die komplette Plugin-Ausgabe anzeigen



Anmerkung

Diese Option ist verfügbar ab Icinga 1.0.3.

Statusmap CGI Background Image

Format: **statusmap_background_image=<image_file>**

Beispiel: **statusmap_background_image=smbbackground.gd2**

Diese Option erlaubt es Ihnen, ein Bild anzugeben, das als Hintergrund im [statusmap CGI](#) benutzt wird, wenn Sie die Layout-Methode mit benutzerdefinierten Koordinaten benutzen. Das Hintergrundbild ist nicht in anderen Layout-Methoden verfügbar. Es wird angenommen, dass sich das Bild im HTML-Image-Pfad befindet (d.h. in /usr/local/icinga/share/images). Dieser Pfad wird automatisch durch das Anhängen von "/images" an den in der [physical_html_path](#)-Direktive ermittelt. Anmerkung: Die Bilddatei kann im GIF-, JPEG-, PNG- oder GD2-Format sein. Das GD2-Format (vorzugsweise im unkomprimierten Format) wird empfohlen, weil es die CPU-Belastung reduziert, wenn das CGI das Kartenbild generiert.

Standard-Statusmap-Layout-Methode

Format: **default_statusmap_layout=<layout_number>**

Beispiel: **default_statusmap_layout=4**

Gültige Werte sind:

<layout_number>-Wert	Layout-Methode
0	User-defined coordinates
1	Depth layers
2	Collapsed tree
3	Balanced tree
4	Circular
5	Circular (Marked Up)
6	Circular (Balloon)

CGI-Aktualisierungsrate

Format: **refresh_rate=<rate_in_seconds>**

Beispiel: **refresh_rate=90**

Diese Option erlaubt es Ihnen, die Anzahl von Sekunden zwischen Seitenaktualisierungen für die [status](#)-, [statusmap](#)- und [extinfo](#)-CGIs festzulegen.

Audio-Alarme

Format: **host_unreachable_sound=<sound_file>**
host_down_sound=<sound_file>
service_critical_sound=<sound_file>
service_warning_sound=<sound_file>
service_unknown_sound=<sound_file>

Beispiele: **host_unreachable_sound=hostu.wav**
host_down_sound=hostd.wav
service_critical_sound=critical.wav
service_warning_sound=warning.wav
service_unknown_sound=unknown.wav

Diese Option erlaubt es Ihnen, eine Audio-Datei anzugeben, die in Ihrem Browser abgespielt wird, wenn es ein Problem gibt, während Sie das [status CGI](#) ansehen. Wenn es mehrere Probleme gibt, wird die Datei für das kritischste Problem abgespielt. Das kritischste Problem sind ein oder mehrere nicht erreichbare Host, während das am wenigsten kritische Problem Services in einem UNKNOWN-Zustand sind (beachten Sie die Reihenfolge im obigen Beispiel). Audio-Dateien werden im **media**-Unterverzeichnis Ihres HTML-Verzeichnisses erwartet (d.h. `/usr/local/icinga/share/media`).

Ping-Syntax

Format: **ping_syntax=<command>**
Beispiel: **ping_syntax=/bin/ping -n -U -c 5 \$HOSTADDRESS\$**

Diese Option legt fest, welche Syntax benutzt wird, wenn ein Host vom WAP-Interface aus ange"ping"t wird (mit Hilfe des [statuswml CGI](#). Sie müssen den kompletten Pfad zum ping-Binary zusammen mit allen benötigten Optionen angeben. Das `$HOSTADDRESS$`-Makro wird durch die Adresse des Hosts ersetzt, bevor der Befehl ausgeführt wird.

Escape HTML Tags Option

Format: **escape_html_tags=[0/1]**
Beispiel: **escape_html_tags=1**

Diese Option legt fest, ob HTML-Tags in Host- und Service-(Plugin-)Ausgaben in CGIs unberücksichtigt bleiben oder nicht. Wenn Sie diese Option aktivieren, wird die Plugin-Ausgabe keine anklickbaren Hyperlinks enthalten.

Notes URL Target

Format: **notes_url_target=[target]**
Beispiel: **notes_url_target=_blank**

Diese Option legt den Namen des Ziel-Frames fest, in dem Anmerkungs-URLs angezeigt werden sollen. Gültige Optionen umfassen `_blank`, `_self`, `_top`, `_parent` oder jeden anderen gültigen Ziellnamen.

Action URL Target

Format: **action_url_target=[target]**

Beispiel: **action_url_target=_blank**

Diese Option legt den Namen des Ziel-Frames fest, in dem Aktions-URLs angezeigt werden sollen. Gültige Optionen umfassen `_blank`, `_self`, `_top`, `_parent` oder jeden anderen gültigen Ziellnamen.

Tac Show Only Hard State

Format: **tac_show_only_hard_state=[0/1]**

Beispiel: **tac_show_only_hard_state=1**

Diese Option erlaubt Ihnen in der Tactical Overview nur HARD States von Host und Services anzeigen zu lassen. Standardmäßig ist diese Option deaktiviert. Setzen Sie `tac_show_only_hard_state=1`, werden in der Tactical Overview nur noch HARD States angezeigt.

Splunk-Integrationsoption

Format: **enable_splunk_integration=[0/1]**

Beispiel: **enable_splunk_integration=1**

Diese Option legt fest, ob die Integration mit Splunk im Web-Interface aktiviert ist oder nicht. Wenn sie aktiviert ist, werden an verschiedenen Stellen "Splunk It"-Links in den CGIs angezeigt (Log-Datei, Alarmhistorie, Host-/Service-Details, usw.). Das ist nützlich, wenn Sie nach den Ursachen suchen, warum ein bestimmtes Problem auftrat. Für mehr Informationen über Splunk besuchen Sie <http://www.splunk.com/>.

Splunk-URL

Format: **splunk_url=<path>**

Beispiel: **splunk_url=http://127.0.0.1:8000/**

Diese Option wird benutzt, um den Basis-URL zu Ihrem Splunk-Interface zu definieren. Dieser URL wird von den CGIs benutzt, wenn Links erzeugt werden, falls die `enable_splunk_integration`-Option aktiviert ist.

Persistente Bestätigungskommentare

Format: **persistent_ack_comments=<0|1>**

Beispiel: **persistent_ack_comments=1**

Diese Option legt fest, ob die Check-Box "persistent comment" zur Bestätigung von Problem-Hosts oder -Services aktiviert ist. Sie kann benutzt werden, um das Verhalten von Nagios 2 wiederherzustellen. Default ist "0", um kompatibel mit Nagios 3 zu sein.

- 0 = Check-Box "persistent comment" deaktiviert lassen (Default)
- 1 = Check-Box "persistent comment" aktivieren



Anmerkung

Diese Option ist verfügbar ab Icinga 1.0.3.

Initialen Status anzeigen

Format: **showlog_initial_states=<0|1>**

Beispiel: **showlog_initial_states=1**

Diese Option legt fest, ob die initialen Zustände von Hosts und Services in showlog.cgi angezeigt werden sollen.



Anmerkung

Diese Option hat nur Auswirkungen, wenn die Option "log_initial_states" in der icinga.cfg gesetzt wurde.

- 0 = initiale Zustände in showlog.cgi nicht anzeigen
- 1 = initiale Zustände in showlog.cgi anzeigen (Default)



Anmerkung

Diese Option ist verfügbar ab Icinga 1.3.

Aktuellen Status anzeigen

Format: **showlog_current_states=<0|1>**

Beispiel: **showlog_current_states=1**

Diese Option legt fest, ob die aktuellen Zustände von Hosts und Services in showlog.cgi angezeigt werden sollen.



Anmerkung

Diese Option hat nur Auswirkungen, wenn die Option "log_current_states" in der icinga.cfg gesetzt wurde.

- 0 = aktuelle Zustände in showlog.cgi nicht anzeigen
- 1 = aktuelle Zustände in showlog.cgi anzeigen (Default)



Anmerkung

Diese Option ist verfügbar ab Icinga 1.3.

Objekttyp in Tab-Titel anzeigen

Format: **tab_friendly_titles=<0|1>**

Beispiel: **tab_friendly_titles=1**

Durch Aktivieren dieser Option ändern sich die Titel der Reiter im Browser, um den jeweiligen Objekttyp anzuzeigen. Sie zeigen dann:

- [Host]
- {Hostgruppe}
- Service-Beschreibung @ Host
- (Servicegruppe)

Diese sind einfache zu lesen (und zu finden), wenn Sie (viele) Reiter in Ihrem Browser anzeigen.

- 0 = Objekttyp im Reiter nicht anzeigen
- 1 = Objekttyp im Reiter anzeigen (Default)



Anmerkung

Diese Option ist verfügbar ab Icinga 1.3.

Service-Zustand und Benachrichtigungsnummer anzeigen

Format: **add_notif_num_hard=n**

add_notif_num_soft=n

Beispiel: **add_notif_num_hard=28**

Wenn der Wert der Direktive(n) größer Null ist, werden in der status.cgi neben dem "Versuch" (z.B. "3/3" für einen Hard-Nicht-OK-Zustand mit max_check_attempts=3) auch die aktuelle Benachrichtigungsnummer ("#0") falls noch keine Benachrichtigung versandt wurde angezeigt. Dies ist hilfreich, um Services zu identifizieren, die oft zwischen verschiedenen Nicht-OK-Zuständen wechseln, oder Services, bei denen first_notification_delay gesetzt ist, die aber noch nicht "in Schwierigkeiten" sind.

Relevante Werte aus include/statusdata.h (sehen Sie *dort* nach, um *wirklich* sicher zu sein):

```
#define SERVICE_PENDING      1
#define SERVICE_OK           2
#define SERVICE_WARNING      4
#define SERVICE_UNKNOWN      8
#define SERVICE_CRITICAL     16
```

Sie werden wahrscheinlich add_notif_num_hard=0 (Default) oder add_notif_num_hard=28 (warn+crit+unknown) setzen.

Der Vollständigkeit halber gibt es auch add_notif_num_soft für Services im SOFT-Zustand.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.3.

HTTP-Zeichensatz einstellen

Format: **http_charset=<Zeichensatz>**

Beispiel: **http_charset=utf-8**

Hiermit kann der Zeichensatz eingestellt werden, der mit den HTTP-Headern gesendet wird. Default ist utf-8.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.3.

Ersten Tag der Woche setzen

Format: **first_day_of_week=<0|1>**

Beispiel: **first_day_of_week=1**

Diese Option legt den ersten Tag der Woche fest, der in verschiedenen CGI-Reports benutzt wird. Default ist 0 = Sonntag. 1 = Montag gilt für Länder, die sich nach ISO 8601 richten.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.4.

CGI-Protokoll (use logging)

Format: **use_logging=<0|1>**

Beispiel: **use_logging=1**

Diese Variable gibt an, ob die CGI-Kommandos protokolliert werden sollen. 0 = nicht protokollieren (Default), 1 = protokollieren.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.4.

CGI-Protokolldatei (CGI-Log File)

Format: **cgi_log_file=<file_name>**

Beispiel: **cgi_log_file=/usr/local/icinga/share/log/icinga-cgi.log**

Diese Variable gibt an, wo Icinga die CGI-Protokolldatei anlegen soll.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.4.

CGI-Protokollrotationsmethode (CGI Log Rotation Method)

Format: **cgi_log_rotation_method=<d/w/m>**

Beispiel: **cgi_log_rotation_method=d**

Dies ist die Rotationsmethode, die Icinga für Ihre CGI-Protokolldatei nutzen soll. Die Werte sind wie folgt:

- d = täglich ("daily" - die Protokolldatei jeden Tag um Mitternacht rotieren - Default)
- w = wöchentlich ("weekly" - die Protokolldatei jeden Samstag um Mitternacht rotieren)
- m = monatlich ("monthly" - die Protokolldatei am letzten Tag des Monats um Mitternacht rotieren)



Anmerkung

Diese Option ist verfügbar ab Icinga 1.4.

CGI-Protokollarchiv-Pfad (CGI Log Archiv Path)

Format: **cgi_log_archive_path=<path>**

Beispiel: **cgi_log_archive_path=/usr/local/icinga/share/log/**

Dies ist das Verzeichnis, in dem Icinga die CGI-Protokolldateien ablegen soll, die rotiert wurden. Diese Option wird ignoriert, wenn Sie die Funktionalität der [CGI-Protokollrotation](#) (CGI log rotation) nicht nutzen.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.4.

Erzwingen von Kommentaren bei Aktionen

Format: **enforce_comments_on_actions=<0|1>**

Beispiel: **enforce_comments_on_actions=1**

Erzwingt die Notwendigkeit zur Eingabe eines Kommentares bei Aktionen per CGI. 0 = keine Kommentare erzwingen (Default), 1 = Kommentar erzwingen.



Anmerkung

Die Option use_logging muss aktiviert sein, anderenfalls werden keine Kommentare protokolliert.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.4.

Tactical Overview-Header anzeigen

Format: **show_tac_header=<0|1>**

Beispiel: **show_tac_header=0**

Innerhalb der CGIs gibt es einen neuen Header, der per Default aktiviert ist. Die neue Ansicht hat ein ähnliches Aussehen wie der Header in Icinga-Web und enthält wichtige taktische und Überwachungs-Performance-Informationen, so dass Sie immer über die aktuelle Situation informiert sind.

Wenn Sie die alte minimalistische Ansicht beibehalten möchten, dann kann der neue Header durch folgenden Eintrag deaktiviert werden: `show_tac_header=0`

Das Layout der Anzeige für die Einträge jedes Typs lautet "X / Y / Z", wobei:

X = Aktiv, unbestätigt (active unacknowledged)

Y = Passiv, unbestätigt (passive unacknowledged)

Z = Bestätigt (acknowledged)

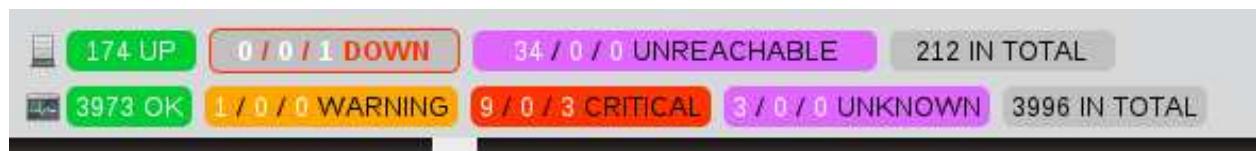
Die X/Y/Z-Zahlen selbst sind anklickbar und führen Sie zu einer Liste mit allen Hosts bzw. Services mit der o.g. Eigenschaft.

Die Farbgebung dieser Einträge richtet sich nach folgendem Schema:

1+ / * / *	= Vollfarbe
0 / 1+ / *	= Farbe abgeschwächt, Rand Vollfarbe
0 / 0 / 1+	= Rand Vollfarbe, Text fett und farbig
0 / 0 / *	= Grau

Diese Farben sind in `html/stylesheets/tacheader.css` abgelegt, damit sie einfach angepasst werden können. Wenn Sie z.B. eine verteilte Umgebung haben und primär passive Prüfungen einsetzen, dann möchten Sie ggf., dass passive Prüfungen farblich wie aktive Prüfungen dargestellt werden.

Abbildung 3.1. Beispiel des neuen Headers



Anmerkung

Diese Option ist verfügbar ab Icinga 1.4.

Pending-Anzahlen in Tactical Overview-Header anzeigen

Format: **show_tac_header_pending=<0|1>**

Beispiel: **show_tac_header_pending=0**

Diese Option aktiviert die Anzeige von Pending-Anzahlen im Tactical-Overview-Header. Wenn Ihre Auflösung kleiner als 1024x768 und diese Option aktiviert ist, dann passen die taktischen Informationen ggf. nicht in den verfügbaren Platz. Diese Option ist per Default deaktiviert.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.4.1.

Default-Ausfallzeit-Dauer setzen

Format: **default_downtime_duration=n**

Beispiel: **default_downtime_duration=7200**

Diese Option setzt die Default-Dauer (in Sekunden) von festen und flexiblen Ausfallzeiten. Default ist 7200 Sekunden (2 Stunden).



Anmerkung

Diese Option ist verfügbar ab Icinga 1.5.

Maintenance/Downtime anzeigen

Format: **suppress_maintenance_downtime=<0|1>**

Beispiel: **suppress_maintenance_downtime=0**

Die aktuelle Classic-UI-Ansicht zeigt den Status von Hosts zusammen mit Aufmerksamkeit erregenden Farben, selbst wenn der Host/ Service sich in einer Ausfallzeit befindet. Dies führt zur Ablenkung von tatsächlich existierenden Problemen durch geplante oder erwartete Ausfälle. Die Option ist per Default deaktiviert, um das alte Verhalten beizubehalten. Falls die Option aktiviert ist, werden die Ausgaben nicht mehr in der regulären Farbe, sondern in grau angezeigt (wie im Pending-Status). Der Status jedoch zeigt weiterhin den tatsächlichen Zustand und das ist beabsichtigt, damit der Benutzer den Zustand sieht, bevor die Ausfallzeit endet. Ein sinnvolles Beispiel dafür ist, dass ein Host/Service wieder online ist, an dem gearbeitet wird. Außerdem werden Hosts in einer geplanten Ausfallzeit in der statusmap.cgi in grau dargestellt.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.5.

Unvollständige Hostgruppen anzeigen

Format: **show_partial_hostgroups=<0|1>**

Beispiel: **show_partial_hostgroups=1**

Per Default (gleichbedeutend mit einer nicht vorhandenen oder deaktivierten Direktive) sieht ein Benutzer eine Hostgruppe und deren Hosts nur, wenn er ein autorisierter Kontakt für *alle* Hosts der jeweiligen Gruppe ist. Das Verhalten kann mit der Direktive `show_partial_hostgroups=1` geändert werden.

Wenn sie aktiviert ist, wird der Hostgruppen-Überblick auch unvollständige Hostgruppen aufführen, bei denen lediglich die Hosts angezeigt werden, für die der Benutzer ein autorisierter Kontakt ist.

Außerdem wird der Hinweis "(Partial Hostgroups Enabled)" oberhalb des Hostgruppen-Überblicks angezeigt, damit keine Verwirrung darüber entsteht, ob diese Direktive aktiviert ist oder nicht. Aus Geheimhaltungsgründen werden unvollständige Hostgruppen nicht als solche gekennzeichnet.

KOMPATIBILITÄTSHINWEIS: Wie bei jeder Änderung an der Ausgabe der CGIs kann die Aktivierung dieser Direktive die Funktion von Drittanbieter-Tools beeinträchtigen, die sich auf eine bestimmte (Bildschirm-)Struktur verlassen. Wenn Sie auf ein solches Problem stoßen, dann deaktivieren Sie diese Option wieder und ermuntern Sie den/die Entwickler des Programms, statt dessen JSON zur Ermittlung der Daten zu nutzen.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.5.

Zeilen in status.cgi hervorheben

Format: `highlight_table_rows=<0|1>`

Beispiel: `highlight_table_rows=0`

Mit dieser Option können Zeilen in der status.cgi hervorgehoben werden. Das kann bei vielen Zeilen oder einer hohen Bildschirmauflösung nützlich sein. Werte: 0 = deaktivieren der Hervorhebung, 1 = aktivieren der Hervorhebung (Default).



Anmerkung

Diese Option ist verfügbar ab Icinga 1.5.

Verfallszeit von Bestätigungen setzen

Format: `default_expiring_acknowledgement_duration=<n>`

Beispiel: `default_expiring_acknowledgement_duration=86400`

Mit dieser Option kann die Verfallszeit von Bestätigungen (acknowledgements) festgelegt werden. Nach Ablauf dieser Zeit werden Bestätigungen automatisch gelöscht. Default ist 86.400 Sekunden (ein Tag).



Anmerkung

Diese Option ist verfügbar ab Icinga 1.6.

Child-Host(s) zeigen

Format: **extinfo_show_child_hosts=<0|1|2>**

Beispiel: **extinfo_show_child_hosts=1**

Mit dieser Option wird festgelegt, ob Child-Host(s) für den ausgewählten Host im Hostextinfo-CGI angezeigt werden.

- 0 = deaktiviert (Default)
- 1 = nur direkte Child-Hosts anzeigen
- 2 = direkte und darunter liegende Child-Hosts anzeigen



Wichtig

Option 2 kann ein richtiger Performance-Killer in großen Installationen sein, deshalb bitte mit Vorsicht einsetzen.



Anmerkung

Diese Option ist per Default deaktiviert, da sie die gesamte Performance beeinträchtigen kann.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.6.

Benutzer(n) VOLLEN Zugriff auf Kommandozeilenanzeigen erlauben

Format: **authorized_for_full_command_resolution=<user1>,<user2>,<user3>,...<usern>**

Beispiel: **authorized_for_full_command_resolution=icingaadmin**

Diese Option ist eine kommaseparierte Liste von Benutzern, die im "Config Command Expander" den Befehl so sehen, wie er von Icinga ausgeführt würde. Damit alle MACROS aufgelöst werden können, benötigt der Web-Server zusätzlich Lesezugriff auf `resource.cfg`.

Alternativ können Sie auch Kontaktgruppen angeben.



Wichtig

`$USERn$`- und benutzerdefinierte Variable können sensible Daten enthalten, so dass Sie den Zugriff strikt begrenzen sollten.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.6.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Maßgeschneiderte
Objektvariablen

[Zum Anfang](#)

Authentifizierung und
Autorisierung in den CGIs

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Authentifizierung und Autorisierung in den CGIs

[Zurück](#)
[Kapitel 3. Icinga konfigurieren](#)
[Weiter](#)

Authentifizierung und Autorisierung in den CGIs

Einführung

Dieses Dokument beschreibt, wie die Icinga-CGIs entscheiden, wer die Überwachungs- und Konfigurationsinformationen sehen darf und wer über das Web-Interface Befehle an den Icinga-Daemon erteilen darf.

Definitionen

Bevor wir fortfahren, ist es wichtig, dass Sie die Bedeutung und den Unterschied zwischen authentifizierten Benutzern und authentifizierten Kontakten verstehen:

- Ein **authentifizierter Benutzer** ist jemand, der sich dem Web-Server gegenüber mit Benutzer und Passwort authentifiziert hat und dem Zugang zum Icinga-Web-Interface gewährt wurde.
- Ein **authentifizierter Kontakt** ist ein authentifizierter Benutzer, dessen Benutzername mit dem Kurznamen einer **Kontakt-Definition** übereinstimmt.

Erstellen von authentifizierten Benutzern

Wenn wir annehmen, dass Sie Ihren Web-Server wie in der [Schnellstart-Anleitung](#) konfiguriert haben, dann sollte er Sie dazu auffordern, sich zu authentifizieren, bevor Sie die Icinga-CGIs benutzen können. Sie sollten außerdem ein Benutzerkonto (*icingadmin*) haben, das Zugang zu den CGIs hat.

Während Sie weitere **Kontakte** definieren, um Host- und Service-Benachrichtigungen zu erhalten, möchten Sie wahrscheinlich auch, dass sie Zugang zum Icinga-Web-Interface haben. Sie können den folgenden Befehl benutzen, um zusätzliche Benutzer hinzuzufügen, die sich bei den CGI authentifizieren können. Ersetzen Sie `<username>` durch den Benutzernamen, den Sie hinzufügen möchten. In den meisten Fällen sollte der Benutzername mit dem Kurznamen eines **Kontakts** übereinstimmen, den Sie definiert haben.

```
htpasswd /usr/local/icinga/etc/htpasswd.users <username>
```

Aktivieren der Authentifizierungs/Autorisierungsfunktionalität in den CGIs

Als nächstes sollten Sie sicherstellen, dass die CGI so konfiguriert sind, dass sie die Authentifizierungs- und Autorisierungsfunktionalität nutzen, um festzulegen, welchen Zugang Benutzer zu Informationen und/oder Befehlen haben. Dies wird durch die **use_authentication**-Variable in der [CGI-Konfigurationsdatei](#) erreicht, die einen Wert ungleich Null haben muss. Beispiel:

```
use_authentication=1
```

Okay, nun sind Sie fertig mit dem Einstellen der grundlegenden Authentifizierungs- und Autorisierungsfunktionalität in den CGIs.

Standardberechtigungen für CGI-Informationen

Welche Standardberechtigungen haben Benutzer in den CGIs, wenn die Authentifizierungs- und Autorisierungsfunktionalität aktiviert ist?

CGI-Daten	Authentifizierte Kontakte *	andere authentifizierte Benutzer *
Host-Statusinformationen	Ja	Nein
Host-Konfigurationsinformationen	Ja	Nein
Host-Verlauf	Ja	Nein
Host-Benachrichtigungen	Ja	Nein
Host-Befehle	Ja	Nein
Service-Statusinformationen	Ja	Nein
Service-Konfigurationsinformationen	Ja	Nein
Service-Verlauf	Ja	Nein
Service-Benachrichtigungen	Ja	Nein
Service-Befehle	Ja	Nein
Alle Konfigurationsinformationen	Nein	Nein
System/Prozessinformationen	Nein	Nein
System/Prozessbefehle	Nein	Nein

*Authentifizierten Kontakten ** werden die folgenden Berechtigungen für jeden **Service** gewährt, bei dem sie als Kontakt eingetragen sind (aber "Nein" für Services, bei denen sie nicht als Kontakt eingetragen sind)...

- Autorisierung, um Service-Statusinformationen zu sehen
- Autorisierung, um Service-Konfigurationsinformationen zu sehen
- Autorisierung, um Verlauf und Benachrichtigungen für den Service zu sehen
- Autorisierung, um Service-Befehle zu erteilen

*Authentifizierten Kontakten ** werden die folgenden Berechtigungen für jeden **Host** gewährt, bei dem sie als Kontakt eingetragen sind (aber "Nein" für Hosts, bei denen sie nicht als Kontakt eingetragen sind)...

- Autorisierung, um Host-Statusinformationen zu sehen
- Autorisierung, um Host-Konfigurationsinformationen zu sehen

- Autorisierung, um Verlauf und Benachrichtigungen für den Host zu sehen
- Autorisierung, um Host-Befehle zu erteilen
- Autorisierung, um Statusinformationen für alle Services des Hosts zu sehen
- Autorisierung, um Konfigurationsinformationen für alle Services des Hosts zu sehen
- Autorisierung, um Verlauf und Benachrichtigungen für alle Services des Host zu sehen
- Autorisierung, um Befehle für alle Services des Hosts zu erteilen

Es ist wichtig anzumerken, dass als Grundeinstellung **keiner** autorisiert ist, das Folgende zu tun:

- die Log-Datei über das [showlog CGI](#) anzusehen
- Icinga-Prozessinformationen über das [extended information CGI](#) anzusehen
- Icinga-Prozessbefehle über das [command CGI](#) zu erteilen
- Definitionen für Hostgruppen, Kontakte, Kontaktgruppen, Zeitfenster und Befehle über das [configuration CGI](#) anzusehen

Sie werden unzweifelhaft Zugang zu diesen Informationen haben wollen, so dass Sie wie unten beschrieben zusätzliche Rechte für sich (und vielleicht andere Benutzer) zuweisen möchten.

Zusätzliche Berechtigungen zu CGI-Informationen gewähren

Uns ist klar, dass die verfügbaren Optionen es nicht erlauben, sehr genau auf bestimmte Berechtigungen einzugehen, aber es ist besser als nichts...

Benutzern können zusätzliche Autorisierungen gegeben werden, indem sie den folgenden Variablen in der CGI-Konfigurationsdatei hinzugefügt werden...

- [authorized_for_system_information](#)
- [authorized_for_system_commands](#)
- [authorized_for_configuration_information](#)
- [authorized_for_all_hosts](#)
- [authorized_for_all_host_commands](#)
- [authorized_for_all_services](#)
- [authorized_for_all_service_commands](#)

CGI-Autorisierungsanforderungen

Wenn Sie verwirrt sind, welche Autorisierung Sie benötigen, um Zugang zu verschiedenen Informationen in den CGIs zu bekommen, lesen Sie [hier](#) den Abschnitt *Autorisierungsanforderungen*, in dem jedes CGI beschrieben ist.

Authentifizierung auf sicheren Web-Servern

Wenn Ihr Web-Server in einer sicheren Domäne steht (d.h. hinter einer Firewall) oder wenn Sie SSL benutzen, dann können Sie einen Standard-Benutzernamen definieren, der verwendet werden kann, um die CGI aufzurufen. Dies wird durch die Definition der [default_user_name](#)-Option in der [CGI-Konfigurationsdatei](#) erreicht. Durch die Definition eines Standard-Benutzernamens, der die CGIs aufrufen kann, können Sie Benutzern erlauben, die CGIs aufzurufen, ohne dass sie sich am Web-Server authentifizieren müssen. Sie möchten das vielleicht nutzen, um die Verwendung der Basis-Web-Authentifizierung zu verhindern, weil diese Passwörter im Klartext über das Internet überträgt.

Wichtig: Definieren Sie *keinen* Standard-Benutzernamen, solange Sie nicht einen sicheren Web-Server haben und sicher sind, dass sich jeder, der die CGIs aufruft, in irgendeiner Weise authentifiziert hat. Wenn Sie diese Variable definieren, dann wird jeder, der sich am Web-Server authentifiziert, alle Rechte dieses Benutzers erben!

[Zurück](#)[Nach oben](#)[Weiter](#)[Optionen](#)
[CGI-Konfigurationsdatei](#)[Zum Anfang](#)[Kapitel 4. Icinga
starten/stoppen/prüfen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Kapitel 4. Icinga starten/stoppen/prüfen

[Zurück](#)

[Weiter](#)

Kapitel 4. Icinga starten/stoppen/prüfen

Inhaltsverzeichnis

[Überprüfen Ihrer Icinga-Konfiguration](#)
[Icinga starten und stoppen](#)

[Zurück](#)

[Weiter](#)

Authentifizierung und
Autorisierung in den CGIs

[Zum Anfang](#)

Überprüfen Ihrer
Icinga-Konfiguration

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Überprüfen Ihrer Icinga-Konfiguration

[Zurück](#)
[Kapitel 4. Icinga starten/stoppen/prüfen](#)
[Weiter](#)

Überprüfen Ihrer Icinga-Konfiguration

Überprüfen Ihrer Konfiguration

Jedes Mal, nachdem Sie Ihre [Konfigurationsdateien](#) verändert haben, sollten Sie sie überprüfen. Es ist wichtig, das zu tun, bevor Sie Icinga (neu)starten, weil Icinga herunterfährt, wenn Ihre Konfiguration Fehler enthält.

Um Ihre Konfiguration zu überprüfen, starten Sie Icinga mit der **-v**-Option wie folgt:

```
#> /usr/local/icinga/bin/icinga -v /usr/local/icinga/etc/icinga.cfg
```

Wenn Sie vergessen haben, kritische Daten einzugeben oder Dinge fehlkonfiguriert haben, spuckt Icinga eine Warnung oder eine Fehlermeldung aus, die Ihnen die Stelle des Problems zeigen sollte. Fehlermeldungen geben grundsätzlich die Zeile der Konfigurationsdatei aus, die die Ursache des Problems zu sein scheint. Bei Fehlern wird Icinga oft sofort die Überprüfung beenden und bereits nach der Ausgabe des ersten Fehlers zur Kommandozeile zurückkehren. Das geschieht, weil dieser erste Fehler im Laufe der restlichen Konfigurationdatei(en) weitere Fehler nach sich ziehen könnte. Sobald Sie Fehlermeldungen bekommen, müssen Sie Ihre Konfigurationsdateien editieren, um das Problem zu beheben. Warnungen können *generell* problemlos ignoriert werden, weil sie lediglich Empfehlungen darstellen und keine Erfordernisse für den Betrieb.

Anstatt die Pfade für das Binary und die Konfigurationsdatei anzugeben können Sie auch den folgenden Befehl eingeben:

```
#> /etc/init.d/icinga checkconfig
```

Die Ausführung ergibt einen Return-Code ungleich Null, wenn Ihre Konfiguration Fehler enthält. Das kann sinnvoll sein, wenn Sie Icinga automatisch erneut starten wollen..

Wenn Sie stattdessen den folgenden Befehl eingeben

```
#> /etc/init.d/icinga show-errors
```

dann wird eine Datei mit den gefundenen Fehlern erstellt. Falls es Fehler gibt, wird der Inhalt der Datei angezeigt ("show-errors" ab Icinga 1.0.2).

Sobald Sie Ihre Konfigurationsdateien überprüft und eventuelle Fehler bereinigt haben, können Sie [Icinga \(neu\)starten](#)

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Kapitel 4. Icinga
starten/stoppen/prüfen

[Zum Anfang](#)

Icinga starten und stoppen

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Icinga starten und stoppen

[Zurück](#)

[Kapitel 4. Icinga starten/stoppen/prüfen](#)

[Weiter](#)

Icinga starten und stoppen

Es gibt mehr als einen Weg, um Icinga zu starten, zu stoppen und erneut zu starten. Hier einige der üblichen...



Hinweis: Stellen Sie immer sicher, dass Sie Ihre [Konfiguration überprüfen](#), bevor Sie Icinga (neu)starten.

Icinga starten

1. **Init-Script:** Der einfachste Weg, den Icinga-Daemon zu starten, ist die Nutzung des Init-Scripts:

```
#> /etc/rc.d/init.d/icinga start
```

2. **manuell:** Sie können Icinga manuell mit der **-d**-Kommandozeilenoption wie folgt starten:

```
#> /usr/local/icinga/bin/icinga -d /usr/local/icinga/etc/icinga.cfg
```

3. **Debug Modus:** Im ziemlich seltenen Fall, dass Icinga sich still ohne Hinweise in den verschiedenen Log-Dateien beendet, können Sie Icinga durch das Weglassen der Daemon-Option starten:

```
#> /usr/local/icinga/bin/icinga /usr/local/icinga/etc/icinga.cfg
```

Auf diese Weise wird es im Vordergrund gestartet, so dass eine Menge von Meldungen über den Bildschirm laufen, aber es könnte zu einem Hinweis ganz am Ende führen.

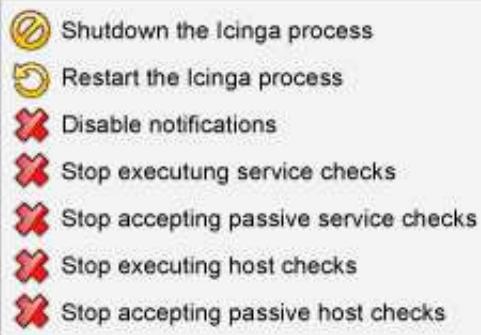
Icinga erneut starten

Ein Neustart bzw. neu laden ist notwendig, wenn Sie Ihre Konfigurationsdateien verändert haben und diese Änderungen aktiv werden sollen.

1. **Init-Script:** Der einfachste Weg, den Icinga-Daemon neu zu starten, ist die Nutzung des Init-Scripts:

```
#> /etc/rc.d/init.d/icinga reload
```

2. **Web-Interface:** Sie können Icinga mit Hilfe des Web-Interfaces neu starten. Klicken Sie auf den "Process Info"-Navigations-Link und wählen Sie "Restart the Icinga process":



3. **manuell:** Sie können den Icinga-Prozess durch Senden eines SIGHUP-Signals neu starten:

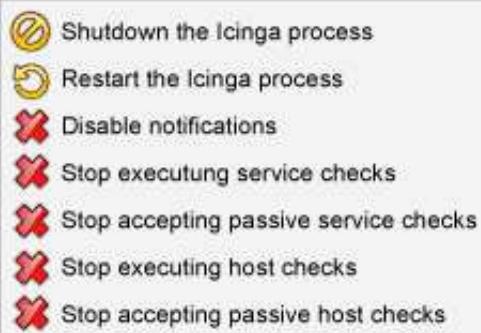
```
#> kill -HUP <icinga_pid>
```

Icinga stoppen

1. **Init-Script:** Der einfachste Weg, den Icinga-Daemon zu stoppen, ist die Nutzung des Init-Script:

```
#> /etc/rc.d/init.d/icinga stop
```

2. **Web-Interface:** Sie können Icinga mit Hilfe des Web-Interfaces stoppen. Klicken Sie auf den "Process Info"-Navigations-Link und wählen Sie "Shutdown the Icinga process":



3. **manuell:** Sie können den Icinga-Prozess durch Senden eines SIGTERM-Signals stoppen:

```
#> kill <icinga_pid>
```

Protokoll-Einstellungen in /usr/local/icinga/etc/icinga.cfg

Einstellungen für den Daemon:

Definiert ob Nachrichten in die Daemon Logdatei geschrieben werden sollen (gewöhnlich nach icinga.log). Der Standardwert ist 1 (ja), setzen des Wertes auf 0 (nein) verhindert den Log der Nachrichten.

```
use_daemon_log=0/1
```

Einstellungen für den Syslog-Dienst:

Wenn Sie möchten, dass Icinga Meldungen an den Syslog-Dienst übergibt, setzen Sie diese Einstellung auf 1.

use_syslog=0/1

Diese Option kann zusätzlich zur use_daemon_log-Option verwendet werden.

Verschiedene Optionen

Falls Sie große Verzögerungen zwischen dem Start von Icinga und den ersten Prüfungen feststellen, dann gibt es verschiedene Optionen, die [hier](#) näher beschrieben sind. Dort finden Sie auch die Option -S, die nähere Informationen zur Scheduling Queue ausgibt.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Überprüfen Ihrer
Icinga-Konfiguration

[Zum Anfang](#)

Kapitel 5. Die Grundlagen

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Kapitel 5. Die Grundlagen

[Zurück](#)

[Weiter](#)

Kapitel 5. Die Grundlagen

Inhaltsverzeichnis

Icinga Plugins

Makros verstehen und wie sie arbeiten

Standard-Makros in Icinga

Host-Prüfungen (Host checks)

Service-Prüfungen (Service Checks)

Aktive Prüfungen (Active Checks)

Passive Prüfungen (Passive Checks)

Statusarten

Zeitfenster

Ermitteln des Zustands und der Erreichbarkeit von Netzwerk-Hosts

Benachrichtigungen

[Zurück](#)

[Weiter](#)

[Icinga starten und stoppen](#)

[Zum Anfang](#)

[Icinga Plugins](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Icinga Plugins

[Zurück](#)
[Kapitel 5. Die Grundlagen](#)
[Weiter](#)

Icinga Plugins

Einführung

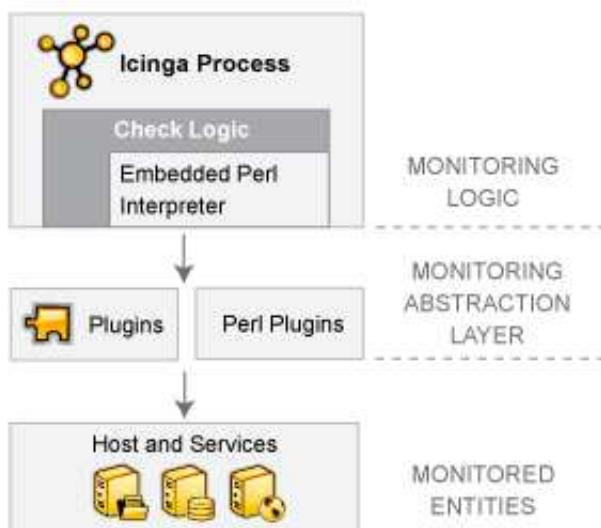
Icinga enthält nicht, wie viele andere Überwachungs-Tools, interne Mechanismen zur Prüfung des Zustands von Hosts und Services in Ihrem Netzwerk. Icinga verlässt sich statt dessen auf externe Programme (Plugins genannt), die all die schmutzige Arbeit tun.

Was sind Plugins?

Plugins sind kompilierte Programme oder Scripts (Perl-Scripts, Shell-Scripts, usw.), die von einer Kommandozeile aus laufen können, um den Status eines Hosts oder Service zu prüfen. Icinga benutzt die Ergebnisse von Plugins, um den aktuellen Status von Hosts oder Services in Ihrem Netzwerk zu ermitteln.

Icinga wird ein Plugin immer dann ausführen, wenn die Notwendigkeit besteht, den Status eines Hosts oder Service zu prüfen. Das Plugin tut *etwas* (beachten Sie den sehr allgemeinen Ausdruck), um die Prüfung auszuführen und dann einfach die Ergebnisse an Icinga zurückzuliefern. Icinga wird die Ergebnisse verarbeiten, die es vom Plugin erhält, und dann notwendige Aktionen ausführen (starten von [Eventhandlern](#), senden von [Benachrichtigungen](#), etc.).

Plugins als eine Abstraktionsschicht



Plugins arbeiten wie eine Abstraktionsschicht zwischen der Überwachungslogik im Icinga-Dämon und den eigentlichen Services und Hosts, die überwacht werden.

Der Vorteil dieses Typs von Plugin-Architektur ist, dass Sie fast alles überwachen können, was Ihnen einfällt. Wenn Sie den Prozess der Überwachung automatisieren können, können Sie es mit Icinga überwachen. Es gibt bereits eine Menge von Plugins, die erzeugt wurden, um grundlegende Ressourcen wie z.B. Prozessorauslastung, Plattenbelegung, Ping-Raten usw. zu überwachen. Wenn Sie etwas anderes überwachen möchten, werfen Sie einen Blick in die Dokumentation zu [Plugins schreiben](#) und erstellen Sie ein eigenes. Es ist einfach!

Der Nachteil dieses Typs von Plugin-Architektur ist die Tatsache, dass Icinga absolut keine Ahnung davon hat, was Sie überwachen. Sie könnten Netzwerkverkehr-Statistiken, Datenfehler-Raten, Raumtemperatur, CPU-Spannung, Lüftergeschwindigkeit, Prozessorauslastung, Plattenbelegung überwachen oder die Fähigkeit Ihres superphantastischen Toasters, am Morgen Ihr Brot ordnungsgemäß zu bräunen... Icinga versteht nicht die Besonderheiten dessen, was überwacht wird - es verfolgt lediglich Veränderungen des *Zustands* dieser Ressourcen. Nur die Plugins selbst wissen genau, was sie überwachen und wie die eigentlichen Prüfungen auszuführen sind.

Welche Plugins sind verfügbar?

Es gibt bereits zahlreiche Plugins, um viele verschiedene Arten von Geräten und Services zu überwachen, u.a.:

- HTTP, POP3, IMAP, FTP, SSH, DHCP
- CPU-Auslastung, Plattenbelegung, Speicherauslastung, Anzahl Benutzer
- Unix/Linux, Windows- und Netware-Server
- Router und Switches
- etc.

Plugins beschaffen

Plugins werden nicht mit Icinga verteilt, aber Sie finden die offiziellen Nagios-Plugins zum Download und viele weitere Plugins, die von Nagios-Benutzern erstellt und gewartet werden, an folgenden Stellen:

- Nagios Plugins Project: <http://sourceforge.net/projects/nagiosplug>
- Nagios Downloads Page: <http://www.nagios.org//download/>
- MonitoringExchange: <http://www.monitoringexchange.org>

Wie benutze ich Plugin X?

Fast alle Plugins zeigen grundlegende Bedienungshinweise an, wenn sie von der Kommandozeile mit der Option '-h' oder '--help' aufgerufen werden. Wenn Sie z.B. wissen möchten, wie das Plugins check_http arbeitet bzw. welche Optionen es akzeptiert, sollten Sie folgenden Befehl ausprobieren:

```
$> ./check_http --help
```



Wichtig

Führen Sie Plugins immer mit dem Icinga-Benutzer aus, denn einige Plugins erstellen temporäre Dateien. Wenn Sie Plugins mit einem anderen Benutzer ausführen, dann kann der Icinga-Benutzer diese Dateien ggf. nicht überschreiben.

Rufen Sie das Plugin nicht mit einem relativen Pfad auf (z.B. `./check_test_plugin`). Benutzen Sie immer absolute Pfade, denn so macht es auch Icinga (z.B. `/usr/local/icinga/libexec/check_test_plugin`).

Integration eines neuen Plugins

Wenn Sie ein neues Plugin integrieren möchten, dann lesen Sie die Dokumentation (falls vorhanden). Sie könnte wichtige Informationen über die Voraussetzungen wie z.B. zusätzliche Pakete oder (Perl-) Module enthalten, wie das Plugin zu installieren ist bzw. distributionsabhängige Hinweise.

Manchmal müssen Sie das Plugin kompilieren, wobei Sie den Vorgang durch den Aufruf von "`./configure`" mit oder ohne Optionen vorbereiten. Bitte prüfen Sie die Datei `config.log` auf mögliche Fehler zu fehlenden (devel-)Paketen vor dem Aufuf des eigentlichen Compile-Vorgangs (meistens "make" oder "make all"). In den meisten Fällen wird das Plugin durch den Aufruf von "make install" in das Plugins-Verzeichnis (z.B. `/usr/local/icinga/libexec`) kopiert.

Manchmal müssen Sie das Plugin auf Ihre Umgebung anpassen (z.B. den Pfad zu "utils.pm").

Nach der Installation des Plugins rufen Sie es mit den nötigen Optionen von der Kommandozeile aus auf. Wenn dies funktioniert, können Sie es in Icinga integrieren.

Stellen Sie sich vor, dass Sie den folgenden Aufruf benutzt haben:

```
/usr/local/icinga/libexec/sample-plugin.pl -H 192.168.1.2 -a argument1 -p parameter -n 5
```

Die command-Definition enthält zwei Direktiven

- `command_name`: dies ist ein Kurzname, der den Befehl identifiziert. Lassen Sie uns `check_sample` benutzen
- `command_line`: hier definieren Sie den auszuführenden Befehl. Sie könnten den Befehl angeben, den Sie auf der Kommandozeile benutzen, aber das wäre zu unflexibel. Normalerweise ändert sich das Plugin-Verzeichnis (`/usr/local/icinga/libexec`) nicht, so dass wir eine `$USERn$`-Variable benutzen können, die in der `resource.cfg` definiert werden. Die IP-Adresse ändert sich von Host zu Host. Es gibt das Makro `$HOSTADDRESS$`, das wir dafür nutzen können. Die Werte der Optionen können sich ändern, so dass auch sie flexibel sein sollten. Das könnte zu folgender Definition führen:

```
define command{
    command_name check_sample
    command_line $USER1$/sample-plugin.pl -H $HOSTADDRESS$ -a $ARG1$ -p $ARG2$ -n $ARG3$
}
```

Dann müssen wir die `check_command`-Direktive definieren, die Teil der Host-/Service-Definition ist. Es beginnt mit dem Kurznamen gefolgt von den Argumenten, die jeweils durch Ausrufezeichen voneinander getrennt sind:

```
check_command check_sample!argument1!parameter!5
```

Wie Sie sehen, wird die IP-Adresse nicht angegeben, denn sie wird aus der Host-Definition genommen.

Neben den bereits genannten gibt es eine Vielzahl von [Makros](#), die die Arbeit erleichtern. Dabei gibt es einige Dinge anzumerken:

- Alle Icinga-Makros benutzen Großbuchstaben und werden in Dollarzeichen (\$) eingeschlossen
- Die meisten Makros haben einen bestimmten Gültigkeitsbereich. Wenn Sie versuchen, ein Makro außerhalb dieses Bereichs zu nutzen, dann werden Sie statt des erwarteten Wertes lediglich ein Dollarzeichen (\$) sehen
- Die [\\$USERn\\$](#)-Makros können genutzt werden, um sensible Informationen wie z.B. Passwörter zu "verstecken", denn die Werte werden im Gegensatz zu den anderen Makros nicht im Web-Interface angezeigt. Außerdem können sie verwendet werden, um bestimmte Sonderzeichen nutzen zu können, die anderenfalls zu Schwierigkeiten führen. Ein Beispiel wäre `USER99=;`. Auf diese Weise können Sie ein Semikolon benutzen, das sonst als Start eines Kommentars in Ihrer Definition behandelt würde
- Nachdem es bei deutschsprachigen Personen oftmals zu Problemen kommt: [\\$HOSTADDRESS\\$](#) wird mit zwei "D" geschrieben

Schwellwert und Bereiche

Einige Plugins unterstützen Bereichsangaben für die Warn- und Kritisch-Werte. Bitte überprüfen Sie die Dokumentation, ob das der Fall für das Plugin ist, das Sie benutzen möchten. Das Folgende ist ein Auszug der (englischsprachigen) [Entwickler-Richtlinien](#):

Ein Bereich ist definiert als ein Start- und Endpunkt (inklusive) auf einer numerischen Skala (ggf. bis zu +/--Unendlich).

Ein Schwellwert ist ein Bereich mit einem Alarmpegel (entweder Warning oder Critical).

In der Theorie wird das Plugin eine Prüfung durchführen, die einen numerischen Wert oder eine Metrik zurückliefert, die dann mit den Warning- und Critical-Schwellwerten verglichen wird

Dies ist das generelle Format für Bereiche:

`[@]start:end`

Anmerkungen:

1. start = end, falls :end nicht angegeben ist
2. start und ":" ist nicht erforderlich, wenn start=0
3. falls der Bereich vom Format "start:" ist und end nicht angegeben wurde, dann ist das Ende als +Unendlich anzunehmen
4. um -Unendlich anzugeben, benutzen Sie "~"
5. Alarm erfolgt, wenn die Metrik außerhalb des durch Start- und Ende angegebenen Bereichs liegt (Endpunkte gehören *nicht* zum Bereich)

6. wenn der Bereich mit "@" beginnt, dann ist zu alarmieren, wenn die Metrik innerhalb des Bereichs liegt (einschließlich der Endpunkte)



Anmerkung

Nicht alle Plugin unterstützen (bisher) die Bereichsnotation.

Beispiele

Bereichsdefinition	Alarm, wenn x...
10	< 0 oder > 10, (außerhalb des Bereichs von {0 .. 10})
10:	< 10, (außerhalb {10 .. Unendlich})
~:10	> 10, (außerhalb des Bereichs von {-Unendlich .. 10})
10:20	< 10 oder > 20, (außerhalb des Bereichs von {10 .. 20})
@10:20	<= 10 and >= 20, (im Bereich von {10 .. 20})

Kommandozeilenbeispiele

Kommandozeile	Erklärung
check_stuff -w10 -c20	kritisch, wenn "stuff" größer als 20, andernfalls warnen, wenn größer als 10 (außerdem kritisch, wenn "stuff" kleiner als 0)
check_stuff -w~:10 -c~:20	das Gleiche wie oben, allerdings ist "stuff" kleiner als Null OK!
check_stuff -w10: -c20	kritisch, wenn "stuff" größer als 20, andernfalls warnen, wenn "stuff" kleiner als 10 (außerdem kritisch, wenn "stuff" kleiner als 0)
check_stuff -c1:	kritisch, wenn "stuff" kleiner als 1
check_stuff -w~:0 -c10	kritisch, wenn "stuff" größer als 10; warnen, wenn "stuff" größer als 0
check_stuff -c5:6	der einzige nicht-kritische Bereich ist 5:6
check_stuff -c@10:20	kritisch, wenn "stuff" zwischen 10 und 20 [1]
check_stuff -w20:30 -c10:40	warnen, wenn "stuff" kleiner als 20 oder größer als 30, kritisch, wenn "stuff" kleiner als 10 oder größer als 40 [2]



Anmerkung

[1]: Bei der Kommandozeile in den Entwickler-Richtlinien fehlt "@", anderenfalls wäre die Erklärung falsch (und es gäbe kein Beispiel für die @-Notation)

[2]: Bitte beachten Sie, dass das letzte Beispiel geschachtelte Bereiche benutzt. Das mag nicht bei allen Plugins funktionieren, die Bereichsangaben unterstützen. Es wurde mit check_snmp getestet

Aktivieren der Definition

Prüfen Sie die Konfiguration mit "/etc/init.d/icinga show-errors" und bereinigen Sie eventuelle Fehler, bevor Sie Icinga mit "/etc/init.d/icinga restart" neu starten. Warten Sie, bis das Objekt geprüft wurde und betrachten Sie die Status-Details. Vielleicht gibt es Fehler.

- "...resulted in a return code of 127"

Das bedeutet, dass das Plugin nicht an der angegebenen Position gefunden wurde oder innerhalb des Plugins eine Datei aufgerufen wurde, die nicht gefunden wurde. Wenn Sie \$USERn\$-Makros beim Aufruf des Plugins benutzen, dann stellen Sie sicher, dass das Makros wirklich auf die Position verweist, wo das Plugin zu finden ist (ist das Makro in resource.cfg definiert?). Benachrichtigungsbefehle rufen oft ein Mail-Programm auf. Stellen Sie sicher, dass der Pfad zum Mail-Programm korrekt ist.

- "...resulted in a return code of 13"

Meistens handelt es sich um ein Berechtigungsproblem. Der Benutzer kann ggf. das Plugin nicht ausführen bzw. darauf und/oder auf zugehörige Dateien zugreifen. Das kann passieren, wenn Sie als root ein Plugin ausgeführt haben, das temporäre Dateien anlegt. Der Icinga-Benutzer ist nicht berechtigt, diese Dateien zu überschreiben.

Plugin API

Informationen zu technischen Aspekten von Plugins sowie zur Erstellung Ihrer eigenen Plugins finden Sie [hier](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Kapitel 5. Die Grundlagen

[Zum Anfang](#)

Makros verstehen und wie sie arbeiten



Makros verstehen und wie sie arbeiten

[Zurück](#)
[Kapitel 5. Die Grundlagen](#)
[Weiter](#)

Makros verstehen und wie sie arbeiten

Makros

Eine der Haupteigenschaften, die Icinga so flexibel machen, ist die Fähigkeit, Makros in Befehlsdefinitionen zu benutzen. Makros erlauben Ihnen, Bezug auf Informationen von Hosts, Services und anderen Quellen zu nehmen.

Makroersetzung - wie Makros arbeiten

Bevor Icinga einen Befehl ausführt, ersetzt es jedes Makro, das es in der Befehlsdefinition findet, durch den entsprechenden Wert. Diese Makroersetzung erfolgt für alle Arten von Befehlen, die Icinga ausführt - Host- und Service-Checks, Benachrichtigungen, Eventhandler usw.

Bestimmte Makros können wieder Makros enthalten. Dazu zählen die Makros \$HOSTNOTES\$, \$HOSTNOTESURL\$, \$HOSTACTIONURL\$, \$SERVICENOTES\$, \$SERVICENOTESURL\$ und \$SERVICEACTIONURL\$.

Beispiel 1: Host-Address Makro

Wenn Sie Host- und Service-Makros in Befehlsdefinitionen benutzen, beziehen sich diese auf Werte für den Host oder Service, für den der Befehl ausgeführt wird. Nehmen wir ein Beispiel. Angenommen, wir benutzen eine Host-Definition und einen *check_ping*-Befehl, die wie folgt definiert sind:

```
define host{
    host_name      linuxbox
    address        192.168.1.2
    check_command   check_ping
    ...
}
define command{
    command_name   check_ping
    command_line   /usr/local/icinga/libexec/check_ping -H $HOSTADDRESS$ -w 100.0,90% -c 200.0,60%
```

die erweiterte/endgültige auszuführende Befehlszeile für die Host-Prüfung würde so aussehen:

```
$> /usr/local/icinga/libexec/check_ping -H 192.168.1.2 -w 100.0,90% -c 200.0,60%
```

Ziemlich einfach, stimmt's? Die Schönheit liegt darin, dass Sie eine einzelne Befehlsdefinition für eine unbegrenzte Zahl von Hosts nutzen können. Jeder Host kann mit der selben Befehlsdefinition geprüft werden, weil jede Host-Adresse automatisch vor der Ausführung in der Befehlszeile ersetzt wird.

Beispiel 2: Befehlsargument-Makros

Sie können auch Argumente an Befehle übergeben, was recht handlich ist, wenn Sie Ihre Befehlsdefinitionen ziemlich generisch halten möchten. Argumente werden in der Objektdefinition (d.h. Host oder Service) angegeben, indem sie durch Ausrufezeichen (!) vom Befehlsnamen getrennt werden:

```
define service{
    host_name          linuxbox
    service_description PING
    check_command      check_ping!200.0,80%!400.0,40%
    ...
}
```

Im obigen Beispiel hat der Service-Check zwei Argumente (auf die mit **\$ARGn\$**-Makros zugegriffen werden kann). Das **\$ARG1\$**-Makro wird "200.0,80%" und **\$ARG2\$** wird "400.0,40%" (beide ohne Anführungszeichen). Angenommen, wir benutzen die vorher angegebene Host-Definition und einen wie folgt definierten *check_ping*-Befehl:

```
define command{
    command_name      check_ping
    command_line      /usr/local/icinga/libexec/check_ping -H $HOSTADDRESS$ -w $ARG1$ -c $ARG2$
```

die erweiterte/endgültige auszuführende Befehlszeile für die Service-Prüfung würde so aussehen:

```
$> /usr/local/icinga/libexec/check_ping -H 192.168.1.2 -w 200.0,80% -c 400.0,40%
```



Hinweis: Falls Sie Ausrufezeichen (!) in Ihren Argumenten übergeben müssen, dann können Sie das tun, indem Sie diese mit einem Backslash (\) maskieren. Falls Sie Backslashes in Ihren Argumenten einsetzen müssen, sind diese ebenfalls mit Backslashes zu maskieren.

On-Demand-Makros

Wenn Sie Host- und Service-Makros in Ihren Befehlsdefinitionen benutzen, dann beziehen sie sich normalerweise auf Werte des Hosts oder Service, für den der Befehl ausgeführt wird. Wenn beispielsweise eine Host-Prüfung für einen Host namens "linuxbox" ausgeführt wird, werden sich all die **Standard-Host-Makros** auf Werte für diesen Host beziehen ("linuxbox").

Wenn Sie möchten, dass sich die Werte eines Befehls auf einen anderen Host oder Service beziehen (für den der Befehl nicht ausgeführt wird), dann können Sie die sogenannten "On-Demand-Makros" benutzen. On-Demand-Makros sehen wie normale Makros aus, außer der Tatsache, dass sie einen Bezeichner für den Host oder Service enthalten, von dem sie ihren Wert erhalten sollen. Hier das grundsätzliche Format von On-Demand-Makros:

- **\$HOSTMACRONAME:host_name\$**
- **\$SERVICEMACRONAME:host_name:service_description\$**

Ersetzen Sie **HOSTMACRONAME** und **SERVICEMACRONAME** durch den Namen eines der Standard-Host- oder Service-Makros, die [hier](#) zu finden sind.

Beachten Sie, dass der Makroname durch einen Doppelpunkt (:) vom Host- oder Service-Bezeichner getrennt ist. Für On-Demand-Service-Makros besteht der Service-Bezeichner aus einem Host-Namen und einer Service-Beschreibung - sie sind ebenfalls durch einen Doppelpunkt (:) voneinander getrennt.



Hinweis: On-Demand-Service-Makros können ein leeres Host-Namen-Feld enthalten. In diesem Fall wird automatisch der Name des Hosts benutzt, der mit dem Service verbunden ist.

Beispiele für On-Demand-Host- und Service-Makros folgen:

```
$HOSTDOWNTIME:myhost$           <--- On-Demand-Host-Makro
$SERVICESTATEID:novellserver:DS Database$ <--- On-Demand-Service-Makro
$SERVICESTATEID::CPU Load$      <--- On-Demand-Service-Makro mit leerem Host-Namen-Feld
```

On-Demand-Makros gibt es auch für hostgroup-, servicegroup-, contact- und contactgroup-Makros. Zum Beispiel:

```
$CONTACTEMAIL:john$           <--- On-Demand-Contact-Makro
$CONTACTGROUPMEMBERS:linux-admins$ <--- On-Demand-Contactgroup-Makro
$HOSTGROUPALIAS:linux-servers$    <--- On-Demand-Hostgroup-Makro
$SERVICEGROUPALIAS:DNS-Cluster$   <--- On-Demand-Servicegroup-Makro
```

On-Demand-Gruppen-Makros

Sie können die Werte eines Makros über alle Kontakte, Hosts oder Services in einer bestimmten Gruppe mit einem speziellen Format Ihrer On-Demand-Makrodeklaration erhalten. Sie tun dies, indem Sie auf eine bestimmte Hostgruppe, Servicegruppe oder Kontaktgruppe in einem On-Demand-Makro verweisen und zwar wie folgt:

- \$HOSTMACRONAME:*hostgroup_name:delimiter\$*
- \$SERVICEMACRONAME:*servicegroup_name:delimiter\$*
- \$CONTACTMACRONAME:*contactgroup_name:delimiter\$*

Ersetzen Sie *HOSTMACRONAME*, *SERVICEMACRONAME* und *CONTACTMACRONAME* durch den Namen eines der Standard-Host-, Service- oder Kontaktmakros, die Sie [hier](#) finden. Der Begrenzer (delimiter), den Sie angeben, wird benutzt, um Makrowerte der einzelnen Gruppenmitglieder von einander zu trennen.

Das folgende Makro wird beispielsweise eine komma-separierte Liste von Host-Status-IDs zurückliefern, die Mitglieder der *hg1*-Hostgruppe sind:

```
$HOSTSTATEID:hg1:, $
```

Diese Makrodefinition wird etwas zurückliefern, was etwa so aussieht:

```
0,2,1,1,0,0,2
```

Benutzervariablen-Makros

Jede [Benutzerobjekt-Variable](#), die Sie in Host-, Service- oder Contact-Definitionen einsetzen, ist auch in Makros verfügbar. Benutzervariablen-Makros werden wie folgt benannt:

- \$_HOST*varname\$*
- \$_SERVICE*varname\$*
- \$_CONTACT*varname\$*

Nehmen Sie die folgende Host-Definition mit einer "*_MACADDRESS*" genannten Benutzervariablen...

```
define host{
    host_name      linuxbox
    address        192.168.1.1
    _MACADDRESS    00:01:02:03:04:05
    ...
}
```

Die Benutzervariable `_MACADDRESS` wäre in einem Makro `$HOSTMACADDRESS$` verfügbar. Weitere Informationen zu Benutzervariablen und wie sie in Makros eingesetzt werden können, finden Sie [hier](#).

Makrobereinigung

Einige Makros werden von potenziell gefährlichen Shell-Metazeichen bereinigt, bevor Ersetzungen in Befehlen stattfinden. Welche Zeichen aus den Makros entfernt werden, hängt von den Einstellungen der `illegal_macro_output_chars`-Direktive ab. Die folgenden Makros werden von potenziell gefährlichen Zeichen bereinigt:

1. `$HOSTOUTPUT$`
2. `$LONGHOSTOUTPUT$`
3. `$HOSTPERFDATA$`
4. `$HOSTACKAUTHOR$`
5. `$HOSTACKCOMMENT$`
6. `$SERVICEOUTPUT$`
7. `$LONGSERVICEOUTPUT$`
8. `$SERVICEPERFDATA$`
9. `$SERVICEACKAUTHOR$`
10. `$SERVICEACKCOMMENT$`

Makros als Umgebungsvariablen

Die meisten Makros werden als Umgebungsvariablen zur Verfügung gestellt, um einen einfachen Einsatz in Scripts oder Befehlen zu ermöglichen, die von Icinga ausgeführt werden. Aus Gründen der Sicherheit und der Vernunft werden `$USERn$` und "on-demand" Host- und Service-Makros nicht als Umgebungsvariablen zur Verfügung gestellt.

Umgebungsvariablen, die Standard-Makros enthalten, werden ebenso wie ihre entsprechenden Makronamen benannt ([hier](#) aufgeführt), wobei ihnen "ICINGA_" vorangestellt wird. Beispielsweise wäre das `$HOSTNAME$`-Makro als Umgebungsvariable "ICINGA_HOSTNAME" verfügbar.

Verfügbare Makros

Eine Liste aller in Icinga verfügbaren Makros sowie eine Tabelle, wann sie eingesetzt werden können, finden Sie [hier](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Icinga Plugins](#)

[Zum Anfang](#)

[Standard-Makros in Icinga](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Standard-Makros in Icinga

[Zurück](#)[Kapitel 5. Die Grundlagen](#)[Weiter](#)

Standard-Makros in Icinga

In Icinga verfügbare Standard-Makros sind hier aufgelistet. On-Demand-Makros und Makros für Benutzervariablen sind [hier](#) beschrieben.

Makro-Geltungsbereich

Obwohl Makros in allen Befehlen benutzt werden können, die Sie definieren, sind sie ggf. nicht "gültig" innerhalb eines bestimmten Befehlstyps. Zum Beispiel sind einige Makros vielleicht nur gültig bei Service-Benachrichtigungen, andere vielleicht nur bei Host-Prüfungen. Es gibt zehn Arten von Befehlen, die Icinga erkennt und unterschiedlich behandelt. Dies sind:

1. Service-Prüfungen
2. Service-Benachrichtigungen
3. Host-Prüfungen
4. Host-Benachrichtigungen
5. Service-[Eventhandler](#) und/oder ein globaler Service-Eventhandler
6. Host-[Eventhandler](#) und/oder ein globaler Host-Eventhandler
7. [OCSP](#) Befehl
8. [OCHP](#) Befehl
9. Service-[Performance-Daten](#) Befehle
10. Host-[Performance-Daten](#) Befehle

Die nachfolgenden Aufstellungen enthalten alle aktuell in Icinga verfügbaren Makros zusammen mit einer kurzen Beschreibung und den Befehlstypen, in denen sie gelten. Wenn ein Makro in einem Befehl benutzt wird, in dem es nicht gültig ist, wird es durch eine leere Zeichenkette ersetzt. Es ist zu beachten, dass Makros aus Großbuchstaben bestehen und in Dollarzeichen (\$) eingeschlossen werden.

Makroverfügbarkeits-Aufstellung

Legende:

Nein	Das Makro ist nicht verfügbar
Ja	Das Makro ist verfügbar

Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Host-Makros: ³								
\$HOSTNAMES	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTDISPLAYNAME\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTALIASS	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTADDRESS	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTADDRESS6\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTSTATES	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTSTATEIDS	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LASTHOSTSTATES	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LASTHOSTSTATEIDS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTSTATETYPES	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTATTEMPTS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$MAXHOSTATTEMPTSS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTEVENTIDS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LASTHOSTEVENTIDS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTPROBLEMSIDS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LASTHOSTPROBLEMSIDS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTLATENCY\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTEXECUTIONTIMES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTDURATIONS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTDURATIONSECS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTDOWNTIMES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTPERCENTCHANGES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTGROUPNAMES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTGROUPNAMESS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LASTHOSTCHECKS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LASTHOSTSTATECHANGES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LASTHOSTUPS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LASTHOSTDOWNS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LASTHOSTUNREACHABLES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTOUTPUTS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LONGHOSTOUTPUTS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTPERFDATAS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTCHECKCOMMANDS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTACKAUTHORS ⁸	Nein	Nein	Nein	Ja	Nein	Nein	Nein	Nein
\$HOSTACKAUTHOR NAMES ⁸	Nein	Nein	Nein	Ja	Nein	Nein	Nein	Nein
\$HOSTACKAUTHORALIASS ⁸	Nein	Nein	Nein	Ja	Nein	Nein	Nein	Nein
\$HOSTACKCOMMENTS ⁸	Nein	Nein	Nein	Ja	Nein	Nein	Nein	Nein
\$HOSTACTIONURLS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTNOTESURLS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTNOTESS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$TOTALHOSTSERVICES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$TOTALHOSTSERVICESOKS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$TOTALHOSTSERVICESWARNINGS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$TOTALHOSTSERVICESUNKNOWN\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$TOTALHOSTSERVICESCRITICALS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Hostgroup-Makros:								
\$HOSTGROUPALIASS ⁵	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTGROUPMEMBERS ⁵	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTGROUPNOTESS ⁵	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTGROUPNOTESURLS ⁵	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTGROUPACTIONURLS ⁵	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Service-Makros:								
\$SERVICEDESCS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEDISPLAYNAMES\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICESTATES\$	Ja ³	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICESTATEIDS\$	Ja ²	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LASTSERVICESTATES\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LASTSERVICESTATEIDS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICESTATETYPES\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEATTEMPTS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$MAXSERVICEATTEMPTSS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEISVOLATILES\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEEVENTIDS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LASTSERVICEEVENTIDS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEPROBLEMSIDS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein

\$LASTSERVICEPROBLEMSID\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICELATENCY\$	Ja	ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEEXECUTIONTIMES	Ja ²	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEDURATIONS	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEDURATIONSECS	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEDOWNTIMES	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEPERCENTCHANGES\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEGROUPNAMES\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEGROUPNAMESS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LASTSERVICECHECKS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LASTSERVICESTATECHANGES\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LASTSERVICEOKS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LASTSERVICEWARNINGS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LASTSERVICEUNKNOWN\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LASTSERVICECRITICALS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEOUTPUT\$	Ja ²	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$LONGSERVICEOUTPUTS\$	Ja ²	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEPERFDATAS\$	Ja ²	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICECHECKCOMMAND\$	Ja	ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICEACKAUTHORS ⁸	Nein	Ja	Nein	Nein	Nein	Nein	Nein	Nein
\$SERVICEACKAUTHORNAME\$ ⁸	Nein	Ja	Nein	Nein	Nein	Nein	Nein	Nein
\$SERVICEACKAUTHORALIAS\$ ⁸	Nein	Ja	Nein	Nein	Nein	Nein	Nein	Nein
\$SERVICEACKCOMMENTS ⁸	Nein	Ja	Nein	Nein	Nein	Nein	Nein	Nein
\$SERVICEACTIONURLS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICENOTESURLS\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
\$SERVICENOTES\$	Ja	Ja	Nein	Nein	Ja	Nein	Ja	Nein
<hr/>								
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Servicegroup-Makros:								
\$SERVICEGROUPALIASS ⁶	Ja	ja	Ja	Ja	Ja	Ja	Ja	Ja
\$SERVICEGROUPMEMBERS\$ ⁶	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$SERVICEGROUPNOTES\$ ⁶	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$SERVICEGROUPNOTESTURLS\$ ⁶	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$SERVICEGROUPACTIONURLS\$ ⁶	Ja	ja	Ja	Ja	Ja	Ja	Ja	Ja
<hr/>								
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Contact-Makros:								
\$CONTACTNAMES\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$CONTACTALIASS\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$CONTACTEMAILS\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$CONTACTPAGERS\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$CONTACTADDRESS\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
<hr/>								
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Contactgroup-Makros:								
\$CONTACTGROUPALIASS ⁷	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$CONTACTGROUPMEMBERS\$ ⁷	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<hr/>								
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Auswertungsmakros:								
\$TOTALHOSTSUP\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALHOSTSDOWN\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALHOSTSUNREACHABLES\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALHOSTSDOWNUNHANDLED\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALHOSTSUNREACHABLEUNHANDLED\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALHOSTPROBLEMSS\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALHOSTPROBLEMSUNHANDLED\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALSERVICESOKS\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALSERVICESWARNINGS\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALSERVICESCRITICALS\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALSERVICESUNKNOWN\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALSERVICESWARNINGUNHANDLED\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALSERVICESCRITICALUNHANDLED\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALSERVICESUNKNOWNUNHANDLED\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALSERVICEPROBLEMS\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
\$TOTALSERVICEPROBLEMSUNHANDLED\$ ¹⁰	Ja	Ja ⁸	Ja	Ja ⁸	Ja	Ja	Ja	Ja
<hr/>								
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Benachrichtigungsmakros:								
\$NOTIFICATIONTYPES\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$NOTIFICATIONRECIPIENT\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$NOTIFICATIONISESCALATED\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$NOTIFICATIONAUTHORS\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein

\$NOTIFICATIONAUTHORNAME\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$NOTIFICATIONAUTHORIAS\$	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$NOTIFICATIONCOMMENTS	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$HOSTNOTIFICATIONNUMBERS	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$HOSTNOTIFICATIONIDS	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$SERVICENOTIFICATIONNUMBERS	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
\$SERVICENOTIFICATIONIDS	Nein	Ja	Nein	Ja	Nein	Nein	Nein	Nein
<hr/>								
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Datums-/Zeitmakros:								
\$LONGDATETIMES	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$SHORTDATETIMES	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$DATES	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$TIMES	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$TIMETS	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$ISVALIDTIME\$ ⁹	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$NEXTVALIDTIME\$ ⁹	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<hr/>								
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
Dateimakros:								
\$MAINCONFIGFILES	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$STATUSDATAFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$COMMENTDATAFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja ⁵
\$DOWNNTIMEDATAFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$RETENTIONDATAFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$OBJECTCACHEFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$TEMPFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$TEMPPATHS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$LOGFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$RESOURCEFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$COMMANDFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$HOSTPERDATAFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$SERVICEPERDATAFILES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<hr/>								
Makroname	Service-Prüfungen	Service-Benachrichtigungen	Host-Prüfungen	Host-Benachrichtigungen	Service-Eventhandler und OCSP	Host-Eventhandler und OCHP	Service-Perf-Daten	Host-Perf-Daten
verschiedene Makros:								
\$PROCESSSTARTTIMES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$EVENTSTARTTIMES\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$ADMINMAILS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$ADMINPAGERS\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$ARGn\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
\$USERn\$	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja

Makrobeschreibungen

Host-Makros: ³	
\$HOSTNAME\$	Kurzname für den Host (z.B. "biglinuxbox"). Dieser Wert wird aus der <i>host_name</i> -Direktive in der Host-Definition genommen.
\$HOSTDISPLAYNAME\$	Ein alternativer Anzeigename für den Host. Dieser Wert wird aus der <i>display_name</i> -Direktive in der Host-Definition genommen.
\$HOSTALIAS\$	Langname/Beschreibung für den Host. Dieser Wert wird aus der <i>alias</i> -Direktive in der Host-Definition genommen.
\$HOSTADDRESS\$	Adresse des Hosts. Dieser Wert wird aus der <i>address</i> -Direktive in der Host-Definition genommen.

\$HOSTADDRESS\$	Sekundäre/IPv6-Adresse des Hosts. Dieser Wert wird aus der <i>address6</i> -Direktive in der Host-Definition genommen.
\$HOSTSTATE\$	Eine Zeichenkette, die den aktuellen Status des Hosts angibt ("UP", "DOWN" oder "UNREACHABLE").
\$HOSTSTATEID\$	Eine Zahl, die dem aktuellen Status des Hosts entspricht: 0=UP, 1=DOWN, 2=UNREACHABLE.
\$LASTHOSTSTATE\$	Eine Zeichenkette, die den letzten Status des Hosts angibt ("UP", "DOWN" oder "UNREACHABLE").
\$LASTHOSTSTATEID\$	Eine Zahl, die dem letzten Status des Hosts entspricht: 0=UP, 1=DOWN, 2=UNREACHABLE.
\$HOSTSTATETYPE\$	Eine Zeichenkette, die den Statustyp der aktuellen Host-Prüfung angibt ("HARD" oder "SOFT"). Ein Soft-Status tritt auf, wenn eine Host-Prüfung einen nicht-OK (nicht-UP) Status zurückliefert und noch Wiederholungen anstehen. Ein Hard-Status liegt vor, wenn die Anzahl der Host-Prüfungs-Wiederholungen einen maximal definierten Wert erreicht hat.
\$HOSTATTEMPT\$	Die Anzahl der aktuellen Host-Prüfungs-Wiederholungen. Wenn dies beispielsweise das zweite Mal ist, dass der Host erneut geprüft wird, dann steht hier die Zahl zwei. Die aktuelle Wiederholungsanzahl ist eigentlich nur dann sinnvoll, wenn man Eventhandler für Soft-Zustände schreibt, die auf einer bestimmten Aktion für diese entsprechende Zahl basieren.
\$MAXHOSTATTEMPTS\$	Die max. Prüfversuche, wie sie für den aktuellen Host definiert sind. Nützlich, wenn man Host-Eventhandler für "Soft"-Zustände schreibt, die eine bestimmte Aktion ausführen basierend auf der Host-Wiederholungsanzahl.
\$HOSTEVENTID\$	Eine global eindeutige Zahl verbunden mit dem aktuellen Status des Hosts. Jedes Mal, wenn eine Host- (oder Service-) Statusänderung eintritt, wird eine globale Ereignis-ID-Nummer um eins (1) erhöht. Falls bei einem Host keine Statusänderung eintritt, wird dieses Makro auf Null (0) gesetzt.

\$LASTHOSTEVENTID\$	Die vorherige (global eindeutige) Ereigniszahl, die für den Host vergeben wurde.
\$HOSTPROBLEMID\$	Eine global eindeutige Zahl verbunden mit dem aktuellen Problemstatus des Hosts. Jedes Mal, wenn ein Host (oder Service) von einem UP- oder OK-Status in einen Problemzustand wechselt, wird eine globale Problem-ID um eins (1) erhöht. Dieses Makro wird ungleich Null sein, wenn der Host sich gerade in einem Zustand ungleich UP befindet. Statuswechsel zwischen Zuständen ungleich UP (z.B. DOWN oder UNREACHABLE) erhöhen diese Problem-ID nicht. Wenn sich der Host gerade in einem UP-Zustand befindet, wird dieses Makro auf Null (0) gesetzt. In Kombination mit Eventhandlern kann dieses Makro benutzt werden, um automatisch ein Trouble-Ticket zu eröffnen, wenn Hosts das erste Mal einen Problemzustand erreichen.
\$LASTHOSTPROBLEMID\$	Die vorherige (global eindeutige) Ereigniszahl, die für den Host vergeben wurde. In Kombination mit Eventhandlern kann dieses Makro benutzt werden, um automatisch ein Trouble-Ticket zu schließen, wenn Hosts in einen UP-Status zurückkehren.
\$HOSTLATENCY\$	Eine (Fließkomma-) Zahl, die die Anzahl von Sekunden angibt, um die eine <i>geplante</i> Host-Prüfung nach der eigentlichen Planungszeit stattfand. Wenn beispielsweise eine Prüfung für 03:14:15 geplant war und erst um 03:14:17 ausgeführt wurde, dann beträgt die Verzögerung 2.0 Sekunden. On-Demand-Host-Prüfungen haben eine Verzögerung von null Sekunden.
\$HOSTEXECUTIONTIME\$	Eine (Fließkomma-) Zahl, die die Dauer der Ausführung einer Host-Prüfung in Sekunden angibt.
\$HOSTDURATION\$	Eine Zeichenkette, die die Zeitdauer angibt, die sich der Host im aktuellen Status befindet. Das Format ist "XXh YYm ZZs" und gibt die Stunden, Minuten und Sekunden an.
\$HOSTDURATIONSEC\$	Eine Zahl, die die Zeitdauer in Sekunden angibt, die sich der Host im aktuellen Status befindet.

\$HOSTDOWNTIME\$	Eine Zahl, die die aktuelle "Downtime-Tiefe" für den Host angibt. Wenn dieser Host sich gerade in einer Phase einer geplanten Downtime befindet, ist dieser Wert größer als Null. Ist der Host nicht gerade in einer Downtime-Phase, ist dieser Wert Null.
\$HOSTPERCENTCHANGE\$	Eine (Fließkomma-) Zahl, die den prozentualen Statuswechsel angibt, dem der Host unterworfen war. Dieser Wert wird vom flap detection -Algorithmus benutzt.
\$HOSTGROUPNAME\$	Der Kurzname der Hostgruppe, zu der dieser Host gehört. Dieser Wert wird aus der <code>hostgroup_name</code> -Direktive in der hostgroup-Definition entnommen. Wenn der Host zu mehreren Hostgruppen gehört, enthält dieses Makro nur einen der Namen.
\$HOSTGROUPNAMES\$	Eine Komma-separierte Liste der Kurznamen aller Hostgruppen, zu denen dieser Host gehört.
\$LASTHOSTCHECK\$	Dieses ist ein Zeitstempel im time_t-Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt eine Prüfung des Hosts stattfand.
\$LASTHOSTSTATECHANGE\$	Dieses ist ein Zeitstempel im time_t-Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt ein Statuswechsel des Hosts stattfand.
\$LASTHOSTUP\$	Dieses ist ein Zeitstempel im time_t-Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt festgestellt wurde, dass sich der Host in einem UP-Zustand befand.
\$LASTHOSTDOWN\$	Dieses ist ein Zeitstempel im time_t-Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt festgestellt wurde, dass sich der Host in einem DOWN-Zustand befand.
\$LASTHOSTUNREACHABLE\$	Dieses ist ein Zeitstempel im time_t-Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt festgestellt wurde, dass sich der Host in einem UNREACHABLE-Zustand befand.
\$HOSTOUTPUT\$	Die erste Zeile von Textausgaben der letzten Host-Prüfung (z.B. "Ping OK")

\$LONGHOSTOUTPUT\$	Die vollständige Textausgabe (außer der ersten Zeile) der letzten Host-Prüfung.
\$HOSTPERFDATA\$	Dieses Makro enthält jegliche Performance-Daten , die von der letzten Host-Prüfung geliefert worden sein könnten.
\$HOSTCHECKCOMMAND\$	Dieses Makro enthält den Namen des Befehls (zusammen mit übergebenen Argumenten), der zur Host-Prüfung benutzt wurde.
\$HOSTACKAUTHOR\$ ⁸	Eine Zeichenkette, die den Namen des Benutzers enthält, der das Host-Problem bestätigt hat. Dieses Makro ist nur gültig bei Benachrichtigungen, bei denen das Makro \$NOTIFICATIONTYPE\$ auf "ACKNOWLEDGEMENT" gesetzt ist.
\$HOSTACKAUTHORNAME\$ ⁸	Eine Zeichenkette, die den Kurznamen der Kontaktperson (falls zutreffend) enthält, die das Host-Problem bestätigt hat. Dieses Makro ist nur gültig bei Benachrichtigungen, bei denen das Makro \$NOTIFICATIONTYPE\$ auf "ACKNOWLEDGEMENT" gesetzt ist.
\$HOSTACKAUTHORALIAS\$ ⁸	Eine Zeichenkette, die den Alias der Kontaktperson (falls zutreffend) enthält, die das Host-Problem bestätigt hat. Dieses Makro ist nur gültig bei Benachrichtigungen, bei denen das Makro \$NOTIFICATIONTYPE\$ auf "ACKNOWLEDGEMENT" gesetzt ist.
\$HOSTACKCOMMENT\$ ⁸	Eine Zeichenkette, die den Bestätigungscommentar enthält, den der Benutzer eingegeben hat, der das Host-Problem bestätigt hat. Dieses Makro ist nur gültig bei Benachrichtigungen, bei denen das Makro \$NOTIFICATIONTYPE\$ auf "ACKNOWLEDGEMENT" gesetzt ist.
\$HOSTACTIONURL\$	Der Action-URL für den Host. Dieses Makro kann andere Makros enthalten (z.B. \$HOSTNAME\$), was nützlich sein kann, wenn man den Hostnamen an eine Web-Seite übergeben will.
\$HOSTNOTESURL\$	Der Anmerkungs-URL für den Host. Dieses Makro kann andere Makros enthalten (z.B. \$HOSTNAME\$), was nützlich sein kann, wenn man den Hostnamen an eine Web-Seite übergeben will.

\$HOSTNOTES\$	Anmerkungen für den Host. Dieses Makro kann andere Makros enthalten (z.B. \$HOSTNAME\$), was nützlich sein kann, wenn man Hostspezifische Statusinformationen in der Beschreibung haben möchte.
\$TOTALHOSTSERVICES\$	Die Gesamtzahl von Services, die mit dem Host verbunden sind.
\$TOTALHOSTSERVICESOK\$	Die Gesamtzahl von Services im OK-Zustand, die mit dem Host verbunden sind.
\$TOTALHOSTSERVICESWARNING\$	Die Gesamtzahl von Services im WARNING-Zustand, die mit dem Host verbunden sind.
\$TOTALHOSTSERVICESUNKNOWN\$	Die Gesamtzahl von Services im UNKNOWN-Zustand, die mit dem Host verbunden sind.
\$TOTALHOSTSERVICESCRITICAL\$	Die Gesamtzahl von Services im CRITICAL-Zustand, die mit dem Host verbunden sind.
Hostgroup-Makros: 5	
\$HOSTGROUPALIAS\$ 5	Der Langname / Alias entweder des 1) Hostgruppennamens, der als On-Demand-Makro-Argument übergeben wurde oder 2) der primären Hostgruppe, die mit dem aktuellen Host verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde). Dieser Wert wird aus der <i>alias</i> -Direktive in der hostgroup-Definition genommen.
\$HOSTGROUPMEMBERS\$ 5	Eine Komma-separierte Liste aller Hosts, die entweder 1) zu dem Hostgruppennamen gehören, der als On-Demand-Makro-Argument übergeben wurde, oder 2) zu der primären Hostgruppe gehören, die mit dem aktuellen Host verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde).

\$HOSTGROUPNOTES\$ ⁵	Die Anmerkungen, die verbunden sind mit entweder 1) dem Hostgroup-Namen, der als On-Demand-Makro-Argument übergeben wurde oder 2) der primären Hostgruppe, die mit dem aktuellen Host verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde). Dieser Wert wird aus der <i>notes</i> -Direktive in der hostgroup-Definition genommen.
\$HOSTGROUPNOTESURL\$ ⁵	Der Anmerkungs-URL, der verbunden ist mit entweder 1) dem Hostgroup-Namen, der als On-Demand-Makro-Argument übergeben wurde oder 2) der primären Hostgruppe, die mit dem aktuellen Host verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde). Dieser Wert wird aus der <i>notes_url</i> -Direktive in der hostgroup-Definition genommen.
\$HOSTGROUPACTIONURL\$ ⁵	Der Action-URL, der verbunden ist mit entweder 1) dem Hostgroup-Namen, der als On-Demand-Makro-Argument übergeben wurde oder 2) der primären Hostgruppe, die mit dem aktuellen Host verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde). Dieser Wert wird aus der <i>action_url</i> -Direktive in der hostgroup-Definition genommen.
Service-Makros:	
\$SERVICEDESC\$	Der Langname/die Beschreibung des Service (z.B. "Main Website"). Dieser Wert wird aus der <i>description</i> -Direktive der Service-Definition genommen.
\$SERVICEDIPLAYNAME\$	Ein alternativer Anzeigenname für den Service. Dieser Wert wird aus der <i>display_name</i> -Direktive der Service-Definition genommen.
\$SERVICESTATE\$	Eine Zeichenkette, die den aktuellen Status des Service anzeigt ("OK", "WARNING", "UNKNOWN" oder "CRITICAL").
\$SERVICESTATEID\$	Eine Zahl, die dem aktuellen Status des Service entspricht: 0=OK, 1=WARNING, 2=CRITICAL, 3=UNKNOWN.

\$LASTSERVICESTATE\$	Eine Zeichenkette, die den letzten Status des Service angibt ("OK", "WARNING", "UNKNOWN" oder "CRITICAL").
\$LASTSERVICESTATEID\$	Eine Zahl, die dem letzten Status des Service entspricht: 0=OK, 1=WARNING, 2=CRITICAL, 3=UNKNOWN.
\$SERVICESTATETYPE\$	Eine Zeichenkette, die den Statustyp für die aktuelle Service-Prüfung anzeigt ("HARD" oder "SOFT"). Ein Soft-Status tritt auf, wenn eine Service-Prüfung einen nicht-OK Status zurückliefert und noch Wiederholungen anstehen. Ein Hard-Status liegt vor, wenn die Anzahl der Service-Prüfungswiederholungen einen maximal definierten Wert erreicht hat.
\$SERVICEATTEMPT\$	Die Anzahl der aktuellen Service-Prüfungswiederholungen. Wenn dies beispielsweise das zweite Mal ist, dass der Service erneut geprüft wird, dann steht hier die Zahl zwei. Die aktuelle Wiederholungsanzahl ist eigentlich nur dann sinnvoll, wenn man Eventhandler für Soft-Zustände schreibt, die auf einer bestimmten Aktion für diese entsprechende Zahl basieren.
\$MAXSERVICEATTEMPTS\$	Die max. Prüfversuche, wie sie für den aktuellen Service definiert sind. Nützlich, wenn man Service-Eventhandler für "Soft"-Zustände schreibt, die eine bestimmte Aktion ausführen basierend auf der Service-Wiederholungsanzahl.
\$SERVICEISVOLATILE\$	Zeigt an, ob der Service als sprunghaft ("volatile") markiert ist: 0 = not volatile, 1 = volatile.
\$SERVICEEVENTID\$	Eine global eindeutige Zahl verbunden mit dem aktuellen Status des Service. Jedes Mal, wenn eine Service- (oder Host-) Statusänderung eintritt, wird eine globale Ereignis-ID-Nummer um eins (1) erhöht. Falls bei einem Service keine Statusänderung eintritt, wird dieses Makro auf Null (0) gesetzt.
\$LASTSERVICEEVENTID\$	Die vorherige (global eindeutige) Ereigniszahl, die für den Service vergeben wurde.

\$SERVICEPROBLEMID\$	Eine global eindeutige Zahl verbunden mit dem aktuellen Problemstatus des Service. Jedes Mal, wenn ein Service (oder Host) von einem UP- oder OK-Status in einen Problemzustand wechselt, wird eine globale Problem-ID um eins (1) erhöht. Dieses Makro wird ungleich Null sein, wenn der Service sich gerade in einem Zustand ungleich OK befindet. Statuswechsel zwischen Zuständen ungleich OK (z.B. DOWN oder UNREACHABLE) erhöhen diese Problem-ID nicht. Wenn sich der Service gerade in einem OK-Zustand befindet, wird dieses Makro auf Null (0) gesetzt. In Kombination mit Eventhandlern kann dieses Makro benutzt werden, um automatisch ein Trouble-Ticket zu eröffnen, wenn Services das erste Mal einen Problemzustand erreichen.
\$LASTSERVICEPROBLEMID\$	Die vorherige (global eindeutige) Ereigniszahl, die für den Service vergeben wurde. In Kombination mit Eventhandlern kann dieses Makro benutzt werden, um automatisch ein Trouble-Ticket zu schließen, wenn Services zu einem Up-Status zurückkehren.
\$SERVICELATENCY\$	Eine (Fließkomma-) Zahl, die die Anzahl von Sekunden angibt, um die eine <i>geplante</i> Service-Prüfung nach der eigentlichen Planungszeit stattfand. Wenn beispielsweise eine Prüfung für 03:14:15 geplant war und erst um 03:14:17 ausgeführt wurde, dann beträgt die Verzögerung 2.0 Sekunden.
\$SERVICEEXECUTIONTIME\$	Eine (Fließkomma-) Zahl, die die Dauer der Ausführung einer Service-Prüfung in Sekunden angibt.
\$SERVICEDURATION\$	Eine Zeichenkette, die die Zeitdauer angibt, die sich der Service im aktuellen Status befindet. Das Format ist "XXh YYm ZZs" und gibt die Stunden, Minuten und Sekunden an.
\$SERVICEDURATIONSECS\$	Eine Zahl, die die Zeitdauer in Sekunden angibt, die sich der Service im aktuellen Status befindet.

\$SERVICEDOWNTIME\$	Eine Zahl, die die aktuelle "Downtime-Tiefe" für den Service angibt. Wenn dieser Service sich gerade in einer Phase einer geplanten Downtime befindet, ist dieser Wert größer als Null. Ist der Service nicht gerade in einer Downtime-Phase, ist dieser Wert Null.
\$SERVICEPERCENTCHANGE\$	Eine (Fließkomma-) Zahl, die den prozentualen Statuswechsel angibt, der der Service unterworfen war. Dieser Wert wird vom flap detection -Algorithmus benutzt.
\$SERVICEGROUPNAME\$	Der Kurzname der Servicegruppe, zu der dieser Service gehört. Dieser Wert wird aus der <code>servicegroup_name</code> -Direktive in der servicegroup-Definition entnommen. Wenn der Service zu mehreren Servicegruppen gehört, enthält dieses Makro nur einen der Namen.
\$SERVICEGROUPNAMES\$	Eine Komma-separierte Liste von Kurznamen aller Servicegruppen, zu denen dieser Service gehört.
\$LASTSERVICECHECK\$	Dieses ist ein Zeitstempel im <code>time_t</code> -Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt eine Prüfung des Service stattfand.
\$LASTSERVICESTATECHANGE\$	Dieses ist ein Zeitstempel im <code>time_t</code> -Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt ein Statuswechsel des Service stattfand.
\$LASTSERVICEOK\$	Dieses ist ein Zeitstempel im <code>time_t</code> -Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt festgestellt wurde, dass sich der Service in einem OK-Zustand befand.
\$LASTSERVICEWARNING\$	Dieses ist ein Zeitstempel im <code>time_t</code> -Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt festgestellt wurde, dass sich der Service in einem WARNING-Zustand befand.
\$LASTSERVICEUNKNOWN\$	Dieses ist ein Zeitstempel im <code>time_t</code> -Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt festgestellt wurde, dass sich der Service in einem UNKNOWN-Zustand befand.

\$LASTSERVICECRITICAL\$	Dieses ist ein Zeitstempel im time_t-Format (Sekunden seit der UNIX-Epoche), der die Zeit angibt, zu der zuletzt festgestellt wurde, dass sich der Service in einem CRITICAL-Zustand befand.
\$SERVICEOUTPUT\$	Die erste Zeile von Textausgaben der letzten Service-Prüfung (z.B. "Ping OK")
\$LONGSERVICEOUTPUT\$	Die vollständige Textausgabe (außer der ersten Zeile) der letzten Service-Prüfung.
\$SERVICEPERFDATA\$	Dieses Makro enthält jegliche Performance-Daten , die von der letzten Service-Prüfung geliefert worden sein könnten.
\$SERVICECHECKCOMMAND\$	Dieses Makro enthält den Namen des Befehls (zusammen mit übergebenen Argumenten), der zur Service-Prüfung benutzt wurde.
\$SERVICEACKAUTHOR\$ ⁸	Eine Zeichenkette, die den Namen des Benutzers enthält, der das Service-Problem bestätigt hat. Dieses Makro ist nur gültig bei Benachrichtigungen, bei denen das Makro \$NOTIFICATIONTYPE\$ auf "ACKNOWLEDGEMENT" gesetzt ist.
\$SERVICEACKAUTHORNAME\$ ⁸	Eine Zeichenkette, die den Kurznamen der Kontaktperson (falls zutreffend) enthält, die das Service-Problem bestätigt hat. Dieses Makro ist nur gültig bei Benachrichtigungen, bei denen das Makro \$NOTIFICATIONTYPE\$ auf "ACKNOWLEDGEMENT" gesetzt ist.
\$SERVICEACKAUTHORALIAS\$ ⁸	Eine Zeichenkette, die den Alias der Kontaktperson (falls zutreffend) enthält, die das Service-Problem bestätigt hat. Dieses Makro ist nur gültig bei Benachrichtigungen, bei denen das Makro \$NOTIFICATIONTYPE\$ auf "ACKNOWLEDGEMENT" gesetzt ist.
\$SERVICEACKCOMMENT\$ ⁸	Eine Zeichenkette, die den Bestätigungscommentar enthält, den der Benutzer eingegeben hat, der das Service-Problem bestätigt hat. Dieses Makro ist nur gültig bei Benachrichtigungen, bei denen das Makro \$NOTIFICATIONTYPE\$ auf "ACKNOWLEDGEMENT" gesetzt ist.

\$SERVICEACTIONURL\$	Der Action-URL für den Service. Dieses Makro kann andere Makros enthalten (z.B. \$HOSTNAME\$ oder \$SERVICEDESC\$), was nützlich sein kann, wenn man den Servicenamen an eine Web-Seite übergeben will.
\$SERVICENOTESURL\$	Der Anmerkungs-URL für den Service. Dieses Makro kann andere Makros enthalten (z.B. \$HOSTNAME\$ oder \$SERVICEDESC\$), was nützlich sein kann, wenn man den Servicenamen an eine Web-Seite übergeben will.
\$SERVICENOTES\$	Anmerkungen für den Service. Dieses Makro kann andere Makros enthalten (z.B. \$HOSTNAME\$ oder \$SERVICEDESC\$), was nützlich sein kann, wenn man Servicespezifische Statusinformationen in der Beschreibung haben möchte.
Servicegroup-Makros: ⁶	
\$SERVICEGROUPALIAS\$ ⁶	Der Langname / Alias entweder des 1) Servicegroup-Namens, der als On-Demand-Makro-Argument übergeben wurde oder 2) der primären Servicegruppe, die mit dem aktuellen Service verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde). Dieser Wert wird aus der <i>alias</i> -Direktive in der servicegroup-Definition genommen.
\$SERVICEGROUPMEMBERS\$ ⁶	Eine Komma-separierte Liste aller Services, die entweder 1) zu dem Servicegruppennamen gehören, der als On-Demand-Makro-Argument übergeben wurde, oder 2) zu der primären Servicegruppe gehören, die mit dem aktuellen Service verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde).
\$SERVICEGROUPNOTES\$ ⁶	Die Anmerkungen, die verbunden sind mit entweder 1) dem Servicegroup-Namen, der als On-Demand-Makro-Argument übergeben wurde oder 2) der primären Servicegruppe, die mit dem aktuellen Host verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde). Dieser Wert wird aus der <i>notes</i> -Direktive in der servicegroup-Definition genommen.

\$SERVICEGROUPNOTESURL\$ ⁶	Der Anmerkungs-URL, der verbunden sind mit entweder 1) dem Servicegroup-Namen, der als On-Demand-Makro-Argument übergeben wurde oder 2) der primären Servicegruppe, die mit dem aktuellen Service verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde). Dieser Wert wird aus der <i>notes_url</i> -Direktive in der servicegroup-Definition genommen.
\$SERVICEGROUPACTIONURL\$ ⁶	Der Action-URL, der verbunden sind mit entweder 1) dem Servicegroup-Namen, der als On-Demand-Makro-Argument übergeben wurde oder 2) der primären Servicegruppe, die mit dem aktuellen Service verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde). Dieser Wert wird aus der <i>action_url</i> -Direktive in der servicegroup-Definition genommen.
Contact-Makros:	
\$CONTACTNAME\$	Kurzname für den Kontakt (z.B. "mmustermann"), der über ein Host- oder Service-Problem informiert wird. Dieser Wert wird aus der <i>contact_name</i> -Direktive in der contact-Definition genommen.
\$CONTACTALIAS\$	Langname/Beschreibung für den Kontakt, der informiert wird. Dieser Wert wird aus der <i>alias</i> -Direktive in der contact-Definition genommen.
\$CONTACTEMAIL\$	e-Mail-Adresse für den Kontakt, der informiert wird. Dieser Wert wird aus der <i>email</i> -Direktive in der contact-Definition genommen.
\$CONTACTPAGER\$	Pager-Nummer/-Adresse für den Kontakt, der informiert wird. Dieser Wert wird aus der <i>pager</i> -Direktive in der contact-Definition genommen.

\$CONTACTADDRESSn\$	Adresse für den Kontakt, der informiert wird. Jeder Kontakt kann sechs verschiedene Adressen haben (zusätzlich zur e-Mail-Adresse und Pager-Nummer). Die Makros für diese Adressen sind \$CONTACTADDRESS1\$ - \$CONTACTADDRESS6\$. Dieser Wert wird aus der <i>addressx</i> -Direktive in der contact-Definition genommen.
\$CONTACTGROUPNAME\$	Der Kurzname für die Kontaktgruppe, deren Mitglied der Kontakt ist. Dieser Wert wird aus der <i>contact_group</i> -Direktive in der contactgroup-Definition genommen.
\$CONTACTGROUPNAMES\$	Eine Komma-separierte Liste der Kurznamen aller Kontaktgruppen, deren Mitglied dieser Kontakt ist.
<hr/>	
Contactgroup-Makros: 5	
\$CONTACTGROUPALIAS\$ 7	Der Langname / Alias entweder des 1) Contactgroup-Namens, der als On-Demand-Makro-Argument übergeben wurde oder 2) der primären Kontaktgruppe, die mit dem aktuellen Kontakt verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde). Dieser Wert wird aus der <i>alias</i> -Direktive in der contactgroup-Definition genommen.
\$CONTACTGROUPMEMBERS\$ 7	Eine Komma-separierte Liste aller Kontakte, die entweder 1) zu dem Kontaktgruppennamen gehören, der als On-Demand-Makro-Argument übergeben wurde, oder 2) zu der primären Kontaktgruppe gehören, die mit dem aktuellen Kontakt verbunden ist (falls sie nicht im Zusammenhang mit einem On-Demand-Makro benutzt wurde).
<hr/>	
Auswertungs-Makros:	
\$TOTALHOSTSUP\$	Dieses Makro gibt die Gesamtzahl der Hosts an, die sich in einem UP-Zustand befinden.
\$TOTALHOSTSDOWN\$	Dieses Makro gibt die Gesamtzahl der Hosts an, die sich in einem DOWN-Zustand befinden.

\$TOTALHOSTSUNREACHABLE\$	Dieses Makro gibt die Gesamtzahl der Hosts an, die sich in einem UNREACHABLE-Zustand befinden.
\$TOTALHOSTSDOWNUNHANDLED\$	Dieses Makro gibt die Gesamtzahl der Hosts an, die sich in einem DOWN-Status befinden und "unbehandelt" sind. Unbehandelte Host-Probleme sind solche, die nicht bestätigt sind, sich nicht in einer geplanten Downtime befinden, und für die Prüfungen momentan aktiviert sind.
\$TOTALHOSTSUNREACHABLEUNHANDLED\$	Dieses Makro gibt die Gesamtzahl der Hosts an, die sich in einem UNREACHABLE-Status befinden und "unbehandelt" sind. Unbehandelte Host-Probleme sind solche, die nicht bestätigt sind, sich nicht in einer geplanten Downtime befinden, und für die Prüfungen momentan aktiviert sind.
\$TOTALHOSTPROBLEMS\$	Dieses Makro gibt die Gesamtzahl der Hosts an, die sich in einem DOWN- oder UNREACHABLE-Status befinden.
\$TOTALHOSTPROBLEMSUNHANDLED\$	Dieses Makro gibt die Gesamtzahl der Hosts an, die sich in einem DOWN- oder UNREACHABLE-Status befinden und "unbehandelt" sind. Unbehandelte Host-Probleme sind solche, die nicht bestätigt sind, sich nicht in einer geplanten Downtime befinden, und für die Prüfungen momentan aktiviert sind.
\$TOTALSERVICESOK\$	Dieses Makro gibt die Gesamtzahl der Services an, die sich in einem OK-Status befinden.
\$TOTALSERVICESWARNING\$	Dieses Makro gibt die Gesamtzahl der Services an, die sich in einem WARNING-Status befinden.
\$TOTALSERVICESCRITICAL\$	Dieses Makro gibt die Gesamtzahl der Services an, die sich in einem CRITICAL-Status befinden.
\$TOTALSERVICESUNKNOWN\$	Dieses Makro gibt die Gesamtzahl der Services an, die sich in einem UNKNOWN-Status befinden.
\$TOTALSERVICESWARNINGUNHANDLED\$	Dieses Makro gibt die Gesamtzahl der Services an, die sich in einem WARNING-Status befinden und "unbehandelt" sind. Unbehandelte Service-Probleme sind solche, die nicht bestätigt sind, sich nicht in einer geplanten Downtime befinden, und für die Prüfungen momentan aktiviert sind.

\$TOTALSERVICESCRITICALUNHANDLED\$	Dieses Makro gibt die Gesamtzahl der Services an, die sich in einem CRITICAL-Status befinden und "unbehandelt" sind. Unbehandelte Service-Probleme sind solche, die nicht bestätigt sind, sich nicht in einer geplanten Downtime befinden, und für die Prüfungen momentan aktiviert sind.
\$TOTALSERVICESUNKNOWNUNHANDLED\$	Dieses Makro gibt die Gesamtzahl der Services an, die sich in einem UNKNOWN-Status befinden und "unbehandelt" sind. Unbehandelte Service-Probleme sind solche, die nicht bestätigt sind, sich nicht in einer geplanten Downtime befinden, und für die Prüfungen momentan aktiviert sind.
\$TOTALSERVICEPROBLEMS\$	Dieses Makro gibt die Gesamtzahl der Services an, die sich in einem WARNING-, CRITICAL- oder UNKNOWN-Status befinden.
\$TOTALSERVICEPROBLEMSUNHANDLED\$	Dieses Makro gibt die Gesamtzahl der Services an, die sich in einem WARNING-, CRITICAL- oder UNKNOWN-Status befinden und "unbehandelt" sind. Unbehandelte Service-Probleme sind solche, die nicht bestätigt sind, sich nicht in einer geplanten Downtime befinden, und für die Prüfungen momentan aktiviert sind.
Benachrichtigungs-Makros:	
\$NOTIFICATIONTYPE\$	Eine Zeichenkette, die den Typ der Benachrichtigung angibt, die versandt wird ("PROBLEM", "RECOVERY", "ACKNOWLEDGEMENT", "FLAPPINGSTART", "FLAPPINGSTOP", "FLAPPINGDISABLED", "DOWNTIMESTART", "DOWNTIMEEND" oder "DOWNTIMECANCELLED").
\$NOTIFICATIONRECIPIENTS\$	Eine Komma-separierte Liste der Kurznamen von allen Kontakten, die über den Host oder Service benachrichtigt werden.

\$NOTIFICATIONISESCALATED\$	Eine Ganzzahl, die angibt, ob diese Benachrichtigung an normale Kontakte für den Host oder Service versandt wurde, oder ob sie eskaliert wurde. 0 = normale (nicht-eskalierte) Benachrichtigung, 1 = eskalierte Benachrichtigung
\$NOTIFICATIONAUTHOR\$	Eine Zeichenkette, die den Namen des Benutzers angibt, der die Benachrichtigung geschrieben hat. Falls das \$NOTIFICATIONTYPE\$-Makro auf "DOWNTIMESTART" oder "DOWNTIMEEND" gesetzt ist, wird es der Name des Benutzers sein, der die Downtime für den Host oder Service geplant hat. Falls das \$NOTIFICATIONTYPE\$-Makro auf "ACKNOWLEDGEMENT" gesetzt ist, wird es der Name des Benutzers sein, der das Problem für den Host oder Service bestätigt hat. Falls das \$NOTIFICATIONTYPE\$-Makro auf "CUSTOM" gesetzt ist, wird es der Name des Benutzers sein, der die benutzerdefinierte Benachrichtigung für den Host oder Service ausgelöst hat.
\$NOTIFICATIONAUTHORNAME\$	Eine Zeichenkette, die den Kurznamen des Kontakts (falls zutreffend) enthält, der im Makro \$NOTIFICATIONAUTHOR\$ angegeben wurde.
\$NOTIFICATIONAUTHORALIAS\$	Eine Zeichenkette, die den Alias des Kontakts (falls zutreffend) enthält, der im Makro \$NOTIFICATIONAUTHOR\$ angegeben wurde.
\$NOTIFICATIONCOMMENT\$	Eine Zeichenkette, die den Kommentar des Benutzers angibt, der die Benachrichtigung geschrieben hat. Falls das \$NOTIFICATIONTYPE\$-Makro auf "DOWNTIMESTART" oder "DOWNTIMEEND" gesetzt ist, wird es der Kommentar des Benutzers sein, der die Downtime für den Host oder Service geplant hat. Falls das \$NOTIFICATIONTYPE\$-Makro auf "ACKNOWLEDGEMENT" gesetzt ist, wird es der Kommentar des Benutzers sein, der das Problem für den Host oder Service bestätigt hat. Falls das \$NOTIFICATIONTYPE\$-Makro auf "CUSTOM" gesetzt ist, wird es der Kommentar des Benutzers sein, der die benutzerdefinierte Benachrichtigung für den Host oder Service ausgelöst hat.

\$HOSTNOTIFICATIONNUMBER\$	Die aktuelle Benachrichtigungsnummer für den Host. Die Benachrichtigungsnummer wird jedes Mal um eins (1) erhöht, wenn eine neue Benachrichtigung für den Host versandt wird (außer bei Bestätigungen). Die Benachrichtigungsnummer wird auf Null (0) zurückgesetzt, wenn der Host wieder im UP-Zustand ist (<i>nachdem</i> die Benachrichtigung versandt wurde). Die Benachrichtigungsnummer wird weder durch Bestätigungen noch durch Benachrichtigungen über "Flap detection" oder geplante Downtimes erhöht.
\$HOSTNOTIFICATIONID\$	Eine eindeutige Zahl, die eine Host-Benachrichtigung identifiziert. Benachrichtigungsnummern sind eindeutig sowohl für Host- als auch Service-Benachrichtigungen, so dass Sie diese eindeutige Zahl als primären Schlüssel in einer Benachrichtigungs-Datenbank benutzen können. Benachrichtigungsnummern sollten über den Restart des Icinga-Prozesses hinweg eindeutig bleiben, solange Sie "state retention" aktiviert haben. Die Benachrichtigungsnummer wird für jede neue Host-Benachrichtigung um eins (1) erhöht, unabhängig von der Anzahl der benachrichtigten Kontakte.
\$SERVICENOTIFICATIONNUMBER\$	Die aktuelle Benachrichtigungsnummer für den Service. Die Benachrichtigungsnummer wird jedes Mal um eins (1) erhöht, wenn eine neue Benachrichtigung für den Service versandt wird (außer bei Bestätigungen). Die Benachrichtigungsnummer wird auf Null (0) zurückgesetzt, wenn der Service wieder im OK-Zustand ist (<i>nachdem</i> die Benachrichtigung versandt wurde). Die Benachrichtigungsnummer wird weder durch Bestätigungen noch durch Benachrichtigungen über "Flap detection" oder geplante Downtimes erhöht.

\$SERVICENOTIFICATIONID\$	Eine eindeutige Zahl, die eine Service-Benachrichtigung identifiziert. Benachrichtigungsnummern sind eindeutig sowohl für Host- als auch Service-Benachrichtigungen, so dass Sie diese eindeutige Zahl als primären Schlüssel in einer Benachrichtigungs-Datenbank benutzen können. Benachrichtigungsnummern sollten über den Restart des Icinga-Prozesses hinweg eindeutig bleiben, solange Sie "state retention" aktiviert haben. Die Benachrichtigungsnummer wird für jede neue Service-Benachrichtigung um eins (1) erhöht, unabhängig von der Anzahl der benachrichtigten Kontakte.
Datum-/Zeit-Makros:	
\$LONGDATETIME\$	Aktueller Datum-/Zeitstempel (z.B. <i>Fri Oct 13 00:30:28 CDT 2000</i>). Das Datum-Format ist festgelegt durch die date_format -Direktive.
\$SHORTDATETIME\$	Aktueller Datum-/Zeitstempel (z.B. <i>10-13-2000 00:30:28</i>). Das Datum-Format ist festgelegt durch die date_format -Direktive.
\$DATE\$	Aktueller Datumstempel (z.B. <i>10-13-2000</i>). Das Datum-Format ist festgelegt durch die date_format -Direktive.
\$TIME\$	Aktueller Zeitstempel (z.B. <i>00:30:28</i>).
\$TIMET\$	Aktueller Zeitstempel im time_t-Format (Sekunden seit der UNIX-Epoche).

\$ISVALIDTIME:\$ ⁹	<p>Dies ist ein spezielles On-Demand-Makro, das 1 oder 0 zurückliefert, abhängig davon, ob eine bestimmte Zeit innerhalb einer angegebenen Zeitperiode gültig ist. Es gibt zwei Arten, dieses Makro zu benutzen:</p> <ol style="list-style-type: none"> 1. \$ISVALIDTIME:24x7\$ wird auf "1" gesetzt, wenn die aktuelle Zeit innerhalb der "24x7"-Zeitperiode gültig ist. Falls nicht, wird es auf "0" gesetzt. 2. \$ISVALIDTIME:24x7:timestamp\$ wird auf "1" gesetzt, wenn die durch das "timestamp"-Argument angegebene Zeit (die im time_t-Format sein muss) innerhalb der "24x7"-Zeitperiode gültig ist. Falls nicht, wird es auf "0" gesetzt.
\$NEXTVALIDTIME:\$ ⁹	<p>Dies ist ein spezielles On-Demand-Makro, das die nächste gültige Zeit (im time_t-Format) für eine angegebene Zeitperiode zurückliefert. Es gibt zwei Arten, dieses Makro zu benutzen:</p> <ol style="list-style-type: none"> 1. \$NEXTVALIDTIME:24x7\$ wird die nächste gültige Zeit zurückliefern - ab der aktuellen Zeit - innerhalb der "24x7"-Zeitperiode. 2. \$NEXTVALIDTIME:24x7:timestamp\$ wird die nächste gültige Zeit zurückliefern - ab der durch das "timestamp"-Argument angegebenen Zeit (die im time_t-Format sein muss) - innerhalb der "24x7"-Zeitperiode. <p>Falls keine gültige Zeit innerhalb der angegebenen Zeitperiode gefunden werden kann, wird das Makro auf "0" gesetzt.</p>
Datei-Makros:	
\$MAINCONFIGFILE\$	Der Standort der Hauptkonfigurationsdatei (main config file).
\$STATUSDATAFILE\$	Der Standort der Statusdaten-Datei (main config file).
\$COMMENTDATAFILE\$	Der Standort der Kommentardaten-Datei (comment data file).

\$DOWNTIMEDATAFILE\$	Der Standort der Ausfallzeitendaten-Datei (downtime data file).
\$RETENTIONDATAFILE\$	Der Standort der Aufbewahrungsdaten-Datei (retention data file).
\$OBJECTCACHEFILE\$	Der Standort der Objektzwischenspeicherungs-Datei (object cache file).
\$TEMPFILE\$	Der Standort der temporären Datei (temp file).
\$TEMPPATH\$	Das durch die temp path -Variable festgelegte Verzeichnis.
\$LOGFILE\$	Der Standort der Protokolldatei (log file).
\$RESOURCEFILE\$	Der Standort der Ressource-Datei (resource file).
\$COMMANDFILE\$	Der Standort der Befehlsdatei (command file).
\$HOSTPERFDATAFILE\$	Der Standort der Host-Performance Daten-Datei (host performance data file; falls definiert).
\$SERVICEPERFDATAFILES\$	Der Standort der Service-Performance Daten-Datei (service performance data file; falls definiert).
Verschiedene Makros:	
\$PROCESSSTARTTIME\$	Zeitstempel im time_t-Format (Sekunden seit der UNIX-Epoche), die angibt, wann der Icinga-Prozess das letzte Mal neu/wieder gestartet wurde. Sie können die Laufzeit durch Subtraktion von \$PROCESSSTARTTIME\$ von \$TIMET\$ ermitteln.
\$EVENTSTARTTIME\$	Zeitstempel im time_t-Format (Sekunden seit der UNIX-Epoche), die angibt, wann der Icinga-Prozess begann, Ereignisse (Prüfungen, usw.) zu verarbeiten. Sie können die Zeit, die Icinga zum Start benötigte, durch Subtraktion von \$PROCESSSTARTTIME\$ von \$EVENTSTARTTIME\$ ermitteln.
\$ADMINEMAIL\$	Globale administrative e-Mail-Adresse. Dieser Wert wird aus der admin_email -Direktive genommen.

\$ADMINPAGER\$	Globale administrative Pager-Nummer/-Adresse. Dieser Wert wird aus der admin_pager -Direktive genommen.
\$ARGn\$	Das <i>n</i> -te an den Befehl (Benachrichtigung, Eventhandler, Service-Prüfungen, usw.) übergebene Argument. Icinga unterstützt bis zu 32 Argument-Makros (\$ARG1\$ bis \$ARG32\$).
\$USERn\$	Das <i>n</i> -te benutzerdefinierbare Makro. Benutzermakros können in ein oder mehreren resource files definiert werden. Icinga unterstützt bis zu 256 User-Makros (\$USER1\$ bis \$USER256\$).

Anmerkungen

¹ Diese Makros sind nicht gültig für den Host, dem sie zugeordnet sind, wenn der Host gerade geprüft wird (denn die Werte konnten noch nicht ermittelt werden).

² Diese Makros sind nicht gültig für den Service, dem sie zugeordnet sind, wenn der Service gerade geprüft wird (denn die Werte konnten noch nicht ermittelt werden).

³ Wenn Host-Makros in Service-bezogenen Befehlen benutzt werden (z.B. Service-Benachrichtigungen, Eventhandler, usw.) verweisen sie auf den Host, dem der Service zugeordnet ist.

⁴ Wenn Host- und Service-Auswertungsmakros in Benachrichtigungen benutzt werden, werden die Summen gefiltert, um so nur die Hosts und Services zu berücksichtigen, für die die Kontaktperson berechtigt ist (z.B. Hosts und Services, für die sie Benachrichtigungen erhalten soll).

⁵ Diese Makros sind normalerweise der ersten/primären Hostgruppe des aktuellen Hosts zugeordnet. Sie können deshalb in vielen Fällen als Host-Makros angesehen werden. Allerdings sind diese Makros nicht als On-Demand-Makros verfügbar. Statt dessen können sie als On-Demand-Hostgroup-Makros benutzt werden, wenn Sie den Namen einer Hostgruppe an das Makro übergeben. Beispielsweise würde \$HOSTGROUPMEMBERS:hg1\$ eine komma-separierte Liste aller (Host)-Mitglieder der Hostgruppe hg1 zurückliefern.

⁶ Diese Makros sind normalerweise der ersten/primären Servicegruppe des aktuellen Service zugeordnet. Sie können deshalb in vielen Fällen als Service-Makros angesehen werden. Allerdings sind diese Makros nicht als On-Demand-Makros verfügbar. Statt dessen können sie als On-Demand-Servicegroup-Makros benutzt werden, wenn Sie den Namen einer Servicegruppe an das Makro übergeben. Beispielsweise würde \$SERVICEGROUPMEMBERS:sg1\$ eine komma-separierte Liste aller (Service)-Mitglieder der Servicegruppe sg1 zurückliefern.

⁷ Diese Makros sind normalerweise der ersten/primären Kontaktgruppe des aktuellen Kontakts zugeordnet. Sie können deshalb in vielen Fällen als Kontakt-Makros angesehen werden. Allerdings sind diese Makros nicht als On-Demand-Makros verfügbar. Statt dessen können sie als On-Demand-Contactgroup-Makros benutzt werden, wenn Sie den Namen einer Kontaktgruppe an das Makro übergeben. Beispielsweise würde \$CONTACTGROUPMEMBERS:cg1\$ eine komma-separierte Liste aller (Kontakt)-Mitglieder der Kontaktgruppe cg1 zurückliefern.

⁸ Diese Bestätigungsmakros sind veraltet. Nutzen Sie statt dessen die mehr generischen Makros \$NOTIFICATIONAUTHOR\$, \$NOTIFICATIONAUTHORNAME\$, \$NOTIFICATIONAUTHORALIAS\$ oder \$NOTIFICATIONCOMMENT\$

⁹ Diese Makros sind nur als On-Demand-Makros verfügbar - d.h. Sie müssen ein zusätzliches Argument übergeben, um sie zu nutzen. Diese Makros sind nicht als Umgebungsvariablen verfügbar.

¹⁰ Auswertungsmakros sind nicht als Umgebungsvariablen verfügbar, wenn die Option `use_large_installation_tweaks` aktiviert ist, weil sie ziemlich CPU-intensiv zu berechnen sind.

[Zurück](#)[Nach oben](#)[Weiter](#)

Makros verstehen und wie sie
arbeiten

[Zum Anfang](#)

Host-Prüfungen (Host checks)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Host-Prüfungen (Host checks)

[Zurück](#)[Kapitel 5. Die Grundlagen](#)[Weiter](#)

Host-Prüfungen (Host checks)

Einführung

Die grundlegenden Tätigkeiten von Host-Prüfungen werden hier beschrieben...

Wann werden Host-Prüfungen durchgeführt?

Hosts werden durch den Icinga-Daemon geprüft

- in regelmäßigen Intervallen, wie sie durch die `check_interval` und `retry_interval`-Optionen in Ihren [Host-Definitionen](#) festgelegt sind.
- nach Bedarf, wenn ein mit dem Host verbundener Service den Status wechselt.
- nach Bedarf als Teil der [Host-Fähigkeit](#)-Logik.
- nach Bedarf bei [vorausschauenden Host-Abhängigkeitsprüfungen](#).

Regelmäßige Host-Prüfungen sind optional. Wenn Sie die `check_interval`-Option in Ihrer Host-Definition auf Null (0) setzen, wird Icinga keine Host-Prüfungen auf planmäßiger Basis durchführen. Es wird jedoch weiterhin nach Bedarf Prüfungen für den Host durchführen für andere Teile der Überwachungslogik.

Prüfungen nach Bedarf werden gemacht, wenn ein mit dem Host verbundener Service den Status wechselt, denn Icinga muss wissen, ob auch der Host den Status gewechselt hat. Services, die den Status wechseln, sind oft ein Indikator dafür, dass auch der Host den Status gewechselt hat. Wenn beispielsweise der mit einem Host verbundene HTTP-Service den Status von CRITICAL auf OK gewechselt hat, kann das bedeuten, dass der Host gerade einen Reboot beendet hat und nun wieder verfügbar ist.

Host-Prüfungen nach Bedarf werden auch als Teil der [Host-Erreichbarkeit](#) erledigt. Icinga ist so konstruiert, dass Netzwerkausfälle so schnell wie möglich erkannt werden und zwischen DOWN- und UNREACHABLE-Zuständen unterschieden werden kann. Das sind sehr unterschiedliche Zustände und es kann dem Admin helfen, schnell die Ursache für einen Netzwerkausfall zu finden.

Prüfungen nach Bedarf werden auch als Teil der [vorausschauenden Host-Abhängigkeitsprüfungs](#)-Logik durchgeführt.

zwischengespeicherte Host-Prüfungen (cached host checks)

Die Performance von Host-Prüfungen nach Bedarf kann signifikant durch den Einsatz von "cached checks" erhöht werden, die es Icinga erlauben, auf eine Host-Prüfung zu verzichten, wenn es feststellt, dass ein relativ frisches Prüfungsergebnis genügt. Mehr Informationen zu "cached checks" finden Sie [hier](#).

Abhängigkeiten und Prüfungen

Sie können [Host-Ausführungs-Abhängigkeiten](#) definieren, die Icinga von der Statusprüfung eines Hosts abhalten in Abhängigkeit vom Status ein oder mehrerer anderer Hosts. Mehr Informationen zu Abhängigkeiten finden Sie [hier](#).

Parallelisierung von Host-Prüfungen

Geplante Host-Prüfungen laufen parallel. Wenn Icinga eine geplante Host-Prüfung ausführt, wird es die Host-Prüfung veranlassen und dann zu anderen Arbeiten zurückkehren (Service-Prüfungen ausführen, etc.). Die Host-Prüfung läuft in einem Kind-Prozess, der vom Haupt-Icinga-Prozess aufgerufen wird ("fork(ed)"). Wenn die Host-Prüfung beendet ist, wird der Kind-Prozess den Haupt-Icinga-Prozess (seinen Eltern-Prozess) über das Ergebnis informieren. Der Haupt-Icinga-Prozess wird dann das Prüfungsergebnis behandeln und geeignete Aktionen durchführen (Eventhandler starten, Benachrichtigungen senden, usw.).

Host-Prüfungen nach Bedarf laufen ebenfalls parallel, falls notwendig. Wie bereits vorher erwähnt kann Icinga auf die eigentliche Ausführung einer Host-Prüfung nach Bedarf verzichten, wenn es das gespeicherte Ergebnis einer relativ frischen Host-Prüfung benutzen kann.

Wenn Icinga die Ergebnisse von geplanten und nach Bedarf ausgeführten Host-Prüfungen verarbeitet, kann es (zusätzliche) Prüfungen anderer Hosts veranlassen. Diese Prüfungen können aus zwei Gründen veranlasst werden: [vorausschauende Abhängigkeitsprüfungen](#) und um den Status des Hosts mit Hilfe von [Netzwerk-Erreichbarkeits-Logik](#) festzustellen. Die zusätzlichen Prüfungen werden normalerweise parallel ausgeführt. Allerdings gibt es eine große Ausnahme, der Sie sich bewusst sein sollten, da sie einen negativen Einfluss auf die Performance haben kann...

 Hosts, deren `max_check_attempts`-Wert auf **1** gesetzt sind, können schwerwiegende Performance-Probleme verursachen. Der Grund? Wenn Icinga den richtigen Status mit Hilfe der [Netzwerk-Erreichbarkeits-Logik](#) ermitteln muss (um zu sehen, ob sie DOWN oder UNREACHABLE sind), muss es **aufeinanderfolgende** Prüfungen für alle direkten Eltern des Hosts starten. Um es noch einmal zu wiederholen, diese Prüfungen laufen *nacheinander* statt parallel, also kann es zu einem Performance-Einbruch kommen. Aus diesem Grund würden wir empfehlen, dass Sie immer einen Wert größer als 1 für die `max_check_attempts`-Direktiven in Ihren Host-Definitionen benutzen.

Host-Zustände

Hosts, die geprüft werden, können in einem von drei unterschiedlichen Zuständen sein

- UP
- DOWN
- UNREACHABLE

Host-Statusermittlung

Host-Prüfungen werden mit Hilfe von [Plugins](#) durchgeführt, die den Status OK, WARNING, UNKNOWN oder CRITICAL zurückliefern können. Wie übersetzt Icinga diese Return-Codes der Plugins in die Host-Zustände UP, DOWN oder UNREACHABLE? Wir werden sehen...

Die nachfolgende Tabelle zeigt, wie sich die Return-Codes von Plugins mit vorläufigen Host-Zuständen decken. Einige Nachbearbeitung (die später beschrieben wird) ergibt den endgültigen Host-Zustand.

Plugin-Ergebnis	vorläufiger Host-Zustand
OK	UP
WARNING	UP oder DOWN*
UNKNOWN	DOWN
CRITICAL	DOWN

 Anmerkung: Das Ergebnis WARNING bedeutet normalerweise, dass der Host UP ist. Trotzdem werden WARNING-Ergebnisse so interpretiert, dass der Host DOWN ist, wenn die [use_aggressive_host_checking](#)-Option aktiviert ist.

Wenn der vorläufige Host-Status DOWN ist, wird Icinga versuchen festzustellen, ob der Host wirklich DOWN ist oder UNREACHABLE. Die Unterscheidung zwischen den Host-Zuständen DOWN und UNREACHABLE ist wichtig, weil es Admins erlaubt, die Grundursache von Netzwerkausfällen schneller zu ermitteln. Die folgende Tabelle zeigt, wie Icinga eine endgültige Zustandsermittlung basierend auf dem Zustand der Eltern des Hosts durchführt. Die Eltern eines Hosts werden in der *parents*-Direktive der Host-Definition festgelegt.

vorläufiger Host-Zustand	Zustand Host-Eltern	endgültiger Host-Zustand
DOWN	mindestens ein Elternteil ist UP	DOWN
DOWN	alle Eltern sind entweder DOWN oder UNREACHABLE	UNREACHABLE

Mehr Informationen, wie Icinga zwischen DOWN- und UNREACHABLE-Zuständen unterscheidet, finden Sie [hier](#).

Host-Statusänderungen

Wie Ihnen wahrscheinlich bereits bewusst ist, bleiben Hosts nicht immer in einem Zustand. Dinge gehen kaputt, Patches werden eingespielt und Server müssen neu gestartet werden. Wenn Icinga den Status von Hosts prüft, ist es in der Lage festzustellen, wenn ein Host zwischen UP-, DOWN- und UNREACHABLE-Zuständen wechselt und geeignete Maßnahmen ergreifen. Diese Zustandsänderungen resultieren in verschiedenen [Statustypen](#) (HARD oder SOFT), was zum Auslösen von [Eventhandlern](#) und dem Versenden von [Benachrichtigungen](#) führen kann. Das Erkennen und Behandeln von Statusänderungen ist das, worum es sich bei Icinga handelt.

Wenn Host-Statusänderungen zu oft erfolgen, werden sie als "flatternd" (flapping) angesehen. Ein gutes Beispiel für einen flatternden Host wäre ein Server, der spontan jedes Mal neu startet, sobald das Betriebssystem lädt. Das ist immer ein spaßiges Szenario, mit dem man sich befassen

muss. Icinga kann erkennen, wenn Hosts anfangen zu flattern, und kann Benachrichtigungen unterdrücken, bis das Flattern stoppt und sich der Host-Status stabilisiert. Mehr Informationen über die Erkennungslogik des Flatterns finden Sie [hier](#).

[Zurück](#)[Nach oben](#)[Weiter](#)[Standard-Makros in Icinga](#)[Zum Anfang](#)[Service-Prüfungen \(Service Checks\)](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Service-Prüfungen (Service Checks)

[Zurück](#)[Kapitel 5. Die Grundlagen](#)[Weiter](#)

Service-Prüfungen (Service Checks)

Einführung

Die grundlegenden Tätigkeiten von Service-Prüfungen werden hier beschrieben...

Wann werden Service-Prüfungen durchgeführt?

Services werden durch den Icinga-Daemon geprüft

- in regelmäßigen Intervallen, wie sie durch die `check_interval` und `retry_interval`-Optionen in Ihren [Service-Definitionen](#) festgelegt sind.
- nach Bedarf bei [vorausschauende Host-Abhängigkeitsprüfungen](#).

Prüfungen nach Bedarf werden als Teil der [vorausschauenden Service-Abhängigkeitsprüfungs](#)-Logik durchgeführt. Diese Prüfungen helfen sicherzustellen, dass die Abhängigkeitslogik so genau wie möglich ist. Falls Sie die [Service-Abhängigkeiten](#) nicht nutzen, wird Icinga keine Service-Prüfungen nach Bedarf durchführen.

zwischengespeicherte Service-Prüfungen (cached service checks)

Die Performance von Service-Prüfungen nach Bedarf kann signifikant durch den Einsatz von "cached checks" erhöht werden, die es Icinga erlauben, auf eine Service-Prüfung zu verzichten, wenn es feststellt, dass ein relativ frisches Prüfungsergebnis genügt. Cached checks werden nur dann einen Performance-Gewinn ergeben, wenn Sie [Service-Abhängigkeiten](#) nutzen. Mehr Informationen zu "cached checks" finden Sie [hier](#).

Abhängigkeiten und Prüfungen

Sie können [Service-Ausführungs-Abhängigkeiten](#) definieren, die Icinga von der Statusprüfung eines Service abhalten in Abhängigkeit vom Status ein oder mehrerer anderer Services. Mehr Informationen zu Abhängigkeiten finden Sie [hier](#).

Parallelisierung von Service-Prüfungen

Geplante Service-Prüfungen laufen parallel. Wenn Icinga eine geplante Service-Prüfung ausführt, wird es die Service-Prüfung veranlassen und dann zu anderen Arbeiten zurückkehren (Host-Prüfungen ausführen, etc.). Die Service-Prüfung läuft in einem Kind-Prozess, der vom Haupt-Icinga-Prozess aufgerufen wird ("fork()ed"). Wenn die Service-Prüfung beendet ist, wird der Kind-Prozess den Haupt-Icinga-Prozess (seinen Eltern-Prozess) über das Ergebnis informieren. Der Haupt-Icinga-Prozess wird dann das Prüfungsergebnis behandeln und geeignete Aktionen durchführen (Eventhandler starten, Benachrichtigungen senden, usw.).

Service-Prüfungen nach Bedarf laufen ebenfalls parallel, falls notwendig. Wie bereits vorher erwähnt kann Icinga auf die eigentliche Ausführung einer Service-Prüfung nach Bedarf verzichten, wenn es das gespeicherte Ergebnis einer relativ frischen Service-Prüfung benutzen kann.

Service-Zustände

Services, die geprüft werden, können in einem von vier unterschiedlichen Zuständen sein

- OK
- WARNING
- UNKNOWN
- CRITICAL

Service-Statusermittlung

Service-Prüfungen werden mit Hilfe von [Plugins](#) durchgeführt, die den Status OK, WARNING, UNKNOWN oder CRITICAL zurückliefern können. Diese Return-Codes der Plugins werden direkt in die Service-Zustände übersetzt. Beispielsweise wird das WARNING-Ergebnis eines Plugins zu einem WARNING-Status eines Service führen.

Service-Statusänderungen

Wenn Icinga den Status von Services prüft, ist es in der Lage festzustellen, wenn ein Service zwischen OK-, WARNING-, UNKNOWN- und CRITICAL-Zuständen wechselt und geeignete Maßnahmen ergreifen. Diese Zustandsänderungen resultieren in verschiedenen [Statustypen](#) (HARD oder SOFT), was zum Auslösen von [Eventhandlern](#) und dem Versenden von [Benachrichtigungen](#) führen kann. Service-Statusänderungen können auch zum Auslösen von [Host-Prüfungen](#) nach Bedarf führen. Das Erkennen und Behandeln von Statusänderungen ist das, worum es sich bei Icinga handelt.

Wenn Service-Statusänderungen zu oft erfolgen, werden sie als "flatternd" (flapping) angesehen. Icinga kann erkennen, wenn Services anfangen zu flattern, und kann Benachrichtigungen unterdrücken, bis das Flattern stoppt und sich der Service-Status stabilisiert. Mehr Informationen über die Erkennungslogik des Flatterns finden Sie [hier](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Host-Prüfungen \(Host checks\)](#)

[Zum Anfang](#)

[Aktive Prüfungen \(Active Checks\)](#)



Aktive Prüfungen (Active Checks)

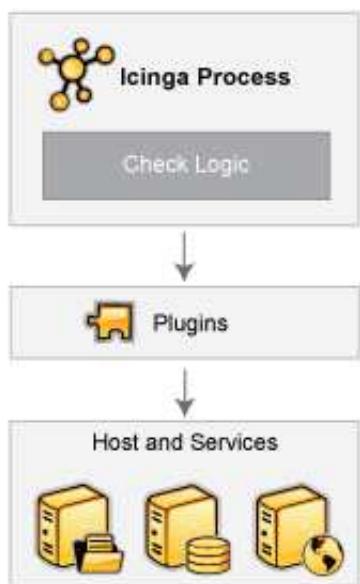
[Zurück](#)[Kapitel 5. Die Grundlagen](#)[Weiter](#)

Aktive Prüfungen (Active Checks)

Einführung

Icinga ist in der Lage, Hosts und Services auf zwei Arten zu überwachen: aktiv und passiv. Passive Prüfungen werden [anderswo](#) beschrieben, so dass wir uns hier auf aktive Prüfungen konzentrieren. Aktive Prüfungen sind die gebräuchlichste Methode zur Überwachung von Hosts und Services. Die Hauptmerkmale von aktiven Prüfungen sind

- aktive Prüfungen werden vom Icinga-Prozess veranlasst
- aktive Prüfungen laufen auf einer regelmäßig geplanten Basis



Wie werden aktive Prüfungen durchgeführt?

Aktive Prüfungen werden durch die Prüfungslogik im Icinga-Daemon veranlasst. Wenn Icinga den Status eines Hosts oder Services prüfen muss, wird es ein Plugin ausführen und die Informationen übergeben, was geprüft werden soll. Das Plugin wird dann den Betriebszustand des Hosts oder Service prüfen und die Ergebnisse an den Icinga-Daemon zurückmelden. Icinga wird die Ergebnisse der Host- oder Service-Prüfung verarbeiten und entsprechend notwendige Aktionen ausführen (z.B. Benachrichtigungen versenden, Eventhandler ausführen, usw.).

Mehr Informationen, wie Plugins arbeiten, finden Sie [hier](#).

Wann werden aktive Prüfungen ausgeführt?

Aktive Prüfungen werden ausgeführt

- in regelmäßigen Intervallen, wie sie in den *check_interval* und *retry_interval*-Optionen in Ihren Host- und Service-Definitionen festgelegt sind
- nach Bedarf

Regelmäßig geplante Prüfungen erfolgen in Intervallen, die den Einstellungen in *check_interval* oder *retry_interval* in Ihren Host- oder Service-Definitionen entsprechen, abhängig davon, in welchem [Statustyp](#) sich der Host oder Service befindet.

Prüfungen nach Bedarf werden ausgeführt, wann immer Icinga die Notwendigkeit sieht, die neuesten Statusinformationen über einen bestimmten Host oder Service zu ermitteln. Wenn Icinga beispielsweise die [Erreichbarkeit](#) eines Hosts feststellt, wird es oft Prüfungen von Eltern- und Kind-Hosts durchführen, um den genauen Status eines bestimmten Netzwerk-Segments zu ermitteln. Prüfungen nach Bedarf finden sich auch in der [vorausschauenden Abhängigkeitsprüfungs-Logik](#), um sicherzustellen, dass Icinga möglichst genaue Statusinformationen hat.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Service-Prüfungen (Service Checks)

[Zum Anfang](#)

Passive Prüfungen (Passive Checks)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Passive Prüfungen (Passive Checks)

[Zurück](#)[Kapitel 5. Die Grundlagen](#)[Weiter](#)

Passive Prüfungen (Passive Checks)

Einführung

In den meisten Fällen werden Sie Icinga nutzen, um Ihre Hosts und Services mit Hilfe von regelmäßig geplanten [aktiven Prüfungen](#) zu überwachen. Aktive Prüfungen können genutzt werden, um ein Gerät oder Service gelegentlich "abzufragen". Icinga unterstützt auch einen Weg, Hosts und Services passiv zu überwachen statt aktiv. Die Hauptmerkmale von passiven Prüfungen sind wie folgt:

- passive Prüfungen werden von externen Anwendungen/Prozessen veranlasst und ausgeführt
- Ergebnisse von passiven Prüfungen werden an Icinga zur Verarbeitung übermittelt

Der Hauptunterschied zwischen aktiven und passiven Prüfungen ist, dass aktive Prüfungen von Icinga veranlasst und ausgeführt werden, während passive Prüfungen von externen Applikationen durchgeführt werden.

Einsatzmöglichkeiten für passive Prüfungen

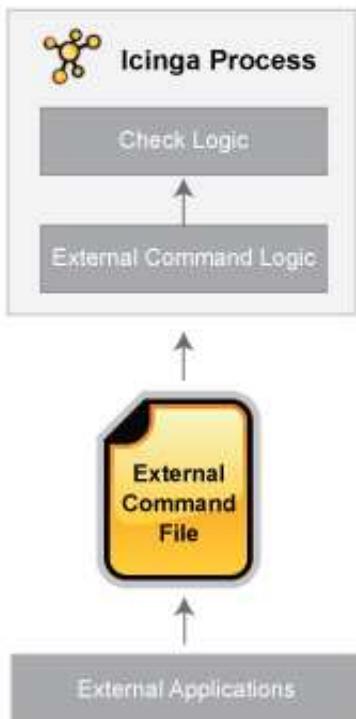
passive Prüfungen sind nützlich, um Services zu überwachen, die

- von Natur aus asynchron sind und nicht effektiv durch Abfrage ihres Zustands auf einer regelmäßig geplanten Basis überwacht werden können
- sich hinter einer Firewall befinden und nicht aktiv vom überwachenden Host aus geprüft werden können

Beispiele für asynchrone Services, bei denen sich eine passive Überwachung lohnt, sind u.a. SNMP-Traps und Sicherheits-Alarme. Sie wissen nie, wie viele (falls überhaupt) Traps oder Alarme Sie innerhalb eines vorgegebenen Zeitfensters erhalten, so dass es nicht sinnvoll ist, ihren Status alle paar Minuten zu überwachen.

Passive Prüfungen werden auch genutzt, um [verteilte](#) oder [redundante](#) Überwachungsinstallationen zu konfigurieren.

Wie passive Prüfungen arbeiten



Hier nun mehr Details, wie passive Prüfungen arbeiten...

1. eine externe Applikation prüft den Status eines Hosts oder Service.
2. die externe Applikation schreibt die Ergebnisse der Prüfung in das [external command file](#).
3. das nächste Mal, wenn Icinga das "external command file" liest, wird es die Ergebnisse aller passiven Prüfungen zur späteren Verarbeitung in eine Queue stellen. Dieselbe Queue, die für die Speicherung von Ergebnissen von aktiven Prüfungen genutzt wird, wird auch für die Speicherung von Ergebnissen von passiven Prüfungen verwendet.
4. Icinga wird periodisch ein [check result reaper event](#) ausführen und die Ergebnis-Queue abfragen. Jedes Service-Prüfungs-Ergebnis, das in der Queue gefunden wird, wird in der gleichen Weise bearbeitet - unabhängig davon, ob die Prüfung aktiv oder passiv war. Icinga kann abhängig vom Prüfergebnis Benachrichtigungen senden, Alarne protokollieren, usw.

Die Verarbeitung von aktiven und passiven Prüfungsergebnissen ist tatsächlich identisch. Dies erlaubt eine nahtlose Integration von externen Applikationen mit Icinga.

Passive Prüfungen aktivieren

Um passive Prüfungen in Icinga zu aktivieren, müssen Sie folgendes tun:

- setzen Sie die [accept_passive_service_checks](#)-Direktive auf 1.
- setzen Sie die [passive_checks_enabled](#)-Direktive in Ihren Host- und Service-Definitionen auf 1.

Wenn Sie die Verarbeitung von passiven Prüfungen global deaktivieren wollen, setzen Sie die [accept_passive_service_checks](#)-Direktive auf 0.

Wenn Sie die Verarbeitung von passiven Prüfungen nur für ein paar Hosts oder Services deaktivieren wollen, nutzen Sie die `passive_checks_enabled`-Direktive in den Host- und/oder Service-Definitionen.

Übermitteln von passiven Service-Prüfungsergebnissen

Externe Applikationen können passive Prüfungsergebnisse an Icinga übermitteln, indem sie ein `PROCESS_SERVICE_CHECK_RESULT` `external command` in das "external command file" schreiben.

Das Format des Befehls lautet wie folgt:

```
[<Zeitstempel>] PROCESS_SERVICE_CHECK_RESULT;<host_name>;<svc_description>;<return_code>;<plugin_output>
```

wobei...

- *timestamp* ist die Zeit im `time_t`-Format (Sekunden seit der UNIX-Epoche), zu der die Service-Prüfung durchgeführt (oder übermittelt) wurde. Bitte beachten Sie das einzelne Leerzeichen nach der rechten Klammer.
- *host_name* ist der Kurzname des Hosts, der mit dem Service in der Service-Definition verbunden ist
- *svc_description* ist die Beschreibung des Service wie in der Service-Definition angegeben
- *return_code* ist der Return-Code der Prüfung (0=OK, 1=WARNING, 2=CRITICAL, 3=UNKNOWN)
- *plugin_output* ist die Textausgabe der Service-Prüfung (also die Ausgabe des Plugins)

 Anmerkung: ein Service muss in Icinga definiert sein, bevor Sie passive Prüfungen für ihn abliefern können! Icinga wird alle Prüfergebnisse für Services ignorieren, die nicht konfiguriert waren, bevor es das letzte Mal (neu) gestartet wurde.



Ein Beispiel-Shell-Script, wie man passive Service-Prüfungsergebnisse an Icinga übermittelt, finden Sie in der Dokumentation zu [sprunghaften Services](#).

Übermitteln von passiven Host-Prüfungsergebnissen

Externe Applikationen können passive Host-Prüfungsergebnisse an Icinga übermitteln, indem sie ein `PROCESS_HOST_CHECK_RESULT` `external command` in das "external command file" schreiben.

Das Format des Befehls lautet wie folgt:

```
[<timestamp>] PROCESS_HOST_CHECK_RESULT;<host_name>;<host_status>;<plugin_output>
```

wobei...

- *timestamp* ist die Zeit im `time_t`-Format (Sekunden seit der UNIX-Epoche), zu der die Host-Prüfung durchgeführt (oder übermittelt) wurde. Bitte beachten Sie das einzelne Leerzeichen nach der rechten Klammer.
- *host_name* ist der Kurzname des Hosts (wie in der Host-Definition angegeben)

- *host_status* ist der Status des Hosts (0=UP, 1=DOWN, 2=UNREACHABLE)
- *plugin_output* ist die Textausgabe der Host-Prüfung (also die Ausgabe des Plugins)

 Anmerkung: ein Host muss in Icinga definiert sein, bevor Sie passive Prüfungen für ihn ablefern können! Icinga wird alle Prüfergebnisse für Hosts ignorieren, die nicht konfiguriert waren, bevor es das letzte Mal (neu) gestartet wurde.

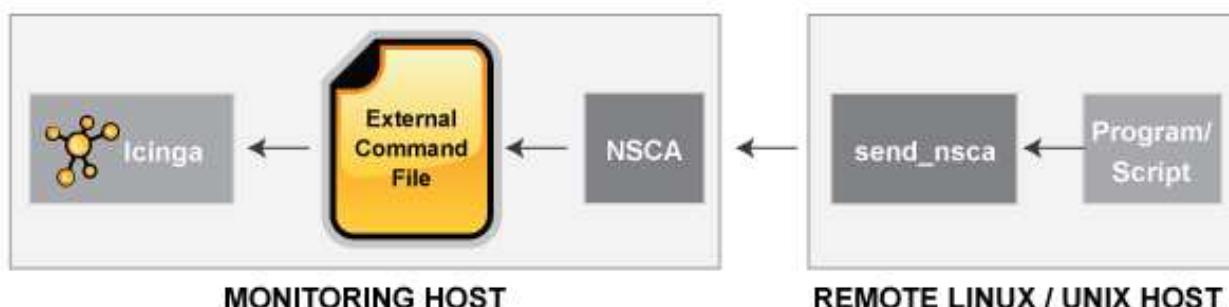
Passive Prüfungen und Host-Zustände

Icinga versucht bei passiven Prüfungen - anders bei aktiven Prüfungen - nicht festzustellen, ob der Host DOWN oder UNREACHABLE ist. Statt dessen nimmt Icinga das passive Prüfergebnis als den wahren Status des Hosts und versucht nicht, den wahren Host-Status mit Hilfe der [Erreichbarkeitslogik](#) zu ermitteln. Dies kann Probleme verursachen, wenn Sie passive Prüfungen von einem entfernten Host übermitteln oder Sie ein [verteiltes Überwachungs-Setup](#) haben, in dem Eltern/Kind-Verhältnisse unterschiedlich sind.

Sie können Icinga anweisen, die passiven Prüfergebnisse DOWN/UNREACHABLE-Zustände mit Hilfe der [translate_passive_host_checks](#)-Variable in ihre "sauberer" Zustände zu übersetzen. Mehr Informationen wie dies funktioniert, finden Sie [hier](#).

 Anmerkung: Passive Host-Prüfungen werden normalerweise als [HARD-Zustände](#) behandelt, falls nicht die [passive_host_checks_are_soft](#)-Option aktiviert ist.

Übermitteln von passiven Prüfungsergebnissen von entfernten Hosts



Wenn eine Applikation, die sich auf dem gleichen Host wie Icinga befindet, passive Host- oder Service-Prüfungsergebnisse sendet, kann es die Ergebnisse einfach direkt in das "external command file" schreiben wie oben skizziert. Allerdings können entfernte Hosts das nicht so einfach tun.

Um es entfernten Hosts zu erlauben, passive Prüfungsergebnisse an den überwachenden Host zu senden, hat Ethan Galstad das [NSCA](#)-Addon entwickelt. Das NSCA-Addon besteht aus einem Daemon, der auf dem Icinga-Host läuft und einem Client, der auf entfernten Hosts ausgeführt wird. Der Daemon lauscht auf Verbindungen von entfernten Hosts, führt mit den Ergebnissen einige grundlegende Gültigkeitsprüfungen durch und schreibt die Prüfergebnisse direkt in das "external command file" (wie oben beschrieben). Mehr Informationen über das NSCA-Addon finden Sie [hier](#).

[Zurück](#)
[Nach oben](#)
[Weiter](#)
[Aktive Prüfungen \(Active Checks\)](#)
[Zum Anfang](#)
[Statustypen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Statustypen

[Zurück](#)

Kapitel 5. Die Grundlagen

[Weiter](#)

Statustypen

Einführung

Der aktuelle Status von überwachten Services und Host wird durch zwei Komponenten ermittelt:

- den Status des Service oder Host (d.h. OK, WARNING, UP, DOWN, etc.)
- den *Typ* des Zustands, in dem der Service oder Host ist

Es gibt zwei Statustypen in Icinga: SOFT- und HARD-Zustände. Diese Statustypen sind ein wichtiger Teil der Überwachungslogik, da sie zur Ermittlung dienen, wann [Eventhandler](#) ausgeführt und [Benachrichtigungen](#) zuerst versandt werden.

Dieses Dokument beschreibt den Unterschied zwischen SOFT- und HARD-Zuständen, wann sie auftreten und was passiert, wenn sie auftreten.

Service- und Host-Prüfungswiederholungen

Um falsche Alarne bei vorübergehenden Problemen zu verhindern, erlaubt Ihnen Icinga zu definieren, wie oft ein Service oder Host (erneut) geprüft werden soll, bevor es als "echtes" Problem angesehen werden soll. Dies wird durch die `max_check_attempts`-Option in den Host- und Service-Definitionen kontrolliert. Zu verstehen, wie Hosts und Services (erneut) geprüft werden, um festzustellen, ob ein echtes Problem besteht, ist wichtig zum Verstehen, wie Statustypen arbeiten.

Soft-Zustände

Soft-Zustände treten in den folgenden Situationen auf...

- wenn ein Host- oder Service-Prüfungsergebnis in einem nicht-OK oder nicht-UP-Status resultiert und die Service-Prüfung noch nicht so oft (erneut) durchgeführt wurde, wie es in der `max_check_attempts`-Direktive der Service- oder Host-Definition angegeben wurde. Das wird als Soft-Error bezeichnet.
- wenn sich ein Service oder Host von einem Soft-Error erholt. Das wird als Soft-Recovery angesehen.

Die folgenden Dinge passieren, wenn bei Hosts oder Services SOFT-Zustandsänderungen auftreten:

- der SOFT-Status wird protokolliert.
- Eventhandler werden zur Behandlung von SOFT-Zuständen ausgeführt

SOFT-Zustände werden nur protokolliert, wenn Sie die [log_service_retries](#)- oder die [log_host_retries](#)-Option in Ihrer Hauptkonfigurationsdatei aktiviert haben.

Das einzige Wichtige, was bei einem Soft-Zustand passiert, ist die Ausführung von Eventhandlern. Eventhandler zu benutzen kann insbesondere dann nützlich sein, wenn Sie versuchen wollen, proaktiv ein Problem zu lösen, bevor es sich in einen HARD-Zustand verwandelt. Die [\\$HOSTSTATETYPE\\$](#)- oder [\\$SERVICESTATETYPE\\$](#)-Makros werden den Wert "SOFT" haben, wenn Eventhandler ausgeführt werden, was es Ihren Eventhandlern erlaubt zu wissen, wann sie fehlerbehebende Aktionen vornehmen sollen. Mehr Informationen zu Eventhandlern finden Sie [hier](#).

Hard-Zustände

Hard-Zustände treten für Hosts und Services in den folgenden Situationen auf...

- wenn ein Host- oder Service-Prüfungsergebnis in einem nicht-OK oder nicht-UP-Status resultiert und die Prüfung bereits so oft (erneut) durchgeführt wurde, wie es in der *max_check_attempts*-Direktive der Service- oder Host-Definition angegeben wurde. Das wird als Hard-Error bezeichnet.
- wenn ein Host oder Service von einem Hard-Error-Zustand in einen anderen Fehlerzustand wechselt (z.B. von WARNING nach CRITICAL).
- wenn eine Service-Prüfung in einem nicht-OK-Status resultiert und der zugehörige Host entweder DOWN oder UNREACHABLE ist.
- wenn ein Host oder Service sich von einem Hard-Error-Zustand erholt. Dies wird als Hard-Recovery angesehen.
- wenn eine [passive Host-Prüfung](#) empfangen wird. Passive Host-Prüfungen werden als HARD angesehen, wenn nicht die [passive_host_checks_are_soft](#)-Option aktiviert ist.

Die folgenden Dinge passieren, wenn bei Hosts oder Services HARD-Zustandsänderungen auftreten:

- der HARD-Status wird protokolliert.
- Eventhandler werden zur Behandlung von HARD-Zuständen ausgeführt.
- Kontakte werden über das Host- oder Service-Problem bzw. die Erholung informiert.

Die [\\$HOSTSTATETYPE\\$](#) oder [\\$SERVICESTATETYPE\\$](#)-Makros werden den Wert "HARD" haben, wenn Eventhandler ausgeführt werden, was es Ihren Eventhandlern erlaubt zu wissen, wann sie fehlerbehebende Aktionen vornehmen sollen. Mehr Informationen zu Eventhandlern finden Sie [hier](#).

Beispiel

Hier ist ein Beispiel, wie Statustypen ermittelt werden, wenn Statusänderungen auftreten und wann Eventhandler ausgeführt und Benachrichtigungen versandt werden. Die nachfolgende Tabelle zeigt aufeinander folgende Prüfungen eines Service. Der Service hat einen *max_check_attempts*-Wert von 3.

Zeit	Prüfung #	Status	Statustyp	Statuswechsel	Anmerkungen
0	1	OK	HARD	Nein	Initialer Zustand des Service
1	1	CRITICAL	SOFT	Ja	erstes Erkennen eines nicht-OK-Zustandes. Eventhandler wird ausgeführt.
2	2	WARNING	SOFT	Ja	Service bleibt in einem nicht-OK-Zustand. Eventhandler wird ausgeführt.
3	3	CRITICAL	HARD	Ja	"max_check_attempts" wurde erreicht, deshalb geht der Service in einen HARD-Zustand. Eventhandler wird ausgeführt und eine Benachrichtigung versandt. Die Check-Anzahl wird auf 1 zurückgesetzt, sofort nachdem dies passiert.
4	1	WARNING	HARD	Ja	Service wechselt in einen HARD-WARNING-Status. Eventhandler wird ausgeführt und eine Problembenachrichtigung versandt.
5	1	WARNING	HARD	Nein	Service stabilisiert sich zu einem HARD-Problemzustand. Abhängig vom Benachrichtigungsintervall für den Service wird ggf. eine weitere Benachrichtigung verschickt.
6	1	OK	HARD	Ja	Service erfährt eine HARD-Recovery. Eventhandler wird ausgeführt und eine Erholungs-Benachrichtigung wird versandt.
7	1	OK	HARD	Nein	Service ist weiterhin OK.
8	1	UNKNOWN	SOFT	Ja	Für den Service wird ein Wechsel zu einem SOFT nicht-OK-Zustand festgestellt. Eventhandler wird ausgeführt.
9	2	OK	SOFT	Ja	Service erfährt eine SOFT-Recovery. Eventhandler wird ausgeführt, aber keine Benachrichtigung versandt, weil dies kein "echtes" Problem war. Der Statustyp wird auf HARD gesetzt und die Check-Anzahl auf 1 zurückgesetzt, sofort nachdem dies passiert.

10	1	OK	HARD	Nein	Service stabilisiert sind zu einem OK-Status.
----	---	----	------	------	---

[Zurück](#)[Nach oben](#)[Weiter](#)[Passive Prüfungen \(Passive Checks\)](#)[Zum Anfang](#)[Zeitfenster](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Zeitfenster

[Zurück](#)

Kapitel 5. Die Grundlagen

[Weiter](#)

Zeitfenster

oder...

Einführung



Zeitfenster-Definitionen erlauben Ihnen zu kontrollieren, wann verschiedene Aspekte der Überwachungs- und Alarmierungslogik arbeiten. Zum Beispiel können Sie einschränken

- wann regelmäßig geplante Host- und Service-Prüfungen ausgeführt werden
- wann Benachrichtigungen versandt werden
- wann Benachrichtigungs-Eskalationen benutzt werden können
- wann Abhängigkeiten gültig sind

Vorrang bei Zeitfenstern

Zeitfenster-**Definitionen** können mehrere Typen von Direktiven enthalten, einschließlich Wochentagen, Monatstagen und Kalenderdaten. Verschiedene Typen von Direktiven haben unterschiedliche Vorrang-Ebenen und können andere Direktiven in Ihren Zeitfenster-Definitionen außer Kraft setzen. Die Rangfolge für verschiedene Typen von Direktiven (in absteigender Reihenfolge) ist wie folgt:

- Kalenderdaten (2008-01-01)
- angegebener Tag des Monats (January 1st)
- generischer Tag des Monats (Day 15)
- Offset Wochentag eines bestimmten Monats (2nd Tuesday in December)
- Offset Wochentag (3rd Monday)

- normaler Wochentag (Tuesday)

Beispiele für verschiedene Zeitfenster-Direktiven finden Sie [hier](#).

Wie Zeitfenster mit Host- und Service-Prüfungen arbeiten

Host- und Service-Definitionen haben eine optionale *check_period*-Direktive, die es Ihnen erlaubt, ein Zeitfenster anzugeben, das zur Einschränkung benutzt werden sollte, wann regelmäßig geplante aktive Prüfungen des Hosts oder Service stattfinden.

Wenn Sie die *check_period*-Direktive nicht nutzen, um ein Zeitfenster anzugeben, wird Icinga in der Lage sein, aktive Prüfungen für den Host oder Service zu jeder Zeit zu planen, wenn es nötig ist. Dies ist in Wirklichkeit ein 24x7-Überwachungsszenario.

Ein Zeitfenster in der *check_period*-Direktive anzugeben erlaubt Ihnen die Einschränkung der Zeit, wann Icinga regelmäßige aktive Host- oder Service-Prüfungen plant. Wenn Icinga versucht, einen Host oder Service neu zu planen, wird es sicherstellen, dass die nächste Prüfung in einen gültigen Zeitbereich im definierten Zeitfenster fällt. Falls das nicht zutreffen sollte, wird Icinga die Zeit der nächsten Prüfung so anpassen, dass sie in die nächste "gültige" Zeit im angegebenen Zeitfenster fällt. Das bedeutet, dass der Host oder Service vielleicht während der nächsten Stunde, des nächsten Tages oder der nächsten Woche, etc. nicht geprüft wird.

 Anmerkung: Prüfungen nach Bedarf und passive Prüfungen sind nicht durch das Zeitfenster beschränkt, das Sie in der *check_period*-Direktive angeben. Nur regelmäßig geplante aktive Prüfungen werden beschränkt.

Außer Sie haben einen guten Grund das zu tun, würden wir raten, dass Sie all Ihre Hosts und Services mit einem Zeitfenster überwachen, das einen 24x7-Zeitbereich abdeckt. Falls Sie das nicht tun, können Sie während der "blackout"-Zeiten in einige Probleme laufen (Zeiten, die nicht gültig sind in der Zeitfenster-Definition):

1. der Status des Hosts oder Service wird in der blackout-Zeit unverändert erscheinen.
2. Kontakte werden während der blackout-Zeit wahrscheinlich nicht erneut über Host- oder Service-Probleme informiert werden.
3. falls sich ein Host oder Service während einer blackout-Zeit erholt, werden Kontakte nicht umgehend über die Erholung informiert.

Wie Zeitfenster mit Kontakt-Benachrichtigungen arbeiten

Durch das Angeben eines Zeitfensters in der *notification_period*-Direktive einer Host- oder Service-Definition kontrollieren Sie, wann Icinga Benachrichtigungen versenden darf, um über Probleme oder Erholungen für den Host oder Service zu informieren. Wenn eine Host-Benachrichtigung versandt werden soll, prüft Icinga, ob die aktuelle Zeit in einem gültigen Bereich der *notification_period* liegt. Wenn eine gültige Zeit vorliegt, wird Icinga versuchen, jeden Kontakt über das Problem oder die Erholung zu informieren.

Sie können Zeitfenster auch nutzen, um zu kontrollieren, wann Benachrichtigungen an einzelne Kontakte versandt werden. Durch die Nutzung der *service_notification_period*- und der *host_notification_period*-Direktiven in den [Kontakt-Definitionen](#) sind Sie in der Lage, eine tatsächliche Rufbereitschaft für jeden Kontakt zu definieren. Kontakte werden Host- und Service-Benachrichtigungen nur während der Zeiten erhalten, die Sie in den Benachrichtigungs-Direktiven angegeben haben.

Beispiele, wie Zeitfenster-Definitionen für Rufbereitschafts-Wechsel angelegt werden, finden Sie [hier](#).

Wie Zeitfenster mit Benachrichtigungs-Eskalationen arbeiten

Service- und Host-[Benachrichtigungs-Eskalationen](#) haben eine optionale *escalation_period*-Direktive, die es Ihnen erlaubt ein Zeitfenster anzugeben, wann die Eskalation gültig ist und benutzt werden kann. Wenn Sie die *escalation_period*-Direktive nicht in einer Eskalations-Definition benutzen, ist diese Eskalation zu allen Zeiten gültig. Wenn Sie ein Zeitfenster in der *escalation_period*-Direktive angeben, wird Icinga die Eskalations-Definition nur zu Zeiten nutzen, die aufgrund der Zeitfenster-Definition gültig sind.

Wie Zeitfenster mit Abhängigkeiten arbeiten

Service- und Host-[Abhängigkeiten](#) haben eine optionale *dependency_period*-Direktive, die es Ihnen erlaubt ein Zeitfenster anzugeben, wann die Abhängigkeit gültig ist und benutzt werden kann. Wenn Sie die *dependency_period*-Direktive nicht in einer Abhängigkeits-Definition benutzen, ist diese Abhängigkeit zu allen Zeiten gültig. Wenn Sie ein Zeitfenster in der *dependency_period*-Direktive angeben, wird Icinga die Abhängigkeits-Definition nur zu Zeiten nutzen, die aufgrund der Zeitfenster-Definition gültig sind.

[Zurück](#)[Nach oben](#)[Weiter](#)[Statustypen](#)[Zum Anfang](#)[Ermitteln des Zustands und der Erreichbarkeit von Netzwerk-Hosts](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Ermitteln des Zustands und der Erreichbarkeit von Netzwerk-Hosts

[Zurück](#)[Kapitel 5. Die Grundlagen](#)[Weiter](#)

Ermitteln des Zustands und der Erreichbarkeit von Netzwerk-Hosts

Einführung

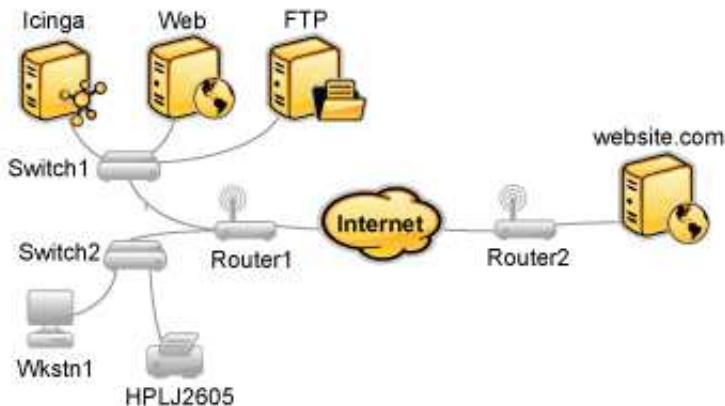
Falls Sie jemals im technischen Support gearbeitet haben, hatten Sie zweifelsohne Benutzer, die Ihnen erzählt haben, "das Internet sei down". Als Techniker waren Sie ziemlich sicher, daß keiner den Stromstecker aus dem Internet gezogen hatte. Irgendetwas muss schiefgehen zwischen dem Stuhl des Benutzers und dem Internet.

Angenommen es ist ein technisches Problem, dann werden Sie nach dem Problem suchen. Vielleicht ist der PC des Benutzers ausgeschaltet oder das Netzwerkkabel ist gezogen oder der zentrale Router Ihres Unternehmens nimmt gerade eine Auszeit. Was immer das Problem sein mag, eines ist sehr sicher - das Internet ist nicht down. Es ist lediglich nicht für den Benutzer erreichbar.

Icinga ist in der Lage festzustellen, ob die Hosts, die Sie überwachen, in einem DOWN- oder UNREACHABLE-Zustand sind. Dies sind sehr unterschiedliche (obwohl durchaus verwandte) Zustände und können Ihnen helfen, schnell die Grundursache für Netzwerkprobleme festzustellen. Hier nun, wie die Netzwerk-Erreichbarkeitslogik arbeitet, um zwischen diesen beiden Zuständen zu unterscheiden...

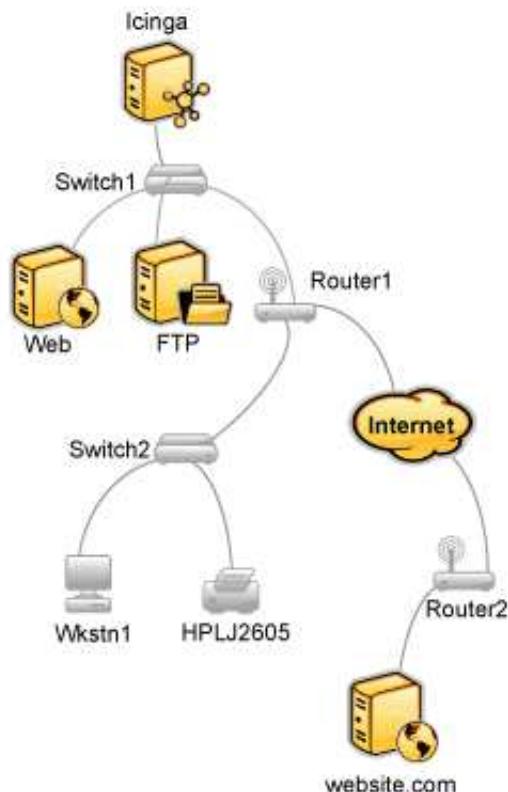
Beispiel-Netzwerk

Werfen Sie einen Blick auf das einfache Netzwerk-Diagramm. Lassen Sie uns annehmen, dass Sie alle Hosts (Server, Router, Switches, etc.) überwachen, die abgebildet sind. Icinga ist installiert und lauffähig auf dem *Icinga*-Host.



Definieren von Eltern/Kind-Beziehungen

Um Icinga in die Lage zu versetzen, zwischen DOWN und UNREACHABLE-Zuständen der überwachten Hosts zu unterscheiden, müssen Sie Icinga mitteilen, wie diese Hosts miteinander verbunden sind - vom Standpunkt des Icinga-Daemons aus gesehen. Um dies zu tun verfolgen Sie den Weg, den ein Datenpaket vom Icinga-Daemon zu jedem einzelnen Host nehmen würde. Jeder Switch, Router und Server, den das Paket trifft oder passiert, wird als "Hop" angesehen und erfordert, dass Sie eine Eltern/Kind-Beziehung in Icinga definieren. Hier nun, wie die Host-Eltern/Kind-Beziehung aus der Sicht von Icinga aussieht:



Nun, da Sie wissen, wie die Eltern/Kind-Beziehungen für überwachte Hosts aussehen, wie konfigurieren Sie Icinga, um sie abzubilden? Die [parents-Direktive](#) in Ihren [Host-Definitionen](#) erlaubt Ihnen, das zu tun. Hier nun, wie die (verkürzten) Host-Definitionen mit Eltern/Kind-Beziehung für dieses Beispiel aussehen würden:

```

define host{
    host_name           Icinga      ; <-- der lokale Host hat keine Eltern - es ist der am weitesten oben stehende Host
}
define host{
    host_name           Switch1
    parents             Icinga
}
define host{
    host_name           Web
    parents             Switch1
}
define host{
    host_name           FTP
    parents             Switch1
}
define host{
    host_name           Router1
    parents             Switch1
}
define host{
    host_name           Switch2
    parents             Router1
}
define host{
    host_name           Wkstn1
    parents             Switch2
}
define host{
    host_name           HPLJ2605
}
  
```

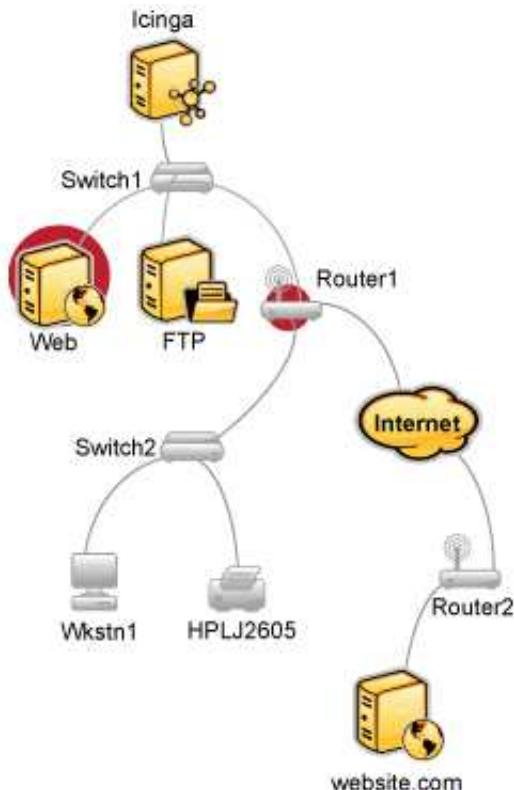
```

parents      Switch2
}
define host{
host_name   Router2
parents     Router1
}
define host{
host_name   somewebsite.com
parents     Router2
}

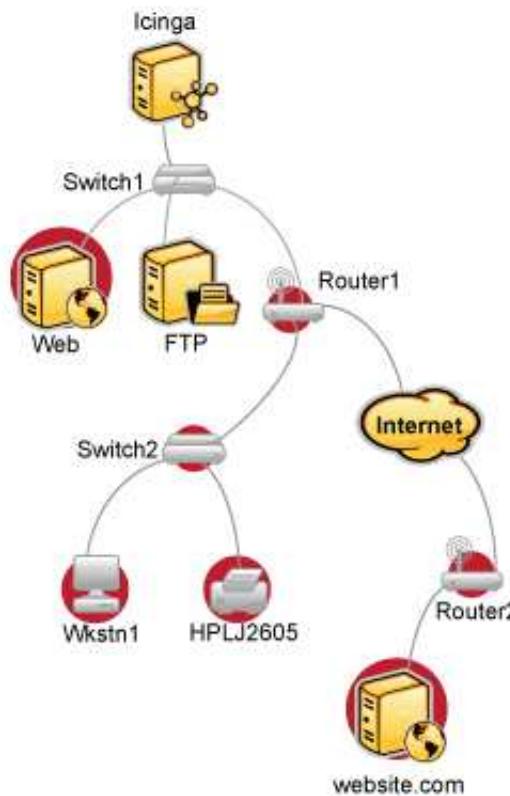
```

Erreichbarkeits-Logik in Aktion

Nachdem Sie Icinga mit den passenden Eltern/Kind-Beziehungen konfiguriert haben, lassen Sie uns sehen, was passiert, wenn Probleme auftauchen. Nehmen Sie an, dass zwei Hosts, *Web* und *Router1*, offline gehen...



Wenn Hosts den Status wechseln (d.h. von UP zu DOWN) wird die Host-Erreichbarkeitslogik in Icinga anspringen. Die Erreichbarkeits-Logik wird parallele Prüfungen der Eltern und Kinder aller Hosts veranlassen, deren Status sich ändert. Dies erlaubt es Icinga schnell den aktuellen Status Ihrer Netzwerk-Infrastruktur zu ermitteln, wenn Änderungen auftreten.



In diesem Beispiel wird Icinga feststellen, dass *Web* und *Router1* beide im DOWN-Status sind, weil der "Pfad" zu diesen Hosts nicht blockiert ist.

Icinga wird feststellen, dass alle Hosts "unterhalb" *Router1* alle in einem UNREACHABLE Status sind, weil Icinga sie nicht erreichen kann. *Router1* ist DOWN und blockiert den Weg zu diesen anderen Hosts. Diese Hosts können wunderbar funktionieren oder offline sein - Icinga weiß es nicht, weil es sie nicht erreichen kann. Deshalb wird Icinga sie als UNREACHABLE ansehen anstatt DOWN.

UNREACHABLE Zustände und Benachrichtigungen

Standardmäßig wird Icinga Kontakte über Hosts im DOWN und UNREACHABLE-Status informieren. Als ein Admin/Techniker möchten Sie vielleicht keine Benachrichtigungen über Hosts erhalten, die UNREACHABLE sind. Sie kennen Ihre Netzwerkstruktur und wenn Icinga Sie informiert, dass der Router/die Firewall unten ist, dann wissen Sie, dass alles dahinter nicht erreichbar ist.

Falls Sie sich eine Flut von Benachrichtigungen über UNREACHABLE-Zustände während eines Netzwerkausfalls ersparen möchten, können Sie die `reachable` (u)-Option der `notification_options`-Direktive in Ihren `Host`-Definitionen und/oder die `host_notification_options`-Direktive in Ihren `Kontakt`-Direktiven ausschließen.

[Zurück](#)
[Nach oben](#)
[Weiter](#)
[Zeitfenster](#)
[Zum Anfang](#)
[Benachrichtigungen](#)



Benachrichtigungen

[Zurück](#)

[Kapitel 5. Die Grundlagen](#)

[Weiter](#)

Benachrichtigungen

Einführung



Es gab eine Menge Fragen, wie genau Benachrichtigungen arbeiten. Wir werden versuchen, genau zu erklären, wann und wie Host- und Service-Benachrichtigungen versandt werden und ebenso, wer sie bekommt.

Benachrichtigungs-Eskalationen werden [hier](#) beschrieben.

Wann erfolgen Benachrichtigungen?

Die Entscheidung, Benachrichtigungen zu senden, wird in der Service- und Host-Prüflogik getroffen. Die Ermittlung, ob eine Benachrichtigung versandt wird oder nicht, erfolgt nur dann, wenn eine Host- oder Service-Prüfung zu dieser Benachrichtigung verarbeitet wird. Es reicht nicht, dass die in der Direktive `<notification_interval>` angegebene Zeit seit der letzten Benachrichtigung vergangen ist. Host- und Service-Benachrichtigungen erfolgen in den folgenden Fällen...

- wenn ein HARD-Statuswechsel erfolgt. Mehr Informationen über Statustypen und Hard-Statuswechsel finden Sie [hier](#).
- wenn ein Host oder Service in einem Hard nicht-OK-Zustand bleibt und die in der `<notification_interval>`-Option der Host- oder Service-Definition angegebene Zeit seit der letzten versandten Benachrichtigung verstrichen ist (für den angegebenen Host oder Service).

Wer wird benachrichtigt?

Jede Host- und Service-Definition hat eine `<contact_groups>`-Option, die angibt, welche Kontaktgruppen Benachrichtigungen für bestimmte Hosts oder Services erhalten. Kontaktgruppen können ein oder mehrere einzelne Kontakte enthalten.

Wenn Icinga eine Host- oder Service-Benachrichtigung versendet, wird es jeden Kontakt informieren, der Mitglied in einer der Kontaktgruppen ist, die in der `<contactgroups>`-Option der Service-Definition angegeben ist. Icinga bemerkt, wenn ein Kontakt Mitglied von mehr als einer Kontaktgruppe ist und entfernt mehrfache Kontaktbenachrichtigungen, bevor es irgendetwas tut.

Welche Filter müssen durchlaufen werden, damit Benachrichtigungen versandt werden?

Nur weil Benachrichtigungen für einen Host- oder Service versandt werden müssen, bedeutet das nicht, dass irgendein Kontakt informiert wird. Es gibt mehrere Filter, die potenzielle Benachrichtigungen durchlaufen müssen, bevor sie als würdig genug angesehen werden, um versandt zu werden. Lassen Sie uns einen genaueren Blick auf die Filter werfen, die zu durchlaufen sind...

Programmweite Filter:

Der erste Filter, den Benachrichtigungen durchlaufen müssen, ist ein Test, ob Benachrichtigungen auf einer programmweiten Basis aktiviert sind. Dies wird ursprünglich durch die `enable_notifications`-Option in der Hauptkonfigurationsdatei festgelegt, kann aber während der Laufzeit über das Web-Interface verändert werden. Falls Benachrichtigungen auf programmweiter Basis deaktiviert sind, werden keine Benachrichtigungen für Hosts oder Services versandt - Punkt. Wenn sie auf programmweiter Basis aktiviert sind, müssen weitere Tests durchlaufen werden...

Service- und Host-Filter:

Der erste Filter für Host- oder Service-Benachrichtigungen ist eine Prüfung, ob sich der Host oder Service in einer [geplanten Ausfallzeit](#) (downtime) befindet. Falls es eine geplante Ausfallzeit ist, **wird niemand informiert**. Wenn es keine Ausfallzeit ist, geht es weiter zum nächsten Filter. Als kleine Randnotiz: Service-Benachrichtigungen werden unterdrückt, falls sich der mit ihnen verbundene Host in einer geplanten Ausfallzeit befindet.

Der zweite Filter für Host- oder Service-Benachrichtigungen ist eine Prüfung, ob der Host oder Service [flattert](#) (wenn Sie Flatter-Erkennung aktiviert haben). Falls der Service oder Host gerade flattert, **wird niemand informiert**. Andernfalls geht es weiter zum nächsten Filter.

Der dritte für Hosts oder Services zu durchlaufende Filter sind die Host- oder Service-spezifischen Benachrichtigungsoptionen. Jede Service-Definition enthält Optionen, die festlegen, ob Benachrichtigungen für Warnungen, kritische Zustände oder Erholungen versandt werden oder nicht. Ähnlich ist es bei Hosts, wo festgelegt wird, ob Benachrichtigungen versandt werden, wenn der Host down geht, unerreichbar wird oder sich wieder erholt. Falls die Host- oder Service-Benachrichtigungen diese Optionen nicht passieren, **wird niemand informiert**. Wenn sie die Optionen durchlaufen, geht es zum nächsten Filter... Anmerkung: Benachrichtigungen über Host- oder Service-Erholungen werden nur dann versandt, wenn auch eine Benachrichtigung über das ursprüngliche Problem versandt wurde. Es ist nicht sinnvoll, eine Benachrichtigung über eine Erholung zu bekommen, wenn Sie nicht wussten, dass ein Problem existiert.

Der vierte Host- oder Service-Filter, der durchlaufen werden muss, ist der Zeitfenster-Test. Jede Host- und Service-Definition hat eine `<notification_period>`-Option, die angibt, welches Zeitfenster gültige Benachrichtigungszeiten für den Host oder Service enthält. Wenn die Zeit der Benachrichtigung nicht in einen gültigen Bereich des Zeitfensters fällt, **wird niemand informiert**. Wenn sie in einen gültigen Bereich fällt, geht es zum nächsten Filter... Anmerkung: falls der Zeitfenster-Filter nicht erfolgreich durchlaufen wird, plant Icinga die nächste Benachrichtigung für den Host oder Service (falls er sich in einem nicht-OK-Status befindet) für die nächste verfügbare gültige Zeit im Zeitfenster. Dies stellt sicher, dass der Kontakt so früh

wie möglich über Probleme informiert wird, wenn die nächste gültige Zeit erreicht wird.

Der letzte Satz von Host- oder Service-Filter ist abhängig von zwei Dingen: (1) zu einem Zeitpunkt in der Vergangenheit wurde bereits eine Benachrichtigung über ein Problem mit dem Host oder Service versandt und (2) blieb der Host oder Service im gleichen nicht-OK-Zustand, der zur Zeit der Benachrichtigung vorlag. Wenn diese beiden Kriterien zutreffen, wird Icinga prüfen und sicherstellen, dass die seit der letzten Benachrichtigung vergangene Zeit den in der Option `<notification_interval>` angegebenen Wert in der Host- oder Service-Definition erreicht oder übertrifft. Falls nicht genug Zeit seit der letzten Benachrichtigung vergangen ist, **wird niemand benachrichtigt**. Wenn entweder genug Zeit seit der letzten Benachrichtigung vergangen ist oder die beiden Kriterien dieses Filters erfüllt wurden, wird die Benachrichtigung versandt. Ob sie tatsächlich an einzelne Kontakte versandt wird, hängt von einem weiteren Satz von Filtern ab...

Kontakt-Filter:

An diesem Punkt hat die Benachrichtigung die programmweiten und alle Host- und Service-Filter durchlaufen und Icinga beginnt, **alle betroffenen Leute zu informieren**. Bedeutet dies, dass jeder Kontakt die Benachrichtigung erhalten wird? Nein. Jeder Kontakt hat seinen eigenen Satz von Filtern, den die Benachrichtigung passieren muss. Anmerkung: Kontaktfilter sind spezifisch für jeden Kontakt und beeinflussen nicht, ob andere Kontakte Benachrichtigungen erhalten oder nicht.

Der erste zu passierende Filter für jeden Kontakt sind die Benachrichtigungsoptionen. Jede Kontaktdefinition enthält Optionen, die festlegen, ob Service-Benachrichtigungen für Warning- und Critical-Zustände und Erholungen versandt werden können. Jede Kontakt-Definition enthält auch Optionen, die festlegen, ob Host-Benachrichtigungen versandt werden, wenn der Host "down" geht, unerreichbar wird oder sich erholt. Falls die Host- oder Service-Benachrichtigung diese Optionen nicht passieren kann, **wird der Kontakt nicht informiert**. Wenn es diese Optionen passiert, wird die Benachrichtigung an den nächsten Filter weitergereicht... Anmerkung: Benachrichtigungen über die Erholung von Host oder Service werden nur dann versandt, wenn eine Benachrichtigung für das ursprüngliche Problem versandt wurde. Es ist sinnlos, eine Benachrichtigung über eine Erholung zu versenden, wenn Sie nicht wussten, dass ein Problem existiert...

Der letzte zu passierende Filter für jeden Kontakt ist der Zeitfenster-Test. Jede Kontaktdefinition hat eine `<notification_period>`-Option, die angibt, welches Zeitfenster gültige Benachrichtigungszeiten für den Kontakt enthält. Wenn die Zeit, in der die Benachrichtigung erstellt wird, nicht in ein gültiges Zeitfenster fällt, **wird der Kontakt nicht informiert**. Wenn sie in ein gültiges Zeitfenster fällt, wird der Kontakt informiert!

Benachrichtigungs-Methoden

Icinga kann Sie über Probleme und Erholungen auf vielfältige Weise informieren: Pager, Handy, e-Mail, SMS, Audio-Hinweis usw. Wie Benachrichtigungen versandt werden, hängt von den **Benachrichtigungs-Befehlen** ab, die in Ihren **Objekt-Definitionsdateien** definiert werden.

 Anmerkung: Wenn Sie Icinga nach den [Schnellstart-Anleitungen](#) installieren, sollte es zum Versand von e-Mail-Benachrichtigungen konfiguriert sein. Sie können die benutzten e-Mail-Befehle ansehen, indem Sie den Inhalt der Datei `/usr/local/icinga/etc/objects/commands.cfg` betrachten.

Spezielle Benachrichtigungs-Methoden (Paging usw.) sind nicht direkt in den Icinga-Code integriert, denn es ist nicht sinnvoll. Der "Kern" von Icinga ist nicht als eierlegende Wollmilchsau gedacht. Wenn Service-Prüfungen im Icinga-Kern enthalten wären, hätten Benutzer große Schwierigkeiten, neue Prüfmethoden hinzuzufügen, bestehende Prüfungen zu modifizieren usw. Benachrichtigungen arbeiten in ähnlicher Weise. Es gibt tausend verschiedene Wege, Benachrichtigungen zu versenden und es gibt bereits viele Pakete, die die schmutzige Arbeit tun, also warum das Rad neu erfinden und sich dann auf einen Fahrrad-Reifen beschränken? Es ist viel einfacher, ein externes Gebilde (das kann ein einfaches Script sein oder ein ausgewachsenes Message-System) die ganze Arbeit tun zu lassen. Einige Message-Pakete, die Benachrichtigungen für Pager und Handys verarbeiten können, sind weiter unten aufgeführt.

Benachrichtigungstyp-Makro

Wenn Sie Benachrichtigungs-Befehle erstellen, müssen Sie beachten, um welchen Typ von Benachrichtigung es sich handelt. Das Makro `$NOTIFICATIONTYPES$` enthält eine Zeichenkette, die genau das angibt. Die nachfolgende Tabelle zeigt die möglichen Werte und deren entsprechende Beschreibungen:

Wert	Beschreibung
PROBLEM	Ein Host oder Service hat gerade einen Problemzustand erreicht (oder ist noch in einem). Wenn dies eine Service-Benachrichtigung ist, bedeutet das, dass der Service in einem WARNING-, UNKNOWN- oder CRITICAL-Zustand ist. Wenn dies eine Host-Benachrichtigung ist, bedeutet das, dass der Host in einem DOWN- oder UNREACHABLE-Zustand ist.
RECOVERY	Ein Service oder Host hat sich erholt. Wenn dies eine Service-Benachrichtigung ist, bedeutet es, dass der Service gerade wieder in einen OK-Zustand zurückgekehrt ist. Wenn dies eine Host-Benachrichtigung ist, bedeutet das, dass der Host gerade wieder in einen UP-Zustand zurückgekehrt ist.
ACKNOWLEDGEMENT	Diese Benachrichtigung ist eine Bestätigung für ein Host- oder Service-Problem. Bestätigungen werden von Kontakten für diesen Host oder Service über das Web-Interface ausgelöst.
FLAPPINGSTART	Der Host oder Service hat gerade angefangen zu flattern .
FLAPPINGSTOP	Der Host oder Service hat gerade aufgehört zu flattern .
FLAPPINGDISABLED	Der Host oder Service hat gerade aufgehört zu flattern , weil die Flatter-Erkennung deaktiviert wurde.
DOWNTIMESTART	Der Host oder Service hat gerade ein geplante Downtime begonnen. Weitere Benachrichtigungen werden unterdrückt.
DOWNTIMESTOP	Der Host oder Service hat gerade eine geplante Downtime beendet. Benachrichtigungen über Probleme werden wieder versandt.
DOWNTIMECANCELLED	Die Phase der geplanten Downtime für den Host oder Service wurde gerade annulliert. Benachrichtigungen über Probleme werden wieder versandt.

Hilfreiche Quellen

Es gibt viele Wege, wie Sie Icinga konfigurieren können, damit Benachrichtigungen versandt werden. Sobald Sie dies tun, müssen Sie notwendige Software installieren und Benachrichtigungs-Befehle konfigurieren, bevor Sie diese benutzen können. Hier sind nur ein paar mögliche Benachrichtigungs-Methoden:

- e-Mail
- Pager
- Telefon (SMS)
- WinPopup-Meldung
- Yahoo-, ICQ- oder MSN-Sofortnachricht
- Audio-Hinweise
- etc...

Im Grunde genommen kann alles, was Sie von einer Kommandozeile aus tun können, so angepasst werden, dass Sie es in einem Benachrichtigungs-Befehl nutzen können.

Wenn Sie nach einer Alternative suchen, um Meldungen per e-Mail an Ihren Pager oder Ihr Handy zu versenden, sollten Sie diese Pakete ausprobieren. Sie können in Verbindung mit Icinga dazu benutzt werden, Benachrichtigungen über ein Modem zu versenden, wenn ein Problem auftritt. Auf diese Weise müssen Sie sich nicht auf e-Mail verlassen, um Benachrichtigungen zu versenden (bedenken Sie, dass e-Mail ggf. *nicht* funktioniert, wenn es ein Netzwerk-Problem gibt). Wir haben diese Pakete nicht selbst ausprobiert, aber andere haben von erfolgreichem Einsatz berichtet...

- [Gnokii](#) (SMS-Software, um Nokia-Telefone über das GSM-Netzwerk zu erreichen)
- [QuickPage](#) (Alphanumerische Pager-Software)
- [Sendpage](#) (Paging-Software)
- [SMS Client](#) (Kommandozeilen-Utility, um Meldungen auf Pager und Mobiltelefone zu senden)

Wenn Sie eine nicht-traditionelle Methode für Benachrichtigungen ausprobieren möchten, können Sie ggf. Audio-Hinweise nutzen. Wenn Sie Audio-Hinweise auf dem Überwachungs-Rechner (mit synthetischer Stimme) abspielen möchten, probieren Sie [Festival](#). Wenn Sie den Überwachungs-Rechner lieber in Ruhe lassen und Audio-Hinweise auf einem anderen Rechner abspielen möchten, dann sehen Sie sich die Projekte [Network Audio System \(NAS\)](#) und [rplay](#) an.

[Zurück](#)

Ermitteln des Zustands und der
Erreichbarkeit von
Netzwerk-Hosts

[Nach oben](#)

[Zum Anfang](#)

[Weiter](#)

Kapitel 6. Die
Benutzeroberflächen

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Kapitel 6. Die Benutzeroberflächen

[Zurück](#)

[Weiter](#)

Kapitel 6. Die Benutzeroberflächen

Inhaltsverzeichnis

- [Icinga Classic UI: Informationen über die CGIs](#)
 - [Informationen zu den CGI-Parametern](#)
 - [Ausführen von CGIs auf der Kommandzeile](#)
 - [Installation des Icinga-Web Frontend](#)
 - [Konfigurationsübersicht Icinga-Web](#)
 - [Aktualisierung von Icinga-Web und Icinga-Web Datenbank](#)
 - [Einführung in Icinga-Web](#)
 - [Einführung in Icinga-Web \(>= 1.6\)](#)
 - [Integration von PNP4Nagios in das Icinga-Web Frontend](#)
-

[Zurück](#)

[Weiter](#)

[Benachrichtigungen](#)

[Zum Anfang](#)

Icinga Classic UI: Informationen
über die CGIs

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Icinga Classic UI: Informationen über die CGIs

[Zurück](#)[Kapitel 6. Die Benutzeroberflächen](#)[Weiter](#)

Icinga Classic UI: Informationen über die CGIs

Einführung in das Icinga Classic UI

Die verschiedenen mit dem Icinga Core gelieferten CGIs werden hier beschrieben, zusammen mit den Autorisierungsanforderungen für den Zugriff und den Gebrauch jedes CGIs. Im Grundzustand erwarten die CGIs, dass Sie sich dem Web-Server gegenüber authentifiziert haben und autorisiert sind, jede Information zu sehen, die Sie anfordern. Mehr Informationen über die Konfiguration der Autorisierung finden Sie [hier](#).

Die CGIs können über verschiedene Parameter gesteuert werden. Mehr Informationen finden Sie [hier](#).

Index

[Status CGI](#)[Status map CGI](#)[WAP interface CGI](#)[Tactical overview CGI](#)[Network outages CGI](#)[Configuration CGI](#)[Command CGI](#)[Extended information CGI](#)[Event log CGI](#)[Alert history CGI](#)[Notifications CGI](#)[Trends CGI](#)[Availability reporting CGI](#)

[Alert histogram CGI](#)

[Alert summary CGI](#)

[Änderungen am Classic UI](#)

[Status CGI](#)



Dateiname: **status.cgi**

Beschreibung:

Dies ist das wichtigste mit Icinga gelieferte CGI. Es erlaubt Ihnen den aktuellen Status aller überwachten Hosts und Services zu sehen. Das Status-CGI kann zwei Arten von Ausgaben liefern - einen Status-Überblick aller Hostgruppen (oder einer bestimmten Hostgruppe) und eine detaillierte Anzeige aller Services (oder diese bezogen auf einen bestimmten Host).

Autorisierungsanforderungen:

- Wenn Sie *für alle Hosts autorisiert* sind, können Sie alle Hosts **und** alle Services ansehen.
- Wenn Sie *für alle Services autorisiert* sind, können Sie alle Services ansehen.
- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie alle Hosts und Services ansehen, deren Kontakt Sie sind.

[Status Map CGI](#)



Dateiname: **statusmap.cgi**

Beschreibung:

Dieses CGI erstellt eine Karte aller Hosts, die Sie in Ihrem Netzwerk definiert haben. Das CGI nutzt Thomas Boutells [gd](#)-Library (Version 1.6.3 oder höher), um ein PNG-Bild Ihrer Netzwerk-Struktur zu erstellen. Die verwendeten Koordinaten (zusammen mit den optionalen Icons) werden aus den [Host](#)-Definitionen genommen. Wenn Sie es vorziehen, dass das CGI automatisch für Sie Koordinaten generiert, nutzen Sie die [default_statusmap_layout](#)-Direktive, um einen Layout-Algorithmus zu definieren.

Autorisierungsanforderungen:

- Wenn Sie *für alle Hosts autorisiert* sind, können Sie alle Hosts **und** alle Services ansehen.
- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie alle Services ansehen, deren Kontakt Sie sind.



Anmerkung

Anmerkung: Benutzer, die nicht autorisiert sind, bestimmte Hosts zu sehen, werden *unbekannte Knoten* an diesen Stellen sehen. Uns ist klar, dass sie eigentlich *überhaupt nichts* dort sehen sollten, aber es ist nicht sinnvoll, eine Karte zu generieren, wenn man nicht die ganzen Host-Abhängigkeiten sehen kann.

WAP Interface CGI



Dateiname: **statuswml.cgi**

Beschreibung:

Dieses CGI dient als WAP-Interface für Netzwerk-Status-Informationen. Wenn Sie ein Gerät mit WAP-Unterstützung haben (z.B. ein Internet-fähiges Handy), können Sie Statusinformationen ansehen, während Sie unterwegs sind. Verschiedene Status-Anzeigen enthalten Hostgruppen-Zusammenfassung, Hostgruppen-Übersicht, Host-Details, Service-Details, alle Probleme und alle unbehandelten Probleme. Zusätzlich zur Ansicht von Statusinformationen können Sie mit Ihrem Handy auch Benachrichtigungen und Prüfungen deaktivieren und Probleme bestätigen. Ziemlich cool, oder?

Autorisierungsanforderungen:

- Wenn Sie *für Systeminformationen autorisiert* sind, können Sie Icinga-Prozess-Informationen ansehen.
- Wenn Sie *für alle Hosts autorisiert* sind, können Sie Zustandsdaten für alle Hosts **und** alle Services ansehen.
- Wenn Sie *für alle Services autorisiert* sind, können Sie Zustandsdaten für alle Services ansehen.
- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie Zustandsdaten für alle Hosts und Services ansehen, deren Kontakt Sie sind.

Tactical Overview CGI



Dateiname: **tac.cgi**

Beschreibung:

Dieses CGI dient als Sicht aus der "Vogelperspektive" auf alle Netzwerk-Überwachungs-Aktivitäten. Es erlaubt Ihnen schnell Netzwerkausfälle sowie Host- und Service-Zustände zu erkennen. Es unterscheidet zwischen Problemen, die auf irgendeine Weise "behandelt" wurden (z.B. bestätigt oder Benachrichtigungen deaktiviert) und solchen, die nicht behandelt wurden und die deshalb Beachtung erfordern. Das ist sehr hilfreich, wenn Sie viele zu überwachende Hosts und Services haben und einen einzelnen Bildschirm zur Alarmierung über Probleme einsetzen möchten.

Autorisierungsanforderungen:

- Wenn Sie *für alle Hosts autorisiert* sind, können Sie alle Hosts **und** alle Services ansehen.
- Wenn Sie *für alle Services autorisiert* sind, können Sie alle Services ansehen.
- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie alle Hosts und Services ansehen, deren Kontakt Sie sind.

Network Outages CGI



Dateiname: **outages.cgi**

Beschreibung:

Dieses CGI zeigt eine Liste von "Problem"-Hosts, die Netzwerkausfälle hervorrufen. Dies kann besonders dann hilfreich sein, wenn Sie ein großes Netzwerk haben und schnell die Quelle des Problems identifizieren möchten. Hosts werden sortiert nach der Schwere des Ausfalls, den sie bewirken.

- Wenn Sie *für alle Hosts autorisiert* sind, können Sie alle Hosts ansehen.
- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie alle Hosts ansehen, deren Kontakt Sie sind.

Configuration CGI



Dateiname: **config.cgi**

Beschreibung:

Dieses CGI erlaubt es Ihnen, Objekte (z.B. Hosts, Hostgruppen, Kontakte, Kontaktgruppen, Zeitfenster, Services, etc.) anzusehen, die Sie in Ihrer/Ihren [Objekt-Konfigurationsdatei\(en\)](#) definiert haben.

Autorisierungsanforderungen:

- Sie müssen *für Konfigurationsinformationen autorisiert* sein, um jegliche Konfigurationsinformationen ansehen zu können.

Command CGI



Dateiname: **cmd.cgi**

Beschreibung:

Dieses CGI erlaubt es Ihnen, Befehle an den Icinga-Prozess zu senden. Obwohl dieses CGI mehrere Argumente hat, sollten Sie besser darauf verzichten. Die meisten wechseln zwischen verschiedenen Revisionen von Icinga. Nutzen Sie das [extended information CGI](#) als Startpunkt, um Befehle zu erteilen.

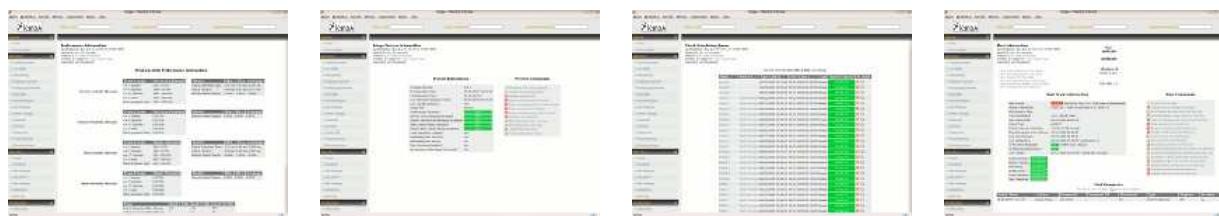
Autorisierungsanforderungen:

- Sie müssen *für System-Befehle autorisiert* sein, um Befehle zu erteilen, die den Icinga-Prozess beeinflussen (Start, Stop, Modus-Wechsel, etc.).
- Wenn Sie *für alle Hosts-Befehle autorisiert* sind, können Sie Befehle für alle Hosts **und** alle Services erteilen.
- Wenn Sie *für alle Service-Befehle autorisiert* sind, können Sie Befehle für alle Services erteilen.
- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie Befehle für alle Hosts und Services erteilen, deren Kontakt Sie sind.

Anmerkungen:

- Wenn Sie sich entschieden haben, die Option `use_authentication` für die CGIs nicht zu nutzen, wird dieses CGI *jedem* die Möglichkeit verweigern, Befehle zu erteilen. Dies geschieht zu Ihrem eigenen Schutz. Wir würden empfehlen, dieses CGI komplett zu entfernen, wenn Sie die Authentifizierung für CGI nicht nutzen.

Extended Information CGI



Dateiname: **extinfo.cgi**

Beschreibung:

Dieses CGI erlaubt es Ihnen, Icinga-Prozess-Informationen, Host- und Service-Zustandsstatistiken, Host- und Service-Kommentare und mehr anzusehen. Es dient auch als Startpunkt, um über das [command CGI](#) Befehle an Icinga zu erteilen. Obwohl dieses CGI mehrere Argumente hat, sollten Sie besser darauf verzichten. Die meisten wechseln zwischen verschiedenen Revisionen von Icinga. Sie können dieses CGI erreichen, indem Sie auf 'Network Health' bzw. 'Process Information' in der seitlichen Navigationsleiste klicken oder auf einen Host- oder Service-Link in der Ausgabe des [status CGI](#).

Autorisierungsanforderungen:

- Sie müssen *für Systeminformationen autorisiert* sein, um Icinga-Prozess-Informationen ansehen zu können.
- Wenn Sie *für alle Hosts autorisiert* sind, können Sie erweiterte Informationen für alle Hosts **und** alle Services ansehen.
- Wenn Sie *für alle Services autorisiert* sind, können Sie erweiterte Informationen für alle Services ansehen.
- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie erweiterte Informationen für alle Hosts und Services ansehen, deren Kontakt Sie sind.

Event Log CGI



Dateiname: **showlog.cgi**

Beschreibung:

Dieses CGI zeigt das [log file](#). Wenn Sie die [log rotation](#) aktiviert haben, können Sie in archivierten Log-Dateien blättern, indem Sie die Navigations-Links oben auf der Seite benutzen.

Autorisierungsanforderungen:

- Sie müssen [für Systeminformationen autorisiert](#) sein, um das Logfile ansehen zu können.

Alert History CGI



Dateiname: **history.cgi**

Beschreibung:

Dieses CGI wird benutzt, um die Problem-Historie für einen oder alle Hosts anzuzeigen. Die Ausgabe ist grundsätzlich ein Auszug der Informationen, die über das [log file CGI](#) angezeigt werden. Sie haben die Möglichkeit, die Ausgabe zu filtern, um nur die Problemtypen anzuzeigen, die Sie sehen wollen (z.B. Hard- und/oder Soft-Alarne, verschiedene Typen von Service- und Host-Alarmen, alle Arten von Alarmen, usw.). Wenn Sie die [log rotation](#) aktiviert haben, können Sie in archivierten Log-Dateien blättern, indem Sie die Navigations-Links oben auf der Seite benutzen.

Autorisierungsanforderungen:

- Wenn Sie [für alle Hosts autorisiert](#) sind, können Sie historische Informationen für alle Hosts **und** alle Services ansehen.
- Wenn Sie [für alle Services autorisiert](#) sind, können Sie historische Informationen für alle Services ansehen.

- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie historische Informationen für alle Hosts und Services ansehen, deren Kontakt Sie sind.

Notifications CGI



Dateiname: **notifications.cgi**

Beschreibung:

Dieses CGI wird genutzt, um Host- und Service-Benachrichtigungen anzuzeigen, die an verschiedene Kontakte versandt wurden. Die Ausgabe ist grundsätzlich ein Auszug der Informationen, die über das [log file CGI](#) angezeigt werden. Sie haben die Möglichkeit, die Ausgabe zu filtern, um nur die Benachrichtigungen anzuzeigen, die Sie sehen wollen (z.B. Service-Benachrichtigungen, Host-Benachrichtigungen, Benachrichtigungen, die an bestimmte Kontakte versandt wurden, usw.). Wenn Sie die [log rotation](#) aktiviert haben, können Sie in archivierten Log-Dateien blättern, indem Sie die Navigations-Links oben auf der Seite benutzen.

Autorisierungsanforderungen:

- Wenn Sie [für alle Hosts autorisiert](#) sind, können Sie Benachrichtigungen für alle Hosts **und** alle Services ansehen.
- Wenn Sie [für alle Services autorisiert](#) sind, können Sie Benachrichtigungen für alle Services ansehen.
- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie Benachrichtigungen für alle Hosts und Services ansehen, deren Kontakt Sie sind.

Trends CGI



Dateiname: **trends.cgi**

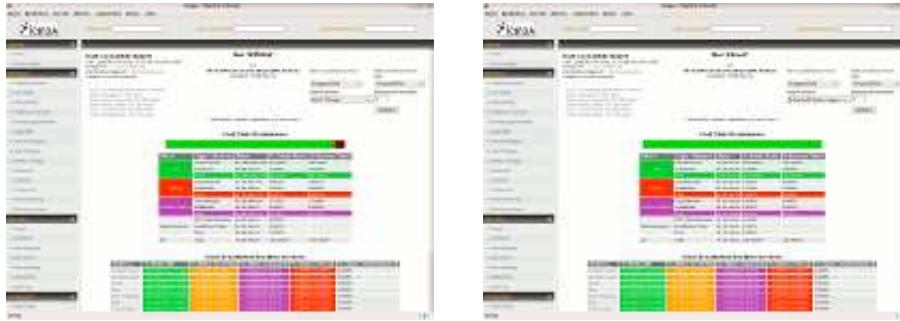
Beschreibung:

Dieses CGI wird genutzt, um einen Graphen über Host- oder Service-Zustände für einen beliebigen Zeitraum zu erstellen. Damit dieses CGI von Wert ist, sollten Sie [log rotation](#) aktivieren und archivierte Logs in dem Verzeichnis lagern, das durch die [log_archive_path](#)-Direktive angegeben wird. Das CGI nutzt Thomas Boutells [gd](#)-Library (Version 1.6.3 oder höher), um die Trend-Grafiken zu erstellen.

Autorisierungsanforderungen:

- Wenn Sie [für alle Hosts autorisiert](#) sind, können Sie Trends für alle Hosts **und** alle Services ansehen.
- Wenn Sie [für alle Services autorisiert](#) sind, können Sie Trends für alle Services ansehen.
- Wenn Sie ein [authentifizierter Kontakt](#) sind, können Sie Trends für alle Hosts und Services ansehen, deren Kontakt Sie sind.

Availability Reporting CGI



Dateiname: **avail.cgi**

Beschreibung:

Dieses CGI wird genutzt, um einen Bericht über die Verfügbarkeit von Hosts oder Service für einen benutzerdefinierten Zeitraum zu erstellen. Damit dieses CGI von Wert ist, sollten Sie [log rotation](#) aktivieren und archivierte Logs in dem Verzeichnis lagern, das durch die [log_archive_path](#)-Direktive angegeben wird.

Autorisierungsanforderungen:

- Wenn Sie [für alle Hosts autorisiert](#) sind, können Sie Verfügbarkeitsdaten für alle Hosts **und** alle Services ansehen.
- Wenn Sie [für alle Services autorisiert](#) sind, können Sie Verfügbarkeitsdaten für alle Services ansehen.
- Wenn Sie ein [authentifizierter Kontakt](#) sind, können Sie Verfügbarkeitsdaten für alle Hosts und Services ansehen, deren Kontakt Sie sind.

Alert Histogram CGI



Dateiname: **histogram.cgi**

Beschreibung:

Dieses CGI wird genutzt, um einen Bericht über die Verfügbarkeit von Hosts oder Service für einen benutzerdefinierten Zeitraum zu erstellen. Damit dieses CGI von Wert ist, sollten Sie [log rotation](#) aktivieren und archivierte Logs in dem Verzeichnis lagern, das durch die [log_archive_path](#)-Direktive angegeben wird. Das CGI nutzt Thomas Boutells [gd](#)-Library (Version 1.6.3 oder höher), um die Histogramm-Grafiken zu erstellen.

Autorisierungsanforderungen:

- Wenn Sie [für alle Hosts autorisiert](#) sind, können Sie Histogramme für alle Hosts **und** alle Services ansehen.
- Wenn Sie [für alle Services autorisiert](#) sind, können Sie Histogramme für alle Services ansehen.
- Wenn Sie ein [authentifizierter Kontakt](#) sind, können Sie Histogramme für alle Hosts und Services ansehen, deren Kontakt Sie sind.

Alert Summary CGI



Dateiname: **summary.cgi**

Beschreibung:

Dieses CGI stellt einige generische Berichte über Host- und Service-Alarmdaten zur Verfügung, darunter Gesamtzahl Alarme, Alarm-Spitzenreiter, etc.

Autorisierungsanforderungen:

- Wenn Sie [für alle Hosts autorisiert](#) sind, können Sie Summary-Informationen für alle Hosts **und** alle Services ansehen.

- Wenn Sie *für alle Services autorisiert* sind, können Sie Summary-Informationen für alle Services ansehen.
- Wenn Sie ein *authentifizierter Kontakt* sind, können Sie Summary-Informationen für alle Hosts und Services ansehen, deren Kontakt Sie sind.

Änderungen am Classic UI

Diese Änderungen sind im Laufe der Zeit eingeflossen, so dass sie ggf. nicht in Ihrer Version von Icinga verfügbar sind.

- Das Aussehen des "General"-Abschnitts auf der linken Seite hat sich erneut geändert

General CGI



File Name: **general.cgi**

Beim Klick auf die entsprechende Flagge können Sie weiterhin auf die Dokumentation in der dargestellten Sprache zugreifen.



Anmerkung

Es gibt keine Option, um die Sprache im Classic UI zu ändern. Dafür sind Anpassungen im Source-Code notwendig.

Nun können Sie wieder nach Hosts suchen, ohne weitere Dinge anzuklicken.

- Das klassische Interface wird in regelmäßigen Intervallen aktualisiert. Manchmal ist das nicht erwünscht, z.B. wenn Sie ein bestimmtes Objekt betrachten möchten. In diesem Fall können Sie die automatische Aktualisierung durch Klicken auf [pause] deaktivieren (direkt neben dem Text "Updated every 90 seconds" links oben im Status-Fenster). Klicken auf [continue] aktiviert die Aktualisierung wieder.

Pause CGI

Updated every 90 seconds [pause]

File Name: **pause.cgi**

Continue CGI

Update is paused [continue]

File Name: **continue.cgi**

- Die Seiten "Host Detail" und "Service Detail" wurden erweitert, so dass Sie nun Befehle für mehrere Objekte gleichzeitig erteilen können (ab Icinga 1.2). Nun können Sie ein oder mehrere Objekte durch Check-Boxen neben den Objekten auswählen. Durch aktivieren der Check-Box neben "Status information" werden alle Services eines Hosts ausgewählt.

Statusinfo CGI

Status Information	
OK - load average: 0.28, 0.55, 1.15	<input type="checkbox"/>
USERS OK - 3 users currently logged in	<input type="checkbox"/>
HTTP WARNING: HTTP/1.1 403 Forbidden	<input type="checkbox"/>
PING OK - Packet loss = 0%, RTA = 0.16 ms	<input type="checkbox"/>

File Name: **statusinfo.cgi**

Commands CGI

Commands for checked services	
Select command	<input type="button" value="Submit"/>

File Name: **commands.cgi**

Nach dem Klick auf "Select command" zeigt eine Drop-Down-Liste die verfügbaren Befehle. Nach der Auswahl einer Aktion und dem anschließenden Klick auf "Submit" wird der Befehl für die ausgewählten Objekte ausgeführt.

- "Export to CSV" wurde auf verschiedenen Seiten hinzugefügt (ab Icinga 1.2).
- Die Zellen der Tabelle in extinfo.cgi haben Namen bekommen. Mit Hilfe von SSI-Fragmenten können Sie JavaScript-Code einbinden, um auf die Daten dieser Zellen zuzugreifen (ab Icinga 1.2.1).

Der Beispiel-Code benutzt Daten der Zelle "comment_data", um einen Link zu erzeugen (Dank an Oliver Graf).

common-header.ssi:

```
<script type='text/javascript'>
function urlify() {
    var comments=document.getElementsByName('comment_data');
    var neu="";
    for (i=0; i<comments.length; i++) {
        comments[i].innerHTML = comments[i].innerHTML.replace(/\\bRT#(\d+)\\b/g, "<a href='https://YOUR-SERVER/Ticket/Display.html?id=$1'>RT#$1</a>");
    }
}
window.onload=urlify;
</script>
```

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Kapitel 6. Die
Benutzeroberflächen

[Zum Anfang](#)

Informationen zu den
CGI-Parametern

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Informationen zu den CGI-Parametern

[Zurück](#)

Kapitel 6. Die Benutzeroberflächen

[Weiter](#)

Informationen zu den CGI-Parametern

Einführung

Das Menü auf der linken Seite des klassischen Web-Interface enthält Einträge, die einen schnellen Zugriff auf die Informationen bieten, die die meisten Leute benötigen. Sie können diese Parameter ändern oder auch andere Optionen wählen. Einige CGIs benötigen einen Objekttyp ("host", "hostgroup", "service" oder "servicegroup"), oftmals gefolgt von einer oder mehreren Optionen. Der beste Weg ist es, einen Blick auf die existierenden URLs zu werben und sie Ihren Bedürfnissen anzupassen.

Nachfolgend finden Sie eine Tabelle mit den Parametern und den CGIs, die sie anbieten. Die Namen der CGIs sind abgekürzt (damit die Tabelle nicht so breit wird). Eine Erklärung der Abkürzungen zusammen mit einem Hinweis auf den Source-Code finden Sie [hier](#).

Nach dieser Tabelle folgt eine Erklärung der Parameter (in Bearbeitung).

Properties / Statustypes

Sie können `status.cgi` benutzen, um Objekte mit einem oder mehreren "Statustypen" (`hoststatustype` / `servicestatustype`) und mit bestimmten Eigenschaften (`hostprops` / `serviceprops`) anzuzeigen. Jeder Statustyp / jede Eigenschaft wird durch eine Zahl identifiziert (die [hier](#) und [hier](#) am Ende des Abschnitts zu finden sind). Bitte addieren Sie die entsprechenden Zahlen und setzen die resultierende Summe jeweils bei der Option ein.

Das Beispiel zeigt alle Services im nicht-OK-Zustand, die weder in einer Ausfallzeit (downtime) sind noch bestätigt (acknowledged) wurden:

```
http://localhost/icinga/cgi-bin/status.cgi?host=all&servicestatustypes=29&serviceprops=10
```

"29" enthält auch "PENDING" Services (die seit dem Start noch nicht geprüft wurden).



Anmerkung

Bitte beachten Sie, dass "properties" nur die Objekte anzeigt, die ALLE Bedingungen gleichzeitig erfüllen, während bei "statustypes" nur EINE Bedingung erfüllt sein muss (kann).

Matrix Parameter / CGIs

Parameter	avail	cmd	config	ext	hgram	hist	notif	out	log	status	map	sum	tac	trends
ahas		X												
alerttypes												X		
archive						X	X		X					
assumeinitialstates	X													X
assumestateretention	X				X									X
assumestatesduringnotrunning	X													X
backtrack	X				X									X
breakdown					X									
broadcast_notification		X												
childoptions		X												
cmd_mod		X												
cmd_typ		X												
columns										X				
com_author		X												
com_data		X												
com_id		X												
contact							X							
createimage					X					X				X
csvoutput	X	[2]	[2]	[2]	[2]	[2]	[2]	[2]	[2]	[2]		[2]	[2]	[2]
displaytype												X		
down_id		X												
eday	X				X							X		X
ehour	X				X							X		X
embedded	X			X	X	X	X	X	X	X	X	X	X	
emin	X				X							X		X
emon	X				X							X		X
end_time		X												
esec	X				X							X		X
eyear	X				X							X		X
fixed		X												
force_check		X												
force_notification		X												
full_log_entries	X													
get_date_parts	X													
graphevents					X									
graphstatetypes					X									
host	X	X		X	X	X	X			X		X		X
hostgroup	X	X		X						X		X		
hostprops										X				
hoststates												X		
hoststatustypes	X									X				

Parameter	avail	cmd	config	ext	hgram	hist	notif	out	log	status	map	sum	tac	trends
hours		X												
includesoftstates	X													
initialassumedhoststate	X													X
initialassumedservicestate	X													X
initialstateslogged					X									
input					X									X
jsonoutput [2][3]	X		X	X		X	X	X	X	X		X	X	
limit													X	
minutes		X												
navbarsearch										X				
newstatesonly					X									
nodowntime						X								
noflapping						X								
nofrills						X				X				
noheader	X			X	X	X	X	X	X	X	X	X	X	X
not_dly		X												
notimebreaks						X			X					
nosystem						X								
oldestfirst						X	X		X					
performance_data	X													
persistent	X													
plugin_output	X													
plugin_state	X													
ptc		X												
report												X		
report_type	X													
rpttimeperiod	X													
sched_dly		X												
sday	X				X							X		
send_notification		X												
service	X	X		X	X	X	X			X				X
servicefilter										X				
servicegroup	X	X		X						X		X		
serviceprops										X				
servicestates													X	
servicestatusatypes										X				
service_divisor								X						
shour	X				X							X		
showscheduleddowntime	X													
show_log_entries	X													
smin	X				X							X		

Parameter	avail	cmd	config	ext	hgram	hist	notif	out	log	status	map	sum	tac	trends
smon	X				X								X	
sortoption				X						X				
sorttype				X						X				
ssec	X				X								X	
standardreport													X	
start_time		X												
statetype						X								
statetypes												X		
sticky_ack		X												
style										X				
syear	X				X							X		X
t1	X				X							X		X
t2	X				X							X		X
timeperiod	X				X							X		X
trigger		X												
type			X	X		X	X							

Einzelheiten zu den Parametern

Mehr Informationen zu den einzelnen Parametern finden Sie nachfolgend. Für ein tiefergehendes Verständnis sollten Sie einen Blick auf den Source-Code werfen.

Parameter	Beschreibung	Mögl. Werte	Beispiel	Anmerkungen
ahas	Der Befehl beeinflusst den Host und seine Services		ahas	
alerttypes	Host- und/oder Service-Alarme anzeigen	1=Host-Alarne; 2=Service-Alarme; 3=Host- und Service-Alarne	alerttypes=3	
archive		0-n		
assumeinitialstates		yes; no		
assumestatesduringnotrunning		yes; no		
assumestateretention		yes; no		
backtrack	Wieviele Archiv-Log-Dateien sollen durchsucht werden, um den initialen Zustand zu ermitteln	0-n	backtrack=1	Bitte beachten Sie, dass die Verarbeitung der Archivdateien eine Weile dauern kann
breakdown	Aufteilen der Daten nach Zeitbereich	0=monatlich; 1=Tag des Monats; 2=Tag der Woche; 3=stündlich	breakdown=2	
broadcast_notification	Benachrichtigung an alle Kontakte versenden ("non-escalated" und "escalated")		broadcast_notification	

Parameter	Beschreibung	Mögl. Werte	Beispiel	Anmerkungen
<code>childoptions</code>	Ausfallzeitbehandlung für abhängige Hosts	0=ohne Berücksichtigung von abh. Hosts; 1="triggered downtime" für abh. Hosts planen; 2="non-triggered downtime" für abh. Hosts planen	<code>childoptions=1</code>	
<code>cmd_mod</code>	Command mode			
<code>cmd_typ</code>	Command type	0 - 169, 999	<code>cmd_typ=160</code>	Einzelheiten siehe .../include/common.h
<code>columns</code>	Anzahl von Übersichtsspalten	>= 1		Default ist 3
<code>com_author</code>	Autor des Kommentars	ein gültiger Benutzer	<code>com_author=icingaadmin</code>	Kann ggf. von der Einstellung von "lock_author_names" in cgi.cfg abhängen
<code>com_data</code>	Inhalt des Kommentars	eine Zeichenkette (urlencoded)		
<code>com_id</code>	Id des Kommentars			
<code>contact</code>	Ein gültiger Kontakt als Mail-Empfänger			
<code>createimage</code>			<code>createimage</code>	
<code>csvoutput</code>	Ob die Ausgabe im CSV-Format sein soll	yes;no		Diese Option setzt automatisch "noheader". In fast allen CGIs verfügbar (siehe [2])
<code>displaytype</code>	Typ der Alarmauswertung	1=recent alerts; 2=alert totals; 3=top alerts; 4=hostgroup alert totals; 5=host alert totals; 6=service alert totals; 7=servicegroup alert totals		
<code>down_id</code>	ID der Ausfallzeit			
<code>eday</code>	Ende des maßgeschneiderten Zeitfensters (Tag)			Nur gültig bei "maßgeschneiderten" Zeitfenstern
<code>ehour</code>	Ende des maßgeschneiderten Zeitfensters (Stunde)			Nur gültig bei "maßgeschneiderten" Zeitfenstern
<code>embedded</code>	verschiedenen HTML-Code und SSI-header/footer weglassen		<code>embedded</code>	
<code>emin</code>	Ende des maßgeschneiderten Zeitfensters (Minute)			Nur gültig bei "maßgeschneiderten" Zeitfenstern
<code>emon</code>	Ende des maßgeschneiderten Zeitfensters (Monat)			Nur gültig bei "maßgeschneiderten" Zeitfenstern
<code>end_time</code>	Endzeit der festen Ausfallzeit			Format "MM-DD-YYYY HH:MI"
<code>esec</code>	Ende des maßgeschneiderten Zeitfensters (Sekunde)			Nur gültig bei "maßgeschneiderten" Zeitfenstern

Parameter	Beschreibung	Mögl. Werte	Beispiel	Anmerkungen
<code>eyear</code>	Ende des maßgeschneiderten Zeitfensters (Jahr)			Nur gültig bei "maßgeschneiderten" Zeitfenstern
<code>fixed</code>	Feste oder flexible Ausfallzeit	0=flexible, >0=fixed		
<code>force_check</code>	Die Service-Prüfung wird erzwungen		forcecheck	
<code>force_notification</code>	Benachrichtigung ungeachtet von Restriktionen versenden (Zeitfenster oder andere)		force_notification	
<code>full_log_entries</code>	Vollständige oder gekürzte Log-Einträge anzeigen		full_log_entries	Default ist gekürzte Ansicht
<code>get_date_parts</code>	maßgeschneiderte Zeitbereiche ermitteln		get_date_parts	
<code>graphevents</code>	Welche Objekte in welchem Zustand dargestellt werden sollen		graphevents=112 (alle Service-Probleme)	Ein logisches ODER von: 1=Host up; 2=Host down; 4=Host unreachable; 8=Service OK; 16=Service Warning; 32=Service Unknown; 64=Service Critical
<code>graphstatetypes</code>	Hard- und/oder Soft-Zustände darstellen	1=Soft-states; 2=Hard states; 3=Hard- und Soft-states	graphstatetypes=3	
<code>host</code>	Alle Hosts oder einen bestimmten Host auswählen, dessen Service angezeigt werden sollen	all; <host name>	host=monitor	Spezielle Zeichen im Namen müssen kodiert werden ("urlencoded", z.B. "%20" statt eines Leerzeichens)
<code>hostgroup</code>	Alle Hostgruppen oder eine bestimmte Hostgruppe auswählen, deren Hosts und Services angezeigt werden sollen	all; <hostgroup name>	hostgroup=linux-boxes	Spezielle Zeichen im Namen müssen kodiert werden ("urlencoded", z.B. "%20" statt eines Leerzeichens)
<code>hostprops</code>	Alle Hosts auswählen, die dem angegebenen Bitmuster entsprechen. Bitte beachten Sie, dass die Hosts ALLE angegebenen Bedingungen entsprechen müssen		hostprops=131088 (aktive Prüfungen, die deaktiviert sind)	Ein logisches ODER der Bedingungen, die in include/cgiutils.c angegeben sind [1] (HOST AND SERVICE FILTER PROPERTIES)
<code>hoststates</code>	Der Zustand, in dem der Host sein sollte	1 - 7	hoststates=3 (Hosts in einem Problemzustand)	Ein logisches ODER der Zustände: 1=DOWN; 2=UNREACHABLE; 4=UP
<code>hoststatustypes</code>	Der Zustand, in dem der Host sein sollte	1 - 15	hoststatustypes=12 (Hosts in einem Problemzustand)	Ein logisches ODER der Zustände: 1=Pending; 2=Up; 4=Down; 8=Unreachable
<code>hours</code>	Dauer der flexiblen Ausfallzeit in Stunden (siehe "minutes")	>= 0		Nur gültig für flexible Ausfallzeiten
<code>includesoftstates</code>	"soft"-Zustände einschließen	yes; no	includesoftstate=yes	Default: "soft"-Zustände nicht berücksichtigen
<code>initialassumedhoststate</code>				

Parameter	Beschreibung	Mögl. Werte	Beispiel	Anmerkungen
<code>initialassumedservicestate</code>				
<code>initialstateslogged</code>				
<code>input</code>				
<code>jsonoutput</code>	Ob die Ausgabe im json-Format sein soll	yes;no		Diese Option setzt automatisch "noheader". In fast allen CGIs verfügbar (siehe [2])
<code>limit</code>	max. Anzahl von anzuzeigenden Einträgen	1-n	limit=10	Default ist 25
<code>minutes</code>	Dauer der flexiblen Ausfallzeit (siehe "hours")	>= 0		
<code>navbarsearch</code>				
<code>newstatesonly</code>	Nur "neue" Zustände anzeigen	yes; no	newstatesonly=yes	Default: alle Zustände anzeigen
<code>nodowntime</code>	Ausfallzeiten nicht anzeigen		nodowntime	
<code>noflapping</code>	"Flatter"-Alarne nicht anzeigen		noflapping	
<code>nofrills</code>	Don't display frills (?)		nofrills	
<code>noheader</code>	Globale Statusinformationen weglassen und nur Status-Details anzeigen		noheader	
<code>not_dly</code>	Benachrichtigung um n Minuten verzögern	>= 0		
<code>notimebreaks</code>	Don't display timebreaks (?)			
<code>nosystem</code>	Keine Systemmeldungen anzeigen		nosystem	Default: Systemmeldungen (des Icinga-Prozesses) anzeigen
<code>oldestfirst</code>	Sortierreihenfolge umdrehen		oldestfirst	Default: aktuelles Einträge zuerst anzeigen
<code>performance_data</code>	Die als Performance-Daten zu sendende Zeichenkette			
<code>persistent</code>	Der Kommentar ist persistent, wenn diese Option gesetzt ist		persistent	
<code>plugin_output</code>	Die als Plugin-Output zu sendende Zeichenkette			Die Länge ist begrenzt durch den Wert von MAX_INPUT_LENGTH (festgelegt während des Compile-Vorgangs)
<code>plugin_state</code>	Zustand des Plugins festlegen	0=OK; 1=Warning; 2=Critical; 3=Unknown	plugin_state=2	
<code>ptc</code>	Der Befehl wird an abhängige Hosts propagiert		ptc	
<code>report</code>	Report erzeugen		report	
<code>report_type</code>	Reporttyp auswählen	hostgroups; servicegroups; hosts; services	report_type=hostgroups	

Parameter	Beschreibung	Mögl. Werte	Beispiel	Anmerkungen
rpttimeperiod	Angaben eines Zeitfensters, die für den Availability-Bericht benutzt wird	Eins der definierten Zeitfenster		Benutzen Sie den Kurznamen der Zeitfenster-Definition
sched_dly	Befehlsausführung um n Minuten verzögern	>= 0		
sday	Start des maßgeschneiderten Zeitfensters (Tag)			
send_notification	Eine Benachrichtigung für die Bestätigung senden		send_notification	
service	Alle oder einen bestimmten Service auswählen, der angezeigt werden soll	all; <service description>	service=PING	Spezielle Zeichen im Namen müssen kodiert werden ("urlencoded", z.B. "%20" statt einen Leerzeichens)
servicefilter	Nur Service selektieren, deren Beschreibung auf das angegebene Muster passt		servicefilter=Current; servicefilter=[PL]	Das Muster ist abhängig von Groß-/Kleinschreibung. Reguläre Ausdrücke scheinen zu funktionieren
servicegroup	Alle oder eine bestimmte Servicegruppe auswählen, deren Hosts und Services angezeigt werden sollen	all; <servicegroup name>	servicegroup=disk	Spezielle Zeichen im Namen müssen kodiert werden ("urlencoded", z.B. "%20" statt einen Leerzeichens)
serviceprops	Alle Services auswählen, die dem angegebenen Bitmuster entsprechen. Bitte beachten Sie, dass die Services ALLE angegebenen Bedingungen entsprechen müssen		serviceprops=131088 (aktive Prüfungen, die deaktiviert sind)	Ein logisches ODER der Bedingungen, die in include/cgiutils.c angegeben sind [1] (HOST AND SERVICE FILTER PROPERTIES)
servicestates	Zustand, in dem die Services sein sollten		servicestates=56 (Services in einem Problemzustand)	Ein logisches ODER der Zustände: 8=Warning; 16=Unknown; 32=Critical; 64=OK
servicestatustypes	Zustand, in dem die Services sein sollten	1 - 31	servicestatustype=28 (services in problem state)	Ein logisches ODER der Zustände: 1=Pending; 2=OK, 4=Warning; 8=Unknown; 16=Critical
service_divisor	Wichtigkeit der Service in Relation zu Hosts	>=1		Services sind 1/n so wichtig wie Hosts. Default: n=4
shour	Start des maßgeschneiderten Zeitfensters (Stunde)			Nur gültig für flexible Ausfallzeiten
showscheduledowntime	Geplante Ausfallzeiten anzeigen	yes; no		Default: yes
show_log_entries	Log-Einträge anzeigen		show_log_entries	Default: keine Log-Einträge anzeigen
smin	Start des maßgeschneiderten Zeitfensters (Minute)			Nur gültig für flexible Ausfallzeiten

Parameter	Beschreibung	Mögl. Werte	Beispiel	Anmerkungen
smon	Start des maßgeschneiderten Zeitfensters (Monat)			Nur gültig für flexible Ausfallzeiten
sortoption	Angeben der Spalte, nach der sortiert werden soll	1-n	sortoption=3	Default ist Spalte 1
sorttype	Sortierrichtung für die Spalte, die über "sortoption=<n>" angegeben wurde	1=ascending; 2=descending	sorttype=2	
ssec	Start des maßgeschneiderten Zeitfensters (Sekunde)			Nur gültig für flexible Ausfallzeiten
standardreport	Standard-Report	1=recent alerts; 2=recent host alerts; 3=recent service alerts; 4=top host alert producers; 5=top service alert producers		
start_time	Start der festen Ausfallzeit			Format "MM-DD-YYYY HH:MI" (kann ggf. von Ihren Ländereinstellungen abhängen, das ist aber unklar)
statetype	Hard- und/oder Soft-states	0=Hard- und Soft-states; 1=Soft-states; 2=Hard-states	statetype=2	
statetypes	Hard- und/oder Soft-states	1=Soft-states; 2=Hard-states; 3=Hard- und Soft-states	statetypes=2	
sticky_ack	Die Bestätigung ist "sticky"		sticky_ack	
style	Angabe der anzuzeigenden Informationen	overview; detail; summary; grid; hostdetail		Trifft nur auf die Objekttypen "hostgroups" und "servicegroups" zu; hostdetail=host status details; detail=service status details; summary=status summary; grid=status grid;
syear	Start des maßgeschneiderten Zeitfensters (Jahr)			Nur gültig für flexible Ausfallzeiten
t1	Startpunkt eines maßgeschneiderten Zeitfensters		t1=1296109300	Unix timestamp
t2	Endpunkt eines maßgeschneiderten Zeitfensters		t2=1296189360	Unix timestamp
timeperiod	Zeitfenster, das für den "Availability report" benutzt werden soll	today; yesterday; thisweek; lastweek; thismonth; lastmonth; thisquarter; lastquarter; thisyear; lastyear; last24hours; last7days; last31days; custom	timeperiod=lastmonth	

Parameter	Beschreibung	Mögl. Werte	Beispiel	Anmerkungen
trigger	Die Ausfallzeit wird von der Downtime-ID <n> ausgelöst	Eine gültige Downtime-ID		
type	Objekttyp	hosts; hostgroups; services; servicegroups; contacts; contactgroups; timeperiods; commands; hostescalations; serviceescalations; hostdependencies; servicedependencies	type=hosts	

Abkürzungen, CGIs, Verweise

Abkürzungen, die in der ersten Tabelle benutzt werden, Beziehungen zu CGIs- und Menüeinträgen sowie Verweise auf den Source-Code in <icinga-core>/cgis.

Abkürzung	CGI	Menueeintrag	Source-Code
avail	avail.cgi	Availability	avail.c
cmd	cmd.cgi	N/A	cmd.c
config	config.cgi	Configuration	config.c
ext	extinfo.cgi	Comments, Downtime, Process Info, Performance Info, Schedulung Info	extinfo.c
hgram	histogram.cgi	Alert Histogram	histogram.c
hist	history.cgi	Alert History	history.c
notif	notifications.cgi	Notifications	notifications.c
out	outages.cgi	Network Outages	outages.c
log	showlog.cgi	Event Log	showlog.c
status	status.cgi	Hostgroup Overview, Servicegroup Overview, Host Problems, Service Problems	status.c
map	statusmap.cgi	Status Map	statusmap.c
summary	summary.cgi	Alert Summary	summary.c
tac	tac.cgi	N/A	tac.c
trends	trends.cgi	Trends	trends.c

Ausschnitt aus include/cgiutils.h

```
***** HOST AND SERVICE FILTER PROPERTIES *****

#define HOST_SCHEDULED_DOWNTIME      1
#define HOST_NO_SCHEDULED_DOWNTIME    2
#define HOST_STATE_ACKNOWLEDGED     4
#define HOST_STATE_UNACKNOWLEDGED   8
#define HOST_CHECKS_DISABLED        16
#define HOST_CHECKS_ENABLED         32
#define HOST_EVENT_HANDLER_DISABLED 64
```

```
#define HOST_EVENT_HANDLER_ENABLED      128
#define HOST_FLAP_DETECTION_DISABLED   256
#define HOST_FLAP_DETECTION_ENABLED    512
#define HOST_IS_FLAPPING              1024
#define HOST_IS_NOT_FLAPPING           2048
#define HOST_NOTIFICATIONS_DISABLED    4096
#define HOST_NOTIFICATIONS_ENABLED     8192
#define HOST_PASSIVE_CHECKS_DISABLED   16384
#define HOST_PASSIVE_CHECKS_ENABLED    32768
#define HOST_PASSIVE_CHECK             65536
#define HOST_ACTIVE_CHECK              131072
#define HOST_HARD_STATE                262144
#define HOST_SOFT_STATE                524288

#define SERVICE_SCHEDULED_DOWNTIME      1
#define SERVICE_NO_SCHEDULED_DOWNTIME   2
#define SERVICE_STATE_ACKNOWLEDGED    4
#define SERVICE_STATE_UNACKNOWLEDGED  8
#define SERVICE_CHECKS_DISABLED        16
#define SERVICE_CHECKS_ENABLED         32
#define SERVICE_EVENT_HANDLER_DISABLED 64
#define SERVICE_EVENT_HANDLER_ENABLED  128
#define SERVICE_FLAP_DETECTION_ENABLED 256
#define SERVICE_FLAP_DETECTION_DISABLED 512
#define SERVICE_IS_FLAPPING            1024
#define SERVICE_IS_NOT_FLAPPING        2048
#define SERVICE_NOTIFICATIONS_DISABLED 4096
#define SERVICE_NOTIFICATIONS_ENABLED  8192
#define SERVICE_PASSIVE_CHECKS_DISABLED 16384
#define SERVICE_PASSIVE_CHECKS_ENABLED  32768
#define SERVICE_PASSIVE_CHECK          65536
#define SERVICE_ACTIVE_CHECK            131072
#define SERVICE_HARD_STATE              262144
#define SERVICE_SOFT_STATE              524288
```

Excerpt from include/statusdata.h

```
***** SERVICE STATES *****/
#define SERVICE_PENDING  1
#define SERVICE_OK      2
#define SERVICE_WARNING 4
#define SERVICE_UNKNOWN 8
#define SERVICE_CRITICAL 16

***** HOST STATES *****/
#define HOST_PENDING     1
#define HOST_UP          2
#define HOST_DOWN        4
#define HOST_UNREACHABLE 8
```

[1] Logisches ODER bedeutet, dass die jeweiligen Zahlen addiert werden. Es werden dann die Objekte angezeigt, die ALLE Bedingungen erfüllen.

[2] Verfügbar ab Icinga 1.4.

[3] avail, log,notif,out,status,sum: Alle Views/Reports unterstützen jsonoutput; config: Alle Typen außer command expansion; ext: Alle Views außer hostgroup/servicegroup info (immer außer Performance-Daten); tac: Datenausgabe im json-Format. Mehr Informationen finden Sie im [Icinga-Wiki](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Icinga Classic UI: Informationen
über die CGIs

[Zum Anfang](#)

Ausführen von CGIs auf der
Kommandzeile

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Ausführen von CGIs auf der Kommandozeile

[Zurück](#)

Kapitel 6. Die Benutzeroberflächen

[Weiter](#)

Ausführen von CGIs auf der Kommandozeile

Einführung

In den meisten Fällen werden Sie Ihren Browser benutzen, um die Informationen anzusehen, die Sie benötigen. Es mag Situationen geben, in denen Sie die Daten mit anderen Werkzeugen bearbeiten möchten, um Wiki-Einträge zu erzeugen, Mails zu verschicken usw. Zusammen mit den [Informationen zu den CGI-Parametern](#) können Sie die CGIs auf der Kommandozeile aufrufen.

Voraussetzungen

Bevor Sie die CGIs tatsächlich aufrufen können, müssen Sie drei Umgebungsvariablen setzen:

- `REMOTE_USER`

Diese Variable enthält einen Benutzer, der berechtigt ist, die Informationen abzurufen. In den meisten Fällen wird dies "icingaadmin" sein (`set REMOTE_USER='icingaadmin'`)

- `REQUEST_METHOD`

`set REQUEST_METHOD='GET'`. Mögliche Werte sind "GET", "POST" und "HEAD"

- `QUERY_STRING`

Anstatt Argumente über die Kommandozeile an die CGIs zu übergeben, müssen Sie die Variable "QUERY_STRING" mit den entsprechenden Werten füllen.



Anmerkung

Die meisten Leute finden es schwierig, HTML-Ausgaben zu lesen, so dass es eine gute Idee ist, der Variable `QUERY_STRING` "jsonoutput" oder "csvoutput" hinzuzufügen (`QUERY_STRING='jsonoutput'` bzw. `QUERY_STRING='csvoutput'`).

Wenn Sie vergessen, die Umgebungsvariablen zu setzen, dann bekommen Sie beim Aufruf die folgenden Zeilen:

```
$> ./status.cgi
getcgivars(): Unsupported REQUEST_METHOD -> ''

I'm guessing you're trying to execute the CGI from a command line.
In order to do that, you need to set the REQUEST_METHOD environment
variable to either "GET", "HEAD", or "POST". When using the
GET and HEAD methods, arguments can be passed to the CGI
by setting the "QUERY_STRING" environment variable. If you're
using the POST method, data is read from standard input. Also of
note: if you've enabled authentication in the CGIs, you must set the
"REMOTE_USER" environment variable to be the name of the user you're
"authenticated" as.
```

Beispiele



Anmerkung

Die CGIs werden aus dem Ordner aufgerufen, in dem sich die *.cgi-Dateien befinden (z.B. /usr/local/icinga/sbin). Dies ist nicht notwendig, sondern dient lediglich der Einfachheit. Solange nichts anderes angegeben ist, gilt REQUEST_METHOD= 'GET'.

Tactical overview

```
$> set QUERY_STRING='jsonoutput'
$> ./tac.cgi
```

Alle Hosts im Zustand DOWN

```
$> set QUERY_STRING='jsonoutput&style=hostdetail&hoststatustypes=4'
$> ./status.cgi
```

Alle Hosts im Zustand DOWN, die "unacknowledged" und nicht in einer Downtime sind

```
$> set QUERY_STRING='jsonoutput&style=hostdetail&hoststatustypes=4&hostprops=10'
$> ./status.cgi
```

Alle Services in einem nicht-OK-Zustand

```
$> set QUERY_STRING='jsonoutput&style=detail&servicestatustypes=28'
$> ./status.cgi
```

Alle passiven Services im Zustand CRITICAL

```
$> set QUERY_STRING='jsonoutput&style=detail&servicestatustypes=28&serviceprops=65536'
$> ./status.cgi
```

Kommentare für alle Objekte

```
$> set QUERY_STRING='jsonoutput&type=3'
$> ./extinfo.cgi
```

Trends für router_02, Zeitangaben durch Unix-Timestamps

```
$> set QUERY_STRING='jsonoutput&host=router_02&timeperiod=custom&t1=130748400&t2=1307570400'
$> ./extinfo.cgi
```

Trends für router_02, Zeitangaben durch Datum und Uhrzeit

```
$> set QUERY_STRING='jsonoutput&host=router_02&timeperiod=custom\
&sday=6&smon=6&syear=2011&shour=0&smin=0&ssec=0\
&eday=7&emon=6&eyear=2011&ehour=0&emin=0&esec=0'
$> ./extinfo.cgi
```

(wird fortgesetzt)

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Informationen zu den
CGI-Parametern

[Zum Anfang](#)

Installation des Icinga-Web
Frontend

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Installation des Icinga-Web Frontend

[Zurück](#)
[Kapitel 6. Die Benutzeroberflächen](#)
[Weiter](#)

Installation des Icinga-Web Frontend

Einleitung

Das neue Icinga-Web wird aktuell sehr stark weiterentwickelt, sodass sich diese Dokumentation unter Umständen noch ändern kann/wird. Falls Sie eine detaillierte Installationsanleitung benötigen, konsultieren Sie bitte doc/INSTALL aus dem Source-Paket. Mehr Informationen über die gesamte Architektur können Sie auf unserer Webseite nachlesen: <http://www.icinga.org/architecture/> Falls Sie mehr über das Icinga-Web-Development und die Modulararchitektur wissen wollen, verfolgen Sie bitte stetig aktualisierte Einträge im Development-Wiki von Icinga-Web: [Icinga-Wiki](#)

Diese Installationsanleitung beschreibt die Installation von Icinga-Web mit MySQL bzw. PostgreSQL als verwendeter Datenbank, außerdem wird Oracle unterstützt.

1. Voraussetzungen

- **MySQL**

Die Pakete mysql und php5 sind installiert (mit PEAR und CLI), Icinga samt IDOUtils läuft, dann ab zu 2., ansonsten bitte:

- **Ubuntu / Debian**

```
#> apt-get install php5 php5-cli php-pear php5-xmlrpc php5-xsl php5-pdo php5-ldap php5-soap php5-gd php5-mysql
```



Anmerkung

Für Ubuntu gibt es kein php5-pdo-Paket, die benötigten PDO-Extensions sind in den Paketen php5 und php5-mysql enthalten!

- **Fedor a / RHEL / CentOS**

Stellen Sie sicher, dass Sie ein Repository bzw. Pakete für PHP 5.2.3 und PCRE 7.6 haben - RHEL/CentOS <= 5.x enthalten bisher lediglich PHP 5.1.6 und PCRE 6.6.

```
#> yum install php php-cli php-pear php-xmlrpc php-xsl php-pdo php-ldap php-soap php-gd php-mysql
```



Anmerkung

Aktuelle Pakete für PHP und PCRE finden Sie beispielsweise unter: [Les RPM de Remi](#) oder
<http://www.jasonlitka.com/category/yum-repo-news/>

- **OpenSuSE**

Benutzen Sie yast, um die Pakete "php5", "php5-pear", "php5-xmlrpc", "php5-xsl", "php5-mysql", "php5-soap", "php5-gettext", "php5-pdo", "php5-ldap", "php5-gd" und "apache2-mod_php5" zu installieren. Die CLI ist im php5-Paket enthalten.



Anmerkung

Zumindest bei SLES10 SP2 fehlt die Funktion hash_hmac.

Alternativ können Sie auch zypper verwenden

```
#> zypper install php5 php5-pear php5-xmlrpc php5-xsl php5-mysql php5-soap \
    php5-gettext php5-pdo php5-ldap php5-gd apache2-mod_php5
```

- **PostgreSQL**

Die Pakete pgsql und php5 sind installiert (mit PEAR und CLI), Icinga samt IDOUtils läuft, dann ab zu 2., ansonsten bitte:

- **Ubuntu / Debian**

```
#> apt-get install php5 php5-cli php-pear php5-xmlrpc php5-xsl php5-pdo php5-ldap php5-gd php5-pgsql
```



Anmerkung

Für Ubuntu gibt es kein php5-pdo-Paket, die benötigten PDO-Extensions sind in den Paketen php5 und php5-pgsql enthalten!

- **Fedora / RHEL / CentOS**

Stellen Sie sicher, dass Sie ein Repository bzw. Pakete für PHP 5.2.3 und PCRE 7.6 haben - RHEL/CentOS <= 5.x enthalten bisher lediglich PHP 5.1.6 und PCRE 6.6.

```
#> yum install php php-cli php-pear php-xmlrpc php-xsl php-pdo php-ldap php-gd php-pgsql
```



Anmerkung

Aktuelle Pakete für PHP und PCRE finden Sie beispielsweise unter: [Les RPM de Remi](#) oder
<http://www.jasonlitka.com/category/yum-repo-news/>

- **OpenSuSE**

Benutzen Sie yast, um die Pakete "php5", "php5-pear", "php5-xmlrpc", "php5-xsl", "php5-pgsql", "php5-soap", "php5-gettext", "php5-pdo", "php5-ldap", "php5-gd" und "apache2-mod_php5" zu installieren. Die CLI ist im php5-Paket enthalten.



Anmerkung

Zumindest bei SLES10 SP2 fehlt die Funktion hash_hmac.

Alternativ können Sie auch zypper verwenden

```
#> zypper install php5 php5-pear php5-xmlrpc php5-xsl php5-mysql php5-soap \
    php5-gettext php5-pdo php5-ldap php5-gd apache2-mod_php5
```

Die Installation von Icinga mit den IDOUtils ist in der [Icinga-Schnellstartanleitung](#) beschrieben

2. Die Installation

Laden Sie das Archiv herunter von <http://sourceforge.net/projects/icinga/files/>. Falls Sie eine Funktion benötigen, die nur in der aktuellen Entwicklerversion vorhanden ist, dann klonen Sie von icinga-web.git:

```
#> git clone git://git.icinga.org/icinga-web.git
```

Extrahieren Sie das Archiv (tarball) wie folgt:

```
#> tar xzvf icinga-web-1.6.0.tar.gz
```

Wechseln Sie in das Verzeichnis:

```
#> cd icinga-web-1.6.0
```

Icinga-Web stellt verschiedene configure Optionen zur Verfügung z.B.

```
#> ./configure
      --prefix=/usr/local/icinga-web
      --with-web-user=www-data
      --with-web-group=www-data
      --with-web-path=/icinga-web
      --with-web-apache-path=/etc/apache2/conf.d
      --with-db-type=mysql
      --with-db-host=localhost
      --with-db-port=3306
      --with-db-name=icinga_web
      --with-db-user=icinga_web
      --with-db-pass=icinga_web
      --with-conf-folder=etc/conf.d
      --with-log-folder=log
      --with-api-subtype=TYPE DB driver or network connection
      --with-api-host=HOST Host to connect (DB or other) (default localhost)
      --with-api-port=PORT Port for connection (default 3306)
      --with-api-socket=PATH Path to socket (default none)
```



Anmerkung

Bitte beachten Sie, dass Sie hier die Icinga-Web-Datenbank konfigurieren, und nicht die Icinga-IDOUtils-Datenbank! User- und Gruppenname des Web-Prozesses sind abhängig von der verwendeten Distribution.

Alle Konfigurationsmöglichkeiten sehen Sie mit:

```
#> ./configure --help
```

Um Icinga-Web für PostgreSQL zu konfigurieren, führen Sie folgenden Befehl aus:

```
#> ./configure --with-db-type=pgsql --with-db-port=5432
```

Falls auch Icinga mit pqsql läuft, sind die api-Optionen mit anzugeben:

```
#> ./configure --with-db-type=pgsql --with-db-port=5432 \
--with-api-subtype=pgsql --with-api-port=5432
```

Ohne weitere Optionen wird das Icinga-Webinterface mit:

```
#> ./configure
#> make install
```

unter /usr/local/icinga-web installiert. Das kann eine Weile dauern, also bitte nicht ungeduldig werden ;-)

Installation der neuen Apache-Konfiguration

```
#> make install-apache-config
```

Wenn Sie das nicht möchten, können Sie das bisherige Verfahren nutzen, durch die Eingabe von:

```
#> make install-javascript
```

Dies installiert die bisherigen Symlinks.

Erzeugen des Installationsberichtes:

```
#> make install-done
```

```
Installation of icinga-web succeeded.
Please check the new Apache2 configuration (etc/apache2/icinga-web.conf).
```

Passwort zurücksetzen:

```
#> make icinga-reset-password
```

Setzt das Passwort für jeden Account auf icinga-web zurück.

3. PHP-Abhängigkeiten

Bitte prüfen Sie die PHP-Abhängigkeiten mit:

```
#> make testdeps
```

Alle "require"-Tests sollten erfolgreich sein. Eventuell müssen Sie die php.ini anpassen:

Die "magic_quote_gpc" in der Apache- und der CLI-php.ini auf "Off" setzen! Evtl. müssen Sie auch "safe_mode" auf "off" setzen (PHP < 5.3.0). Bitte beachten Sie, dass die Pfade von der Distribution abhängig sind.

```
#> vi /etc/php5/apache/php.ini
magic_quotes_gpc = off
safe_mode = off
#> vi /etc/php5/cli/php.ini
magic_quotes_gpc = off
```



Anmerkung

Beide Dateien müssen vorhanden sein, sonst erhalten Sie eine Agavi-Meldung, weil der Default von magic_quotes_gpc "ON" ist.

4. Datenbank-Installation

Icinga-Web benötigt eine eigene Datenbank, z.B. icinga_web. Sie können die Datenbank von IDOUtils mitverwenden, dies wird allerdings nicht empfohlen, um bei Upgrades keine Probleme zu haben.

Anlegen des Datenbankbenutzers:

Der Benutzer muss mindestens diese Privilegien haben: SELECT, UPDATE, INSERT, DELETE.

```
SQL> GRANT USAGE ON *.* TO 'icinga_web'@'localhost' IDENTIFIED BY 'icinga_web';

SQL> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, INDEX
    ON icinga_web.* TO 'icinga_web'@'localhost';
```

- MySQL

```
#> mysql -u root -p

mysql> GRANT USAGE ON icinga_web.* TO 'icinga_web'@'localhost'
        IDENTIFIED BY 'icinga_web'
        WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0;

        GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER, INDEX
        ON icinga_web.* TO 'icinga_web'@'localhost';

mysql> quit
```

- PostgreSQL

```
#> su - postgres

$ psql

postgres=# CREATE USER icinga_web;

postgres=# ALTER USER icinga_web WITH PASSWORD 'icinga_web' CREATEDB;
<Ctrl>+<D>
```

Anlegen der Datenbank

Icinga-Web bringt Doctrine mit, womit in Zukunft die Datenbank verwaltet wird. Mittels 'make' kann diese initialisiert oder gelöscht werden.

```
#> make db-initialize      - legt eine Icinga Web Datenbank an
                                - und füllt diese mit initialen Werten

#> make db-drop            - löscht die komplette Datenbank, inklusive
                                - Security-Abfrage, um ungewolltes Löschen zu verhindern

#> make db-doc2sql         - falls Sie SQL scripts beispielsweise für Packaging oder
                                - manuelles Setup benötigen, können Sie diese Option verwenden,
                                - verwenden, um die SQL-Skripts aus Doctrine zu extrahieren
```

Um die Datenbankanlege-Befehle nutzen zu können, müssen Sie dem Benutzer, der diese DB-Befehle ausführen soll, auch die entsprechenden Privilegien zuordnen. Falls derjenige Benutzer, der für das Icinga-Web in config.php vorgesehen wurde, zu wenig Berechtigungen hat, gibt es in 'make' eine Abfrage, ob Sie einen anderen Benutzer mit mehr Privilegien verwenden wollen. Falls dies nicht funktioniert, sollten Sie etc/build.properties editieren und einen root-Benutzer einfügen.

Das Installieren der Datenbank funktioniert mittels:

```
#> make db-initialize
```

Manuelles Anlegen der Datenbank

Sofern Sie die Datenbank manuell installieren wollen, z.B. für Package building, können Sie mit diesem Befehl die SQL-Scripts extrahieren

```
#> make db-doc2sql
```

und diese dann in die frisch angelegte Datenbank importieren.

PostgreSQL

Bitte editieren Sie die Datei `pg_hba.conf` und tragen Sie die folgenden Zeilen ein:

```
#> vi /etc/postgresql/9.x/main/pg_hba.conf

# database administrative login by UNIX sockets
local all postgres ident

# TYPE DATABASE USER CIDR-ADDRESS METHOD
#icinga_web
local icinga_web icinga_web trust

# "local" is for Unix domain socket connections only
local all all trust

# IPv4 local connections
host all all 127.0.0.1/32 trust

# IPV6 local connections
host all all ::1/128 trust
```

Denken Sie anschließend daran, PostgreSQL neu zu laden:

```
#> /etc/init.d/postgresql reload
```

5. Icinga-Web-Konfiguration

Im Normalfall können Sie die Datenbankeinstellungen während `configure` vornehmen. Sollten Sie diese allerdings anpassen oder im Fehlerfall kontrollieren wollen, öffnen Sie folgende Datei mit einem Editor Ihrer Wahl.

```
#> vi app/config/databases.xml
```



Anmerkung

Optional: Ihre spezifischen Icinga Datenbankeinstellungen können Sie vornehmen in `etc/conf.d/database.xml` (oder dem `web-conf`-folder Ziel). Es gibt zwei Datenbanken: `icinga_web` (welche für die interne Verwaltung von `icinga_web` verwendet wird) und `icinga`, welche auf die Datenbank zeigt, in die `ido2db` schreibt. Diese Informationen bleiben während eines Update-Prozesses erhalten.

```
<databases default="icinga_web">
  <database name="icinga_web" class="AgaviDoctrineDatabase">
    <!--
      Doctrine dsn strings:
      http://www.doctrine-project.org/documentation/manual/1_1/en/introduction-to-connections
    -->
    <!-- ##### MySQL ##### -->
```

```

<ae:parameter name="dsn">mysql://icinga_web:icinga_web@127.0.0.1:3306/icinga_web</ae:parameter>
<!-- ##### PostgreSQL ##### -->
<ae:parameter name="dsn">pgsql://icinga_web:icinga_web@localhost:5432/icinga_web</ae:parameter>

<!-- Generic credentials -->
<!-- <ae:parameter name="username">icinga_web</ae:parameter> -->
<!-- <ae:parameter name="password">icinga_web</ae:parameter> -->

<!-- DB encoding type -->
<ae:parameter name="charset">utf8</ae:parameter>

<!--
      Doctrine_Manager configuration
-->
<ae:parameter name="manager_attributes">
    <!-- This allows lazy loading of the models -->
    <ae:parameter name="Doctrine_Core::ATTR_MODEL_LOADING">CONSERVATIVE</ae:parameter>
</ae:parameter>

<!-- The path to our models -->
<ae:parameter name="load_models">%core.module_dir%/AppKit/lib/database/models/generated</ae:parameter>
<ae:parameter name="models_directory">%core.module_dir%/AppKit/lib/database/models</ae:parameter>

</database>

<!-- Die Icinga Datenbank für ido2db -->
<database name="icinga" class="AgaviDoctrineDatabase">

    <!--
          Doctrine dsn strings:
          http://www.doctrine-project.org/documentation/manual/1_1/en/introduction-to-connections
-->
    <!-- #### MySQL ##### -->
    <ae:parameter name="dsn">mysql://icinga:icinga@127.0.0.1:3306/icinga</ae:parameter>
    <!-- #### PostgreSQL ##### -->
    <ae:parameter name="dsn">pgsql://icinga:icinga@localhost:5432/icinga</ae:parameter>

    <!-- Generic credentials -->
    <!-- <ae:parameter name="username">icinga</ae:parameter> -->
    <!-- <ae:parameter name="password">icinga</ae:parameter> -->

    <!-- DB encoding type -->
    <ae:parameter name="charset">utf8</ae:parameter>

    <!--
          Doctrine_Manager configuration
-->
    <ae:parameter name="manager_attributes">
        <!-- This allows lazy loading of the models -->
        <ae:parameter name="Doctrine_Core::ATTR_MODEL_LOADING">CONSERVATIVE</ae:parameter>
    </ae:parameter>

    <!-- The path to our models -->
    <ae:parameter name="load_models">%core.module_dir%/Api/lib/database/models/generated</ae:parameter>
    <ae:parameter name="models_directory">%core.module_dir%/Api/lib/database/models</ae:parameter>

</database>
</databases>

```

In der access.xml können Sie die Informationen zu Ihrer command-pipe, den Zugangsdaten und der Zugriffsmethode (ssh oder via local console) angeben. Außerdem legen Sie fest, welche Icinga-Instanz welchem Host zugeordnet ist.

```

<!-- Map your instances to hosts here -->
<instances>
    <instance name="default">localhost</instance>
</instances>
<!-- default location of the icinga pipe -->
<defaults>
    <access>
        <write>
            <files>
                <resource name="icinga_pipe">/usr/local/icinga/var/rw/icinga.cmd</resource>
            </files>
        </write>
    </defaults>

    <!-- Hosts that can be accessed via the console interface -->
    <hosts>
        <host name="localhost">
            <type>local</type>
            <access useDefaults="true">

                </access>
        </host>

        <!-- Example for ssh connection with user/password auth -->
        <host name="remot_host">
            <type>ssh</type>
            <ssh-config>
                <host>localhost</host>
                <port>22</port>
                <auth>

```

```

<type>password</type>
<user>john</user>
<password>doe</password>
</auth>
</ssh-config>

<access useDefaults="true" />

</host>
</hosts>
```



Anmerkung

Nachdem Sie Änderungen an diesen Konfigurationen vorgenommen haben, müssen Sie den Cache leeren!

```
#> rm -rf app/cache/config/*.php
oder /path/to/clearcache.sh
#> /usr/local/icinga-web/bin/clearcache.sh
```

6. Apache-Konfiguration

Voraussetzungen:

- mod_rewrite, vielleicht müssen Sie einen Verweis erstellen:

```
#> ln -s /etc/apache2/mods-available/rewrite.load /etc/apache2/mods-enabled/rewrite.load
```

Bei OpenSuSE und SLES können Sie das Modul mit "a2enmod rewrite" aktivieren. Falls das nicht funktioniert, gibt es in der Datei /etc/sysconfig/apache2 die Zeile "APACHE_MODULES=...", der das Modul "rewrite" hinzugefügt werden muss.

Bei Debian und Ubuntu können Sie das Modul ebenfalls mit "a2enmod rewrite" aktivieren.

Bei RHEL/Fedora/CentOS ist die Unterstützung bereits im httpd enthalten.

- Über htaccess aktivierte Einstellungen

Editieren Sie die .htaccess unter /usr/local/icinga-web/pub und ändern Sie die RewriteBase (ab Zeile 14):

```
DirectoryIndex index.php

Options -MultiViews -Indexes +FollowSymLinks
Order allow,deny
Allow from all

<IfModule mod_rewrite.c>
    RewriteEngine On

    # This depends on your path
    # on independent hosts the base is ''
    RewriteBase /icinga-web/

    # If the requested URL does not exist (it's likely an agavi route),
    # pass it as path info to index.php, the Agavi dispatch script.
    RewriteRule ^$ index.php?/ [QSA,L]
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule (.*) index.php?/$1 [QSA,L]
</IfModule>
```

```

<IfModule mod_deflate.c>
    SetOutputFilter DEFLATE

    BrowserMatch ^Mozilla/4 gzip-only-text/html
    BrowserMatch ^Mozilla/4\.[678] no-gzip

    BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
    BrowserMatch \bMSI[E] !no-gzip !gzip-only-text/html

    Header append Vary User-Agent env=!dont-vary
</IfModule>

<IfDefine APACHE2>
    AcceptPathInf On
</IfDefine>

#<IfModule mod_auth_basic.c>
#    AuthType Basic
#    AuthName "My http basic auth realm"
#    AuthUserFile /path/to/my/htusers
#    require valid-user
#</IfModule>

```

Bitte wechseln Sie in das Konfigurationsverzeichnis Ihres Webservers. Überprüfen Sie, ob die mit **make install-apache-config** erstellte Konfiguration Ihren Anforderungen entspricht, oder erstellen Sie noch einen neuen Alias im Konfigurationsverzeichnis des Webservers (hier in der icinga-web.conf):

```

#> vi /etc/apache2/conf.d/icinga-web.conf
#
# icinga-web apache configuration
# - Enable all options .htaccess
# - Add extjs library to alias
#
Alias /icinga-web/js/ext3 /usr/local/icinga-web/lib/ext3
Alias /icinga-web /usr/local/icinga-web/pub
<Directory /usr/local/icinga-web/lib/ext3>
    Order allow,deny
    Allow from all
</Directory>
    Alias /icinga-web /usr/local/icinga-web/pub
    <Directory /usr/local/icinga-web/pub>
        AllowOverride All
    </Directory>

```

Leeren Sie den Cache:

```
#> rm /usr/local/icinga-web/app/cache/config/*.php
```

oder /path/to/clearcache.sh

```
#> /usr/local/icinga-web/bin/clearcache.sh
```

und starten Sie den Webserver neu:

```
#> service apache2 restart
```

bzw.

```
#> /etc/init.d/apache2 restart
```

oder

```
#> /etc/init.d/httpd restart
```

7. Testen



Anmerkung

Bitte achten Sie darauf, dass Ihre Datenbank, Apache, IDOUtils und Icinga gestartet sind!

Öffnen Sie im Webbrower `http://localhost/icinga-web`. Überprüfen Sie, ob das Webinterface ohne Fehler ("exceptions") startet, und loggen Sie sich mit dem User 'root' und dem Passwort 'password' ein.

Viel Spaß :-)

8. Test & Fehler?

Diese Sammlung enthält einige nützliche Informationen, wie Sie Fehler finden und lösen können. Ebenso wird darauf hingewiesen, welche Informationen Sie im Fehlerfall angeben sollten, wenn Sie eine Frage auf den Mailinglisten oder auf <http://www.icinga-portal.org> stellen.

- * Geben Sie immer die verwendete Version an - tar.gz oder GIT?

- * Browser, Version

- * Falls das Problem mit der Datenquelle besteht: Ausführliche Informationen zu API, IDOUtils, Core (Version, Debug Logs).

Wo können Sie nachsehen?

- * Apache Error Logs, PHP Errors, PHP Dateien können nicht gefunden werden

- * `/var/log/messages`, `/path/to/icinga/var/icinga.log`

Änderungen an der Icinga-Web-Konfiguration (z.B. API/IDOUtils-Einstellungen geändert) werden nicht aktualisiert?

- * Löschen Sie den Konfigurationscache in `app/cache/config/*.php`

```
/usr/local/icinga-web/bin/clearcache.sh
```

Icinga-Web zeigt eine leere Seite?

- * Apache Errors Logs => mod_rewrite enabled, PHP-Abhängigkeiten ok? 'make testdeps'. Bei Debian gibt es u.a. die folgende Fehlermeldung: ".htaccess: Invalid command 'RewriteEngine', perhaps misspelled or defined by a module not included in the server configuration"

Icinga Web - Login erfolgt- aber die Seite lädt und lädt...

- * Request failed, Ressource `/icinga-web/appkit/ext/application` State could not be loaded
- is the url correct? => mod_rewrite enabled ?

- * .htaccess/VHost-Konfiguration => Pfade nicht korrekt? Icinga-Web zeigt keine Daten?
- * Datenbankzugriff verweigert => Überprüfen Sie, ob die Datenbankeinstellungen für Icinga-Web korrekt sind.

IDOUtils-Datenbank wird nicht gefüllt?

- * "Error writing to data sink" => Überprüfen Sie die IDOUtils (ido2db läuft 2x - ok?), ido2db.cfg debug_level=-1, debug_verbosity=2, restarten Sie IDOUtils und suchen Sie Fehler in ido2db.debug
- * Keine Daten vorhanden => Überprüfen Sie icinga.log, ob IDOMOD zu Beginn geladen wird. Falls nicht, enablen Sie das Event Broker Modul in icinga.cfg so wie im Icinga Core im [IDOUtils Quickstart Guide](#) beschrieben.
- * IDOUtils DB-Schema ist die aktuellste Version? => Falls nicht, überprüfen Sie mögliche Upgrades anhand der Docs dafür.
- * Sockets sind korrekt definiert? => Unix oder TCP Socket, TCP wahlweise mit oder ohne SSL

Testing the Web (stellen Sie sicher, dass PHPUnit installiert ist):

```
$> make test
```



Anmerkung

Falls Sie Ihren root-Benutzer zum Testen benutzen, dann stellen Sie vorher sicher, dass für den angegebenen Web-Benutzer in der /etc/passwd eine gültige Shell eingetragen ist. Andernfalls werden verschiedene Tests fehlschlagen. Abhängig von der verwendeten Icinga-Web-Version kann es sein, dass die Berechtigungen für die Dateien in /usr/local/icinga-web/log falsch gesetzt sind, so dass Icinga-Web mit der Meldung "Loading" stehenbleibt.



Anmerkung

Nicht vergessen - Ändern von PHP-Einstellungen in der php.ini erfordert einen Apache reload/restart!

- * PHP Fatal error: Allowed memory size of ... bytes exhausted (tried to allocate ... bytes) => Überprüfen Sie Ihre php.ini (apache2 und cli) und setzen Sie den Wert memory_limit auf 128M oder höher.
- * PHP Fatal error: Uncaught exception 'AgaviCacheException' with message 'Failed to write cache file /usr/local/icinga-web/app/cache/config/config_handlers.xml_development_xxxx.php' generated from configuration file /usr/local/icinga-web/app/config/config_handlers.xml". Please make sure you have set correct write permissions for directory /usr/local/icinga-web/app/cache.... => Setzen von safe_mode = off in /etc/php5/apache/php.ini.



Anmerkung

Das kann auch ein Hinweis auf ein aktiviertes SELinux sein.

- * [PHP Error] strtotime(): It is not safe to rely on the system's timezone settings. You are *required* to use the date.timezone setting or the date_default_timezone_set() function. In case you used any of those methods and you are still getting this warning, you most likely misspelled the timezone identifier. We selected 'Europe/Berlin' for 'CEST/2.0/DST' instead [line 1548 of /home/xxx/icinga/icinga-web/lib/doctrine/lib/Doctrine/Record.php] => Sie müssen date.timezone in der php.ini definieren.
 - * -> 500 internal server error! Uncaught exception AgaviException thrown! Your default timezone is 'System/Localtime', ... => Sie müssen date.timezone in der php.ini definieren.
 - * Keine Verbindung zur API. The API Connector returned the following message: getConnection failed: Database connection failed: SQLSTATE[28000] [1045] Access denied for user 'icinga'@'localhost' (using password: YES)
- => Überprüfen Sie Ihre IDOUtils DB-Referenzen in der ido2db.cfg und fügen Sie diese in Ihre Icinga-Web-Konfiguration als bevorzugte DB-Referenzen für Ihre IDO hinzu (siehe Konfiguration der Icinga-API). Ab Icinga-Web1.0.3 können die Werte direkt während des configure gesetzt werden.
- * touch: cannot touch '/usr/local/icinga-web/.../cache/testfile.txt': Permission denied
 - => Die Konfiguration in den xml-Dateien wird vom Framework ge-'pre-cached'. Daher müssen spezielle Berechtigungen für die Caching-Verzeichnisse gesetzt werden. Durch die Ausführung von icingaWebTesting.php in etc/tests können die richtigen Berechtigungen automatisch gesetzt werden.
 - * PHP Fatal error: Uncaught exception '...' with message 'Couldn't locate driver named mysql' => Stellen Sie sicher, dass php pdo installiert und geladen ist, auch wenn testdeps sagt, dass alles in Ordnung ist.
 - * Das Login wird nicht angezeigt => Aktivieren Sie short_open_tag in Ihrer php.ini.
 - => Editieren Sie open_basedir in Ihrer php.ini und fügen Sie die Installationsverzeichnisse von Icinga-Web und von Icinga-API (z.B. /usr/local/icinga/share/) hinzu.
 - * Leeres Icinga-Web? => Wenn mod_rewrite aktiviert ist und 'index.php' erscheint in der angefragten URL, dann funktioniert das Portal nicht. Entfernen Sie index.php aus Ihrer URL und alles sollte funktionieren
 - * Die Ergebnisse in den Status Cronks passen nicht zu Ihrer Konfiguration? => Überprüfen Sie in Ihrem Backend, d.h. IDOUtils DB, welche Werte für die Status-Tabellen selektiert werden.
 - * Keine Daten in den Cronks ? => Überprüfen Sie ob alle Berechtigungen korrekt gesetzt sind, insbesondere von log/. Weiterhin können Sie die angezeigten Daten in Ihrem Browser debuggen. Ausführliche Informationen dazu finden Sie im Icinga Wiki <https://wiki.icinga.org/display/Dev/Icinga+Web+Testing>
- Sofern Sie Fragen oder Updates haben, zögern Sie bitte nicht, uns diese mitzuteilen! :-)

[Zurück](#)[Nach oben](#)[Weiter](#)

Ausführen von CGIs auf der Kommandzeile

[Zum Anfang](#)Konfigurationsübersicht
Icinga-Web

© 2009-2011 Icinga Development Team, <http://www.icinga.org>

**Konfigurationsübersicht Icinga-Web**[Zurück](#)[Kapitel 6. Die Benutzeroberflächen](#)[Weiter](#)

Konfigurationsübersicht Icinga-Web

Wo sind meine Konfigurationsdateien?

Wir versuchen die Verwendung von globalen Konfigurationsdateien zu minimieren. Die Icinga-Web-Konfigurationseinstellungen für die Icinga-API, die Authentifizierung und die Datenbank werden in `etc/conf.d/` gespeichert. Diese Dateien werden bei einer Aktualisierung von Icinga-Web nicht überschrieben. Das Verzeichnis kann bei der Installation mit `--with-conf-folder` geändert werden.

Icinga-Web arbeitet mit Modulen und jedes Modul hat seine eigene Konfiguration. Das gilt ebenfalls für die Libraries. Wenn Sie weitere Informationen zu den Cronk-Libraries benötigen, schauen Sie in `app/modules/Cronks/lib` (für js in `app/modules/Cronks/lib/js`).

Ein Modul von Icinga-Web ist wie folgt aufgebaut:

```
tree -d -L 1 app/modules/AppKit/
app/modules/AppKit/
|-- actions
|-- config
|-- lib
|-- models
|-- templates
|-- validate
|-- views
```

Index

- [Globale Konfiguration](#)
 - [databases.xml](#)
 - [access.xml](#)
- [Session Cookie Lifetime](#)
- [Zeitzone](#)
- [Module-Konfiguration](#)
- [Authentifizierung](#)

- [Icinga-API Verbindungseinstellung](#)
- [Benutzerdefinierte Konfiguration](#)

Globale Konfiguration

app/config

Hier finden Sie die globalen Konfigurationsdateien für z.B. die Web-Session, den Icinga-Web-Pfad und die Datenbankinformationen.

Die wichtigsten Dateien:

- `databases.xml` - enthält die Verbindungseinstellungen für Ihre Icinga-Web-Datenbank

- *Allgemeine Einstellungen*

Die folgenden Einstellungen gelten für jede in Icinga-Web definierte Datenbank.

Wenn Sie sich die Standarddefinition der `icinga_web`-Datenbank ansehen, dann finden Sie dort die meisten Einstellungen einer generischen Datenbankverbindung:

```
<db:database name="icinga_web" class="AppKitDoctrineDatabase">
<ae:parameter name="dsn">mysql://icinga_web:icinga_web@localhost:3306/icinga_web</ae:parameter>
<ae:parameter name="charset">utf8</ae:parameter>
<ae:parameter name="manager_attributes">
<ae:parameter name="Doctrine_Core::ATTR_MODEL_LOADING">CONSERVATIVE</ae:parameter>
</ae:parameter>

<ae:parameter name="load_models">%core.module_dir%/AppKit/lib/database/models/generated</ae:parameter>
<ae:parameter name="models_directory">%core.module_dir%/AppKit/lib/database/models</ae:parameter>
<ae:parameter name="date_format"><![CDATA[YYYY-MM-DD HH24:MI:SS]]></ae:parameter>

<ae:parameter name="caching">
<ae:parameter name="enabled">false</ae:parameter>
<ae:parameter name="driver">apc</ae:parameter>
<ae:parameter name="use_query_cache">true</ae:parameter>
<ae:parameter name="use_result_cache">true</ae:parameter>
<ae:parameter name="result_cache_lifespan">60</ae:parameter>
</ae:parameter>
</db:database>
```

- Datenbank-Identifier

```
<db:database name="%Database name%" class="%Handler%">
  ...
</db:database>
```

Mit dem `db:database`-Tag informieren Sie Icinga-Web, dass es sich um eine Datenbankdefinition handelt. Dieses Tag muss zwei Attribute haben:

- name: Der Name für Ihre Datenbank. Zu dieser Zeit ($\geq v1.5$) ist 'icinga-web' für die interne Icinga-Web-Datenbank reserviert, die Benutzer- und Anmeldeinformationen, Persistenz-Einstellungen usw. enthält, und 'icinga' für die Datenbank, die von ido2db genutzt wird, um die Informationen von Icinga zu speichern.
- class: Benutzen Sie immer "AppKitDoctrineDatabase", nur die 'icinga'-Datenbank erfordert 'IcingaDoctrineDatabase' (siehe unten)
- dsn (Data Source Name):

Definiert die Anmeldeinformationen, den Typ und Standort der Datenbank:

Für MySQL, PostgreSQL und Oracle:

```
<db:database ...>
  ...
  <ae:parameter name="dsn">%driver%://%username%:%password%@%host%:%port%/%database name%</ae:parameter>
  ...
</db:database>
```

Für sqlite3:

```
<db:database ...>
  ...
  <ae:parameter name="dsn">sqlite:///path to your db file%</ae:parameter>
  ...
</db:database>
```



Anmerkung

SQLite-Datenbanken müssen für den Webserver-Benutzern les- und beschreibbar sein, außerdem das Verzeichnis, in dem sich die Datei befindet.

- %driver%: Das zu benutzende Datenbank-Backend
 - mysql: Eine MySQL-Datenbank
 - pgsql: Eine PostgreSQL-Datenbank
 - oracle: Eine Oracle-Datenbank, auf die über den [oci8](#)-Treiber zugegriffen wird

Benutzen Sie nicht den PDO-Treiber, denn dieser ist weit von produktiver Nutzung entfernt.

 - icingaOracle: Eine spezielle Implementation, die genutzt werden muss, wenn eine Icinga-Datenbank verwendet wird (einige Tabellennamen sind unterschiedlich und die Längen von Bezeichnern muss automatisch berücksichtigt werden, um sicherzustellen, dass sich kein Entwickler während der Entwicklung umbringt).
- %username%: Der Benutzername zur DB-Authentifizierung
- %password%: Das Passwort zur DB-Authentifizierung
- %host%: Der Host Ihres DB-Servers
- %port%: Der Port Ihres DB-Servers
 - mysql-default: 3306
 - postgresql-default: 5432
 - oracle-default: 1521
- %database name%: Der Name Ihrer Datenbank

- charset

```
<db:database ...>
  ...
    <ae:parameter name="charset">utf8</ae:parameter>
  ...
</db:database>
```

Im Allgemeinen sollte Ihr Datenbank-Zeichensatz utf8 sein.

- manager_attributes, load_models, models_directory

```
<db:database ...>
  ...
    <ae:parameter name="manager_attributes">
      <ae:parameter name="Doctrine_Core::ATTR_MODEL_LOADING">CONSERVATIVE</ae:parameter>
    </ae:parameter>
  ...
</db:database>
```

Sie können beruhigt diesen Abschnitt ignorieren und ihn einfach kopieren. Er enthält lediglich Doctrine-spezifische Einstellungen (insbesondere wie Datenbankmodelle geladen werden und wo sie zu finden sind. Wenn Sie mehr wissen möchten, dann schauen Sie in die [doctrine documentation](#)).

- date_formate

Dies ist erforderlich, um sauber auf Oracle-Datenbanken zuzugreifen. Es setzt das für die Datenbank zu benutzende Datumsformat.

```
<db:database ...>
  ...
    <ae:parameter name="date_format"><! [CDATA[ YYYY-MM-DD HH24:MI:SS ]]></ae:parameter>
  ...
</db:database>
```

- Caching

```
<db:database ...>
  ...
    <ae:parameter name="caching">
      <ae:parameter name="enabled">false</ae:parameter>
      <ae:parameter name="driver">apc</ae:parameter>
      <ae:parameter name="use_query_cache">true</ae:parameter>
      <ae:parameter name="use_result_cache">true</ae:parameter>
      <ae:parameter name="result_cache_lifespan">60</ae:parameter>
    </ae:parameter>
</db:database>
```

Wenn Sie apc oder memcache benutzen, dann können Sie entweder Datenbankabfragen oder -ergebnisse zwischenspeichern. Während das Zwischenspeichern von Abfragen eigentlich nie eine schlechte Sache ist (solange Sie nicht entwickeln), ist das Zwischenspeichern von Ergebnissen ein ziemlich spezieller Fall - denn Sie könnten veraltete Daten aus Ihrer Datenbank bekommen. **Also benutzen Sie es bitte nie für Ihre Icinga-Datenbank!**

- enabled: Caching für diese Datenbank ein- oder ausschalten
- driver: apc oder memcache, wobei memcache experimentell ist (also probieren Sie es aus !)

- `use_query_cache`: Abfragen zwischenspeichern (nur die SQL-Abfragen, nicht die Ergebnisse)
- `use_result_cache`: Datenbankergebnisse zwischenspeichern (**Gefahr!**)
- `result_cache_lifespan`: Wie lange Ergebnisse zwischengespeichert werden sollen (in Sekunden), wenn `use_result_cache` benutzt wird
- *Die "icinga_web"-Datenbank*

Sie müssen eine Icinga-Web-Datenbank haben, die Informationen über Benutzer, Anmeldeinformationen, View-Persistenz, etc. enthält. Diese Datenbank muss 'icinga_web' heißen. Alle Einstellungen sind im vorigen Abschnitt beschrieben.

- *Die "icinga"-Datenbank*

Ab Icinga-Web v1.5 wird die Icinga-API über Doctrine angesprochen (vorher gab es ein eigenes Projekt, die 'icinga-api'). Im Folgenden werden nur spezielle bzw. zusätzliche Einstellungen erklärt. Solange nicht anders erwähnt, gelten alle Regeln aus "[Allgemeine Einstellungen](#)".

- Datenbank-Identifier

Der Icinga-Datenbank-Identifier muss "IcingaDoctrineDatabase" als 'class'-Attribut und 'icinga' als Datenbankname benutzen, also:

```
<db:database name="icinga" class="IcingaDoctrineDatabase">
...
</db:database>
```

- dsn

Sie können die gleichen Anmeldeinformationen wie in der `ido2db.cfg` benutzen, aber als Sicherheitsgründen ist es sehr ratsam, einen Benutzer nur mit Leserechten für Icinga-Web anzulegen. Nachfolgend ein Beispiel, welche Werte aus der `ido2db.cfg` passen.

```
<ae:parameter name="dsn">mysql://db_user:db_pass@db_host:db_port/db_name</ae:parameter>
```

- prefix

Definiert den Präfix wie in `ido2db.cfg` angegeben

```
<db:database name="icinga" class="IcingaDoctrineDatabase">
...
    <ae:parameter name="prefix">icinga_</ae:parameter>
...
</db:database>
```

Für Oracle benutzen Sie einen leeren Wert

```
<db:database name="icinga" class="IcingaDoctrineDatabase">
...
    <ae:parameter name="prefix"></ae:parameter>
...
</db:database>
```

- `use_retained`

Whether to use retained or original data dumps (see the value in idomod.cfg)

```
<db:database name="icinga" class="IcingaDoctrineDatabase">
...
    <ae:parameter name="use_retained">true</ae:parameter>
</db:database>
```

- *Komplettes Listing*

```
<?xml version="1.0" encoding="UTF-8"?>
<databases xmlns:db="http://agavi.org/agavi/config/parts/databases/1.0" xmlns:ae="http://agavi.org/agavi/config/global/envelope/1.0">

    <db:database name="icinga_web" class="AppKitDoctrineDatabase">
        <ae:parameter name="dsn">mysql://icinga_web:icinga_web@localhost:3306/icinga_web</ae:parameter>
        <ae:parameter name="charset">utf8</ae:parameter>
        <ae:parameter name="manager_attributes">
            <ae:parameter name="Doctrine_Core::ATTR_MODEL_LOADING">CONSERVATIVE</ae:parameter>
        </ae:parameter>

        <ae:parameter name="load_models">%core.module_dir%/AppKit/lib/database/models/generated</ae:parameter>
        <ae:parameter name="models_directory">%core.module_dir%/AppKit/lib/database/models</ae:parameter>
        <ae:parameter name="date_format"><![CDATA[YYYY-MM-DD HH24:MI:SS]]></ae:parameter>

        <ae:parameter name="caching">
            <ae:parameter name="enabled">false</ae:parameter>
            <ae:parameter name="driver">apc</ae:parameter>
            <ae:parameter name="use_query_cache">true</ae:parameter>
            <ae:parameter name="use_result_cache">true</ae:parameter>
            <ae:parameter name="result_cache_lifespan">60</ae:parameter>
        </ae:parameter>
    </db:database>

    <db:database xmlns="http://agavi.org/agavi/config/parts/databases/1.0" name="icinga" class="IcingaDoctrineDatabase">
        <ae:parameter name="dsn">mysql://icinga:icinga@localhost:3306/icinga</ae:parameter>
        <ae:parameter name="prefix">icinga_</ae:parameter>
        <ae:parameter name="charset">utf8</ae:parameter>
        <ae:parameter name="use_retained">true</ae:parameter>
        <ae:parameter name="manager_attributes">
            <ae:parameter name="Doctrine_Core::ATTR_MODEL_LOADING">CONSERVATIVE</ae:parameter>
        </ae:parameter>
        <ae:parameter name="load_models">%core.module_dir%/Api/lib/database/models/generated</ae:parameter>
        <ae:parameter name="models_directory">%core.module_dir%/Api/lib/database/models</ae:parameter>
        <ae:parameter name="caching">
            <ae:parameter name="enabled">false</ae:parameter>
            <ae:parameter name="driver">apc</ae:parameter>
            <ae:parameter name="use_query_cache">true</ae:parameter>
        </ae:parameter>
    </db:database>
</databases>
```

- *access.xml*

Die Konfigurationsdatei *access.xml* hat zwei Hauptzwecke:

- definieren von Hosts, auf die von Icinga-Web (via System oder ssh) zugegriffen werden kann und definieren von Dateien, auf die lesend/schreibend zugegriffen werden kann bzw. auszuführenden Programmen
- zuordnen von Icinga-Instanzen zu diesen Hosts

Die *access.xml*-Konfiguration wird von Icinga-Webs Console-Handler-Implementation genutzt und bietet zusätzliche Sicherheit, weil sie willkürliche Ausführung/Manipulation von Dateien in letzter Instanz verhindert.

Abschnitte

- *Instanzen*

Das Einrichten von Instanzen erfolgt im *<instances>*-Abschnitt:

Beispiel: Zuordnen der Instanz 'default' zum Host 'localhost'

```
<!-- Map your instances to hosts here -->
<instances>
    <instance name="default">localhost</instance>
</instances>
```

Zulässige Unterelemente

- instance: Nur "instance" ist als Unterelement zulässig. Dieser Knoten ordnet einem Host (s.u.) einen Instanznamen zu, der durch den Wert des Attributs "name" festgelegt ist:

```
<instance name="%icinga instance name%">%host name%</instance>
```

○ Standardwerte

Der "default"-Abschnitt definiert Standardwerte für Host-Einstellungen, z.B. auf welche Dateien schreibend/lesend zugegriffen werden bzw. welche ausgeführt werden können. Auch der Standard-Host, der von Icinga-Web benutzt werden soll, wird hier definiert.

Zulässige Unterelemente

- defaultHost: definiert den Host, der für Aktionen benutzt wird, wenn kein Host angegeben ist. Momentan (v1.6) ist dies nur für einige Module wichtig und es betrifft nicht Icinga-Webs Standardleistungsumfang.

```
<defaultHost>localhost</defaultHost>
```

Der Host muss im hosts-Abschnitt definiert werden.

- access: Icinga-Web (und Module, wenn sie korrekt geschrieben sind) greifen nicht direkt auf System-Ressourcen zu, sondern über ein Console-Interface, das sich um die Sicherheit kümmert. Im Zugriffsabschnitt definieren Sie, welche System-Ressources von dieser Behandlungsroutine aufgerufen werden dürfen und wo sie sich befinden. Außerdem geben Sie System-Ressourcen einen Ressourcen-Namen. Der Console-Handler bspw. greift auf die Icinga-Pipe nur über den Ressource-Namen "icinga_pipe" zu.

Der Zugriffsabschnitt hat vier zulässige Unterelemente: 'readwrite', 'read', 'write' und 'execute', die die Zugriffsebene für darunterliegende Ressourcen definieren. Ressourcen selbst werden unter den Tags 'folders' oder 'files' definiert und mit 'resource' eingeschlossen: Das folgende Beispiel erlaubt Schreib-/Lesezugriff auf Icingas objects-Ordner. Schreiben, Lesen und Ausführen funktionieren analog, das 'readwrite'-Tag wird jeweils durch das entsprechende Tag ausgetauscht.

```
<readwrite>
  <folders>
    <resource name="icinga_objects">/usr/local/icinga/etc/objects</resource>
  </folders>
  <files>
    <resource name="icinga_cfg">/usr/local/icinga/etc/icinga.cfg</resource>
  </files>
</readwrite>
```

Wie oben erwähnt ordnet jede Ressource ein Ressourcen-Symbol (Name) einem Pfad zu. Dies ist optional, wird aber empfohlen. Wenn ein Symbol mehrfach existiert, dann wird das letzte benutzt.



Anmerkung

Eigene Default-Definitionen in Ihrem Konfigurationsordner

If you override a custom section, like `readwrite`, all previously set definitions will be ignored and must be redefined if you want to use them. Das betrifft nicht Modul-Konfigurationen. This doesn't affect module configurations.

- *Hosts*

Im host-Abschnitt werden Zugriffsmethoden und Anmeldeinformation, aber auch spezifische Rechte für Hosts definiert.

Zulässige Unterelemente:

- `host`: definiert einen einzelnen Host und gibt ihm einen Namen, der benutzt werden kann, um auf diesen Host zu referenzieren (wie z.B. im "instance"-Abschnitt).

```
<!-- Example for ssh connection with user/password auth -->
<host name="vm_host1">
    <type>ssh</type>
    <ssh-config>
        <host>localhost</host>
        <port>22</port>
        <auth>
            <type>password</type>
            <user>john_doe</user>
            <password>foobar</password>
        </auth>
    </ssh-config>

    <access useDefaults="true" />
</host>
```

Das 'host'-Tag hat eine Reihe von Unterelementen:

- `type`: Entweder "local" oder "ssh". Definiert, ob Befehl direkt auf der Maschine ausgeführt werden, auf der Icinga-Web läuft, oder via ssh.
- `access`: sh. den "access"-Abschnitt weiter oben. Zusätzlich kann 'useDefaults' gesetzt werden, um dem Command-Interface mitzuteilen, dass alle Standardzugriffsdefinitionen benutzt werden sollen (wenn sie nicht in diesem Abschnitt überschrieben werden).
- `ssh-config`: (nur falls `type:ssh`) Dies legt fest, wie auf den Host per ssh zugegriffen werden soll. Es gibt drei Authentifizierungsmethoden für ssh: 'none', 'password' oder 'key', die im folgenden Abschnitt erläutert werden. Unabhängig davon, welche Methode Sie wählen, müssen die 'host'- und 'port'-Einträge für jeden Host gesetzt werden.

- *SSH-Config auth*

Dieser Abschnitt erklärt den Authentifizierungsblock der ssh-Konfigurationseinstellungen, die wir im vorigen Abschnitt angesprochen haben. Die Elemente unterhalb des 'auth'-Tags sind abhängig vom ausgewählten Authentifizierungstyp.

- Auth type: none

Dies teilt der Console-Verbindung mit, dass nur ein Benutzername für die Authentifizierung verwendet wird:

```
<ssh-config>
  <host>localhost</host>
  <port>22</port>
  <auth>
    <type>none</type>
    <user>john_doe</user>
  </auth>
</ssh-config>
```

In diesem Beispiel wird nur der Benutzername 'john_doe' für die Authentifizierung geliefert. Dies kann sinnvoll sein, wenn Ihre Maschinen automatische Schlüssel-Authentifizierung benutzen.

- Auth type: password

Dies teilt der Console-Verbindung mit, dass Benutzername und Passwort für die Authentifizierung verwendet werden:

```
<ssh-config>
  <host>localhost</host>
  <port>22</port>
  <auth>
    <type>password</type>
    <user>john_doe</user>
    <password>foobar</password>
  </auth>
</ssh-config>
```

In diesem Beispiel werden nur der Benutzername 'john_doe' und das Passwort 'foobar' für die Authentifizierung verwendet.

- Auth type: key



Wichtig

Experimentell! Probieren Sie es aus und öffnen Sie ein Ticket, wenn Sie Probleme haben!

Dies teilt der Console-Verbindung mit, eine Schlüsseldatei für die Authentifizierung zu benutzen, die optional mit einem Password geschützt ist

```
<ssh-config>
  <host>localhost</host>
  <port>22</port>
  <auth>
    <type>key</type>
    <user>testuser</user>
    <private-key>/usr/local/icinga-web/app/modules/Api/lib/.ssh/host1_rsa</private-key>
    <password>secret123</password>
  </auth>
</ssh-config>
```

Hier wird der private Schlüssel benutzt, der im 'private-key'-Tag definiert wird. Sie sollten und müssen diesen Pfad *nicht* im host-Zugriffsabschnitt definieren!

Fehlersuche

- Ich kann keine Befehle senden/ausführen!

Stellen Sie sicher, dass der WebServer-Benutzer Schreibberechtigung auf die Icinga-Pipe besitzt und dass das "icinga_pipe"-Symbol im "readwrite"- oder "write"-Abschnitt Ihres Host (oder im default-Abschnitt) vorhanden ist. Stellen Sie sicher, dass der Pfad korrekt ist.

- `factories.xml` - enthält die Konfiguration für Ihre Web Session
- `icinga.xml` - enthält die Konfiguration für Ihr Icinga-Web ROOT-Verzeichnis und den Icinga-Web Pfad.

Session Cookie Lifetime

Beispiel: Ändern von `session_cookie_lifetime`

Die Session Lifetime ist die Zeit in Sekunden, bis die Icinga-Web- Session abläuft. Sie kann auf globaler Ebene in der Datei `app/config/factories.xml` konfiguriert werden.

```
#> vi app/config/factories.xml
<ae:parameter name="session_cookie_lifetime">3600</ae:parameter>
```

Wenn Sie die `session_cookie_lifetime` ändern möchten, editieren Sie bitte `app/config/factories.site.xml`

Icinga-Web- Zeitzone

Beispiel: Ändern der Zeitzone für Icinga-Web

Wenn die Zeitzone von Icinga-Web von Ihrer lokalen Zeitzone abweicht, überprüfen Sie bitte den Parameter "date.timezone" in `app/modules/AppKit/config/module.xml` (z.B. 'Europe/Berlin')

```
#> vi app/modules/AppKit/config/module.xml
<ae:parameter name="date.timezone">GMT</ae:parameter>
```

Modul-Konfiguration:

- **app/modules/AppKit**

Hier "lebt" das Framework: Authentifikation, Menüs und weiteres.

Authentifizierung

Beispiel: LDAP-Authentifizierung

Öffnen Sie `app/modules/AppKit/config/auth.xml`.

Ein Anbieter ist wie folgt aufgebaut:

```
<ae:parameter name="msad-ldap1">
  <ae:parameter name="auth_module">AppKit</ae:parameter>
  <ae:parameter name="auth_provider">Auth.Provider.LDAP</ae:parameter>
  <ae:parameter name="auth_enable">true</ae:parameter>
  <ae:parameter name="auth_authoritative">true</ae:parameter>
  <ae:parameter name="auth_create">true</ae:parameter>
  <ae:parameter name="auth_update">true</ae:parameter>

  <ae:parameter name="auth_map">
    <ae:parameter name="user_firstname">givenName</ae:parameter>
```

```

<ae:parameter name="user_lastname">sn</ae:parameter>
<ae:parameter name="user_email">mail</ae:parameter>
</ae:parameter>

<ae:parameter name="ldap_dsn">ldap://ad.icinga.org</ae:parameter>
<ae:parameter name="ldap_basedn">DC=ad,DC=icinga,DC=org</ae:parameter>
<ae:parameter name="ldap_binddn">ldap@AD.icinga.org</ae:parameter>
<ae:parameter name="ldap_bindpw"><![CDATA[XXXXXXXXXX]]></ae:parameter>
<ae:parameter name="ldap_userattr">uid</ae:parameter>
<ae:parameter name="ldap_filter_user"><![CDATA[(&(sAMAccountName=__USERNAME__))]]></ae:parameter>
</ae:parameter>

```

Die auth.xml hält die Dokumentation für die globale Konfiguration. Die LDAP-Authentifizierung sollte mit einigen grundlegenden LDAP-Kenntnissen möglich sein.

Sie können die Anbieter duplizieren und so Ihre Authentifizierungs-Basis vergrößern.

Bitte speichern Sie Ihre Konfiguration in auth.site.xml !

● app/modules/Cronks

Alle Cronks werden hier implementiert: Grids und iframes. Sie sind einfache HTML-Seiten, die ExtJS-Komponenten-Code enthalten. Wenn Sie einen neuen Cronk hinzufügen möchten, wird dieses Modul Ihr Freund sein.

Wenn Sie einen neuen Cronk entwickeln möchten, schauen Sie hier: [HowToDevelopCronks](#)

Die Konfiguration können Sie im Cronk-Module ändern:

```
#> ls app/modules/Cronks/config
autoload.xml config_handlers.xml cronks.xml module.xml validators.xml
```

- module.xml - definieren von neuen Kategorien in denen die Cronks erscheinen, die Datei module.xml hält dazu alle Informationen
- cronks.xml - um auf neue Cronks zu zugreifen, definieren von neuen iframe Cronks

● app/modules/Web

Oder besser: **Icinga**. Dieses Modul enthält alle Icinga relevanten Dinge wie IcingaAPI2Json und die Statusinformationen. Auch die Icinga-API-Verbindungseinstellungen werden hier konfiguriert.

Icinga-API Verbindungseinstellungen

Beispiel: Ändern der Icinga-API-Verbindungseinstellungen

Sehen Sie in app/modules/Web/config/icinga-io.xml, hier finden Sie die Standardeinstellungen für die Icinga-API-Verbindung.

```
#> vi app/modules/Web/config/icinga-io.xml

<setting name="api.interfaces.data">
    <!-- IcingaApi connection interface -->
    <ae:parameter name="api_type">IcingaApi::CONNECTION_IDO</ae:parameter>

    <!-- Suits for all interfaces -->
    <ae:parameter name="config_type">mysql</ae:parameter>
    <ae:parameter name="config_host">localhost</ae:parameter>
    <ae:parameter name="config_port">3306</ae:parameter>

    <!-- ###BEGIN_CONNECTION_IDO## -->
    <!-- Database specific (IcingaApi::CONNECTION_IDO) -->
    <ae:parameter name="config_database">icinga</ae:parameter>
    <ae:parameter name="config_user">icinga</ae:parameter>
    <ae:parameter name="config_password">icinga</ae:parameter>
    <ae:parameter name="config_table_prefix">icinga_</ae:parameter>
    <!-- ###END_CONNECTION_IDO## -->
```

Wenn Sie diese Einstellungen ändern möchten, editieren Sie bitte `app/modules/Web/config/icinga-io.site.xml`.

Benutzerdefinierte Konfiguration

Bitte beachten:

Wenn Sie Konfigurationsdateien ändern oder erstellen, denken Sie bitte an folgendes:

- Zeilen die mit <!-- beginnen und enden mit --> werden als Kommentare interpretiert.
- Variablennamen sind case-sensitive
- Die Icinga-Web-Konfigurationseinstellungen für die Icinga-API, die Authentifizierung und die Datenbank werden in `etc/conf.d/` gespeichert (vor v1.5 waren sie in `app/modules/Appkit(/Cronks/Web)/config/` abgelegt). Diese Dateien werden bei einer Aktualisierung von Icinga-Web nicht überschrieben. Das Verzeichnis kann bei der Installation mit `--with-conf-folder` geändert werden.



Anmerkung

Nach dem Ändern von Konfigurationsdateien leeren Sie bitte den Cache!

```
#> rm -rf app/cache/config/*.php
```

oder

```
#> /usr/local/icinga-web/bin/clearcache.sh
```

Benötigen Sie weitere Informationen? Schauen Sie bitte in unserem [Development Wiki](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Installation des Icinga-Web
Frontend

[Zum Anfang](#)

Aktualisierung von Icinga-Web
und Icinga-Web Datenbank

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Aktualisierung von Icinga-Web und Icinga-Web Datenbank

[Zurück](#)[Kapitel 6. Die Benutzeroberflächen](#)[Weiter](#)

Aktualisierung von Icinga-Web und Icinga-Web Datenbank

Aktualisieren von Icinga-Web

Sobald neuere Versionen von Icinga-Web herauskommen, sollten Sie dringend über eine Aktualisierung nachdenken. Neuere Ausgaben enthalten Behebungen von kritischen Fehlern, so dass es wichtig ist, aktuell zu sein. Wenn Sie bereits Icinga-Web, wie in den Schnellstartanleitungen beschrieben, aus dem Quellcode installiert haben, dann können Sie einfach neuere Versionen installieren.

Stellen Sie sicher, dass Sie eine vollständige Datensicherung Ihrer bestehenden Icinga-Web-Installation und der Konfigurationsdateien haben (wenn Sie Ihre benutzerdefinierte Konfiguration in *.site.xml- Dateien erstellt haben, so werden diese während der Aktualisierung nicht überschrieben!). Wenn irgendetwas schief geht oder nicht funktioniert, dann können Sie auf diese Weise schnell Ihre alte Icinga-Web-Version wiederherstellen.

Bitte denken Sie daran, dass `configure` mit den gleichen Optionen wie bei der vorherigen Installation aufzurufen ist! Die verwendeten Optionen sehen Sie im `config.log`.



Anmerkung

Bitte beachten Sie, dass es einige wesentliche Änderungen mit Icinga 1.5 gab, die ggf. zusätzliche Aktionen erfordern könnten:

- Die *.site.xml-Datei für lokale Konfigurationsanpassungen wurden entfernt, Konfigurationen werden nun unter etc/conf.d/ abgelegt (oder dem Pfad, der während der Installation mit --with-conf-folder angegeben wurde)
- Das Log-Verzeichnis befindet sich nun unter log/ (oder dem Pfad, der während der Installation mit --with-log-folder angegeben wurde)
- Die Icinga-API wurde entfernt und nun wird Doctrine für alle Datenbank-Transaktionen benutzt (icinga-io.xml ist nun databases.xml)
- Befehle werden nun durch den internen Console-Handler versandt (weil die Icinga-API entfernt wurde)

Falls Sie Icinga-Web <= 1.4x benutzen, dann empfehlen wir eine Neuinstallation anstatt einer Aktualisierung. Es ist besser, Ihre alten Konfigurationsdateien in die neue Struktur zu kopieren, weil es viele geänderte und gelöschte Dateien gibt, um die wir uns bei einer Aktualisierung ggf. nicht kümmern.

Die Aktualisierung von Icinga-Web

Bitte laden Sie das Archiv von <http://sourceforge.net/projects/icinga/files/> herunter. Falls Sie ein Problem haben, das in der aktuellen Entwicklerversion behoben ist, dann klonen Sie den aktuellsten git- Branch mit :

```
#> git clone git://git.icinga.org/icinga-web.git
```

Entpacken Sie das Archiv (tarball):

```
#> tar xzvf icinga-web-1.6.0.tar.gz
```

Wechseln Sie in das Verzeichnis

```
#> cd icinga-web-1.6.0
```

Icinga-Web bietet diverse Konfigurationsoptionen an:

```
#> ./configure
--prefix=/usr/local/icinga-web
--with-web-user=www-data
--with-web-group=www-data
--with-web-path=/icinga-web
--with-web-apache-path=/etc/apache2/conf.d
--with-db-type=mysql
--with-db-host=localhost
--with-db-port=3306
--with-db-name=icinga_web
--with-db-user=icinga_web
--with-db-pass=icinga_web
--with-icinga-api=/usr/local/icinga/share/icinga-api
--with-api-type=APICON API type (default CONNECTION_IDO)
--with-api-subtype=TYPE DB driver or network connection
--with-api-host=HOST Host to connect (DB or other) (default localhost)
--with-api-port=PORT Port for connection (default 3306)
--with-api-socket=PATH Path to socket (default none)
```



Anmerkung

Bitte beachten Sie, dass Sie hier die Icinga-Web-Datenbank konfigurieren, und nicht die Icinga-IDOUtils-Datenbank! User- und Gruppenname des Web-Prozesses sind abhängig von der verwendeten Distribution.

Alle configure- Optionen sehen Sie mit:

```
#> ./configure --help
```



Anmerkung

Wenn Sie configure ohne weitere Optionen ausführen, erwartet der Installer die Icinga-API unter /usr/local/icinga/share/icinga-api.

Die Aktualisierung von Icinga-Web unter /usr/local/icinga-web erfolgt mit:

```
#> ./configure
#> make upgrade
```



Anmerkung

Bitte denken Sie daran den Cache zu leeren!

```
#> rm -rf app/cache/config/*.php
```

oder /path/to/clearcache.sh

```
#> /usr/local/icinga-web/bin/clearcache.sh
```

Das war's, Sie können sich nun an Ihrem aktualisierten Icinga-Web anmelden.

Bekannte Fehler sind [hier](#) beschrieben.

Aktualisieren der Icinga-Web Datenbank

Die Aktualisierung Ihrer Icinga-Web Datenbank ist optional, aber es mag einen Bug im Datenbankschema geben, der behoben wurde. Wenn Sie eine ältere Icinga-Web Datenbank-Version aktualisieren, dann müssen Sie außerdem diese Anpassungen manuell ausführen. Wenn Sie rpm/deb-Pakete benutzen, lesen Sie bitte die Hinweise und/oder fragen Sie den Maintainer, der diese Anpassungen in der Installationsroutine hinzugefügt hat.



Anmerkung

Abhängig von den Änderungen und der Größe Ihrer Datenbank kann es eine Weile dauern, die Anpassungen durchzuführen. Bitte haben Sie ein wenig Geduld und brechen Sie das Script nicht ab, weil sonst ggf. Ihre Daten unbrauchbar sind.

Die Update-Dateien finden Sie zusammen mit den Datenbank-Installationsdateien in /path/to/icinga-web/etc/schema/updates

Die Syntax ist wie folgt

```
<rdbm>_<alteVersion>_to_<neueVersion>.sql
```

wobei <rdbm> mysql, pgsql oder oracle sein kann und <neueVersion> zeigt auf die Version, auf die Sie aktualisieren wollen.



Anmerkung

Wenn Sie eine ältere Version aktualisieren wollen und zwischen dieser und der aktuellen noch andere Versionen liegen, dann sollten Sie beachten, dass Sie auch die dazwischen liegenden Updates inkrementell installieren müssen!

1. Sichern Sie Ihre aktuelle Datenbank vor der Aktualisierung!
2. Prüfen Sie die laufende Icinga-Web Datenbank- Version und die Zielversion. Prüfen Sie, ob zwischen diesen beiden Versionen noch andere Versionen liegen und aktualisieren Sie ggf. schrittweise.
3. Führen Sie die Aktualisierung(en) mit einem Benutzer durch, der über die notwendigen Berechtigungen verfügt.

MySQL

```
$ mysql -u root -p icinga_web < /path/to/icinga-web/etc/schema/updates/mysql_<alteVersion>_to_<neueVersion>.sql
```

Postgresql

```
#> su - postgres
$ psql -U icinga_web -d icinga_web < /path/to/icinga-web/etc/schema/updates/pgsql_<alteVersion>_to_<neueVersion>.sql
```

Oracle

```
#> su - oracle
$ sqlplus dbuser/dbpass
SQL> @oracle_<alteVersion>_to_<neueVersion>.sql
```

[Zurück](#)
[Nach oben](#)
[Weiter](#)
[Konfigurationsübersicht
Icinga-Web](#)
[Zum Anfang](#)
[Einführung in Icinga-Web](#)



Einführung in Icinga-Web

[Zurück](#)

Kapitel 6. Die Benutzeroberflächen

[Weiter](#)

Einführung in Icinga-Web

Überblick

Diese Einführung wird Sie auf eine kurze Tour einladen, um diverse Aspekte von Icinga-Web zu skizzieren. Es besteht kein Anspruch auf Vollständigkeit, da dieses Dokument stetig erweitert wird.

Bedingt durch ständige Weiterentwicklung ändern sich einige Dinge mit der Zeit und Optionen werden hinzugefügt. Bitte werfen Sie einen Blick auf die aktuelle [Version 1.6](#).

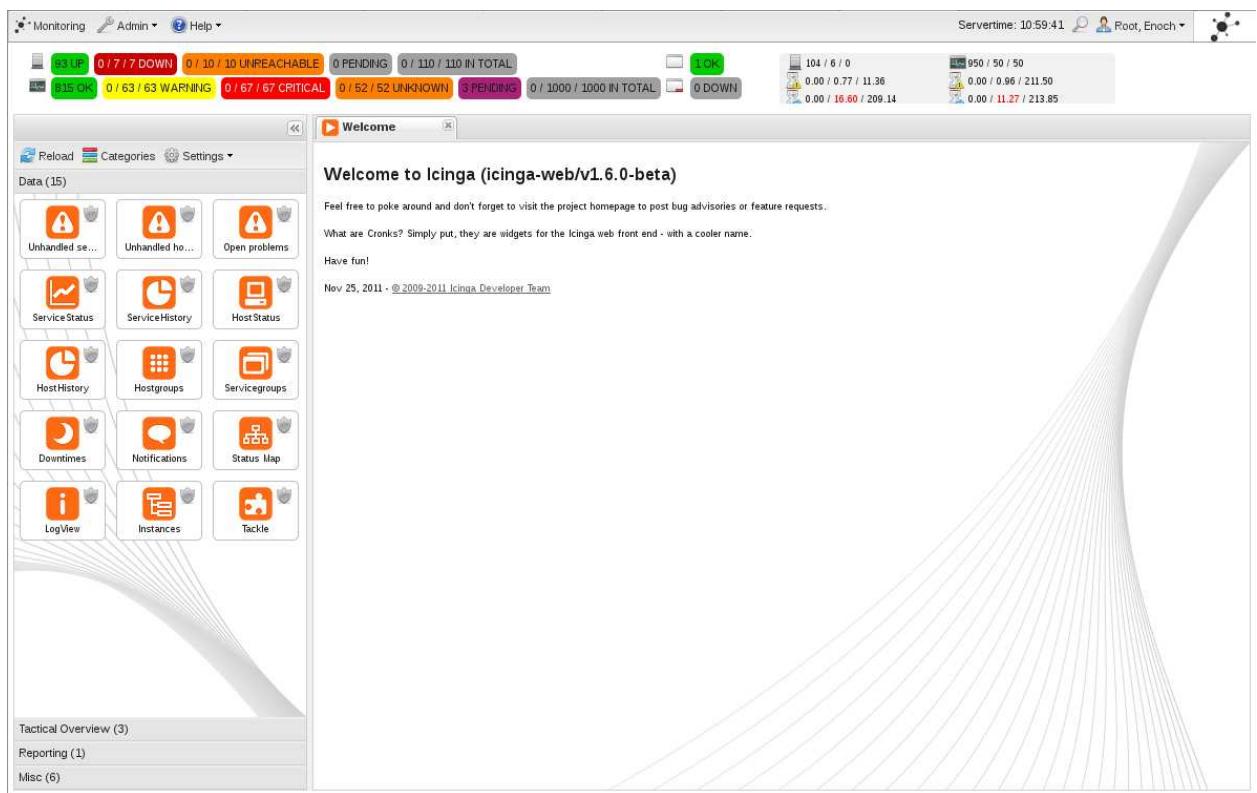
Einführung in Icinga-Web (>= 1.6)

Wenn Sie den Schnellstart-Anleitungen ([IDOUtils](#) und [Icinga-Web](#)) gefolgt sind, sollten Sie nun `http://<icinga server>/icinga-web` aufrufen können und den Login-Bildschirm sehen

Abbildung 6.1. Icinga-Web Login-Bildschirm

Sie können sich mittels "root" und "password" einloggen. Dies wird Sie auf eine Überblickseite weiterleiten, wo der Status von überwachten Hosts und Services dargestellt wird.

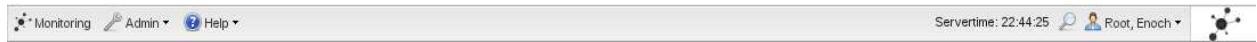
Abbildung 6.2. Icinga-Web Überblick



Oberes Menü

Das obere Menü beherbergt generelle Informationen zu Icinga sowie ein Benutzer-, Gruppen-, Log- sowie Task-Administrierungs menü. Die Live-Such öffnet sich, wenn Sie auf das Vergrößerungsglas neben der Uhrzeit klicken. Rechts oben können Sie den gerade eingeloggten Benutzer bearbeiten oder sich aus Icinga-Web ausloggen.

Abbildung 6.3. Icinga-Web Top-Menü



Suche

Die Suche zeigt live Resultate, während Sie tippen. Diese werden in einem eigenen Inlay-Fenster angezeigt. Indem Sie ein Ergebnis anklicken, wird ein neuer Tab mit näheren Informationen geöffnet. Die Suche beginnt am Anfang der Zeichenkette, aber Sie können auch Wildcards benutzen.

Abbildung 6.4. Icinga-Web Live-Suche

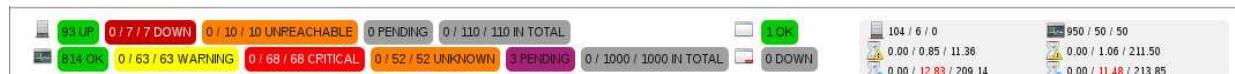
Type	Name	Status	Info	Attempt
Type: host [100 Items]				
host_000(127.0.0.1) random_000		UNREACHABLE		1 / 3
host_001(127.0.1.2) down_001		DOWN		1 / 3
host_002(127.0.2.3) up_002		UP		1 / 3
host_003(127.0.3.4) pending_003		UP		1 / 3
host_004(127.0.4.5) random_004		UP		1 / 3
host_005(127.0.5.6) random_005		UP		1 / 3
host_006(127.0.6.7) up_006		UP		1 / 3
host_007(127.0.7.8) up_007		UP		1 / 3
host_008(127.0.8.9)		UP		1 / 3

Status-Cronk

Der Status-Cronk zeigt die aktuelle Anzahl von Hosts und Services aufgeteilt nach ihren aktuellen Zuständen. Sofern ein Zähler Null (0) ist, wird dieser grau hinterlegt angezeigt. Klicken Sie auf einen Zustand, um einen neuen Tab zu öffnen, der nur den gewählten Zustand anzeigt. Daneben finden Sie die Anzahl und den Zustand der Icinga-Instanzen. Der verbleibende Platz wird für die Anzeige allgemeiner Informationen benutzt, das sind:

- Hosts | Services (aktiv/passiv)
- Host | Service Ausführungszeit (min/avg/max)
- Host | Services Latenzzeit (min/avg/max)

Abbildung 6.5. Icinga-Web Status-Cronk



Linkes Menü

Im linken (versteckbaren) Menü können Sie aus verschiedenen Widgets (wir nennen diese "Cronks") auswählen, die das Arbeiten mit Icinga-Web erleichtern werden. Sie können entweder einen Cronk doppelklicken oder diesen mit dem Mauszeiger in die zentrale Tab-Leiste ziehen. Die Kategorien sind wie folgt aufgeteilt:

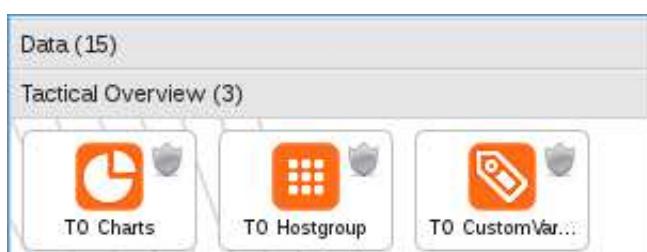
- "Data", um Status-, historische und Konfigurationsdaten zu erhalten. In Icinga 1.6 gibt es zwei neue Cronks: eins für Informationen über Instanzen und eins für die "Tackle View"

Abbildung 6.6. Icinga-Web Data-Cronks



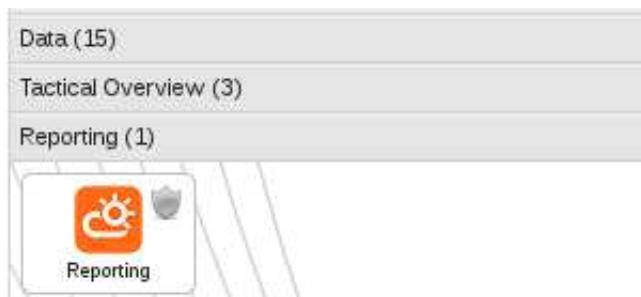
- "Tactical Overview" bietet generelle Charts und eigene angepasste an (Custom Variables)

Abbildung 6.7. Icinga-Web Tactical Overview-Cronks



- "Reporting" bietet Auswertungen

Abbildung 6.8. Icinga-Web Reporting-Cronks



- "Misc" enthält verschiedene nützliche Cronks, wie iframe für externe Webseiten oder 1,2,3-Spaltenansichten sowie Links auf die englische bzw. deutsche Dokumentation

Abbildung 6.9. Icinga-Web "Misc"-Cronks



Zentrale Übersicht

Hier findet sich alles zusammen: Sie können mittels drag-and-drop neue Fenster hereinziehen. Die Ansichten für den aktuellen Benutzer sind konfigurierbar (diese sind persistent), z.B. durch verschieben der Spaltenüberschrift an die gewünschte Stelle. Rechtsklicken auf die Spaltenüberschrift ermöglicht die Einstellung der Sortierreihenfolge oder das Verstecken von Spalten. Zu öffnende Suchresultate werden ebenfalls hier in einem neuen Tab geöffnet; das gilt auch für das Öffnen verfügbarer Cronks.

Abbildung 6.10. Icinga-Web Zentrale Übersicht

Host	Service	Host status	Service status	Last che...	Last che...	Attempt(...)	Attempt(...)	Hostche...	Servicec...
host_088	warning_...	UNREACHABLE	WARNING	2011-11...	2011-11...	1/5	1/3	host_088...	host_088...

Host Status

Die HostStatus-Ansicht zeigt den Host-Status und darunter die Anzahl von Warning/Critical der zugehörigen Services.

Abbildung 6.11. Icinga-Web Host-Status

	Host		Status	Last check	Duration	Info	Attempt	Output
	host_057		UP	2011-11-25 22:5...	38m 23s		1 / 5	host_057 (che...
	host_061		UNREACHABLE	2011-11-25 22:5...	19s		2 / 5	host_061 (che...
	host_058		UP	2011-11-25 22:5...	38m 21s		1 / 5	host_058 (che...

Service Status

Die ServiceStatus-Ansicht zeigt die Services für jeden Host.

Abbildung 6.12. Icinga-Web Service-Status

	Service		Status	Last check	Duration	Info	Attempt	Output
	warning_09		WARNING	2011-11-25 22:5...	42m 25s		1 / 3	host_000 (che...

Hostgroup Status

Die Hostgroup-Ansicht zeigt den Status und die Anzahl der Hosts mit dem jeweiligen Status gruppiert nach den Hostgruppen.

Abbildung 6.13. Icinga-Web Hostgruppenstatus

	Alias		Status	Count
Hostgroup: down [1 Item]				
	All Down Hosts		DOWN	5
Hostgroup: flap [2 Items]				
	All Flapping Hosts		UP	3
	All Flapping Hosts		UNREACHABLE	2

Log

Am unteren Fensterrand wird das aktuelle Icinga-Protokoll eingeblendet. Das Log wird automatisch aktualisiert, und kann auch minimiert werden, um mehr Platz für die zentrale Übersicht zu schaffen.

Abbildung 6.14. Icinga-Web Log

Timestamp			Type	Message
Hour: 2011-11-25 22:00:00 (25 Items)				
2011-11-25 22:58:07		host down		HOST ALERT: host_094;DOWN;SOFT;3;host_094 (checked by pc3239f15) DOWN: up hostcheck: parent host down
2011-11-25 22:58:07		host up		HOST ALERT: host_039;UP;SOFT;3;host_039 (checked by pc3239f15) FLAP: flap hostcheck up

Tackle

Der Tackle-Cronk bietet schnellen Zugriff auf alle Informationen von Hosts und Services. "Batch commands" öffnet ein neues Fenster, in dem Sie mit zahlreichen Optionen die Anzahl der Objekte einschränken können, auf die ein bestimmter Befehl angewendet werden soll. Das Anklicken der Status-Buttons beschränkt die Anzeige der Hosts auf die aktiven Zustände.

Durch das Auswählen eines Objekts in der oberen Hälfte sehen Sie weitere Informationen in der unteren Hälfte. Der "Default"-Reiter zeigt die betreffenden Direktiven, ohne dass Sie dazu verschiedene Konfigurations- oder Statusdateien öffnen müssen.

Abbildung 6.15. Icinga-Web "Tackle"-Cronk

The screenshot shows the Icinga-Web interface with the 'HostStatus' tab selected. The main area displays a table of hosts with their status, last check time, and output logs. Below this, a detailed view for 'host_001' is shown, including state information, plugin output, and perfdata.

Host	Service health	SLA	Last check	Flags	Output
host_000			! 2011-11-26 11:05:32		host_000 (checked by pc3239f15) DOWN: random hostcheck
host_001			! 2011-11-26 11:12:52		host_001 (checked by pc3239f15) DOWN: down hostcheck
host_002			26 min, 40 sec. ago		host_002 (checked by pc3239f15) OK: ok hostcheck
host_003			! 0000-00-00 00:00:00		
host_004			19 min, 33 sec. ago		host_004 (checked by pc3239f15) OK: random hostcheck
host_005			50 sec. ago		host_005 (checked by pc3239f15) OK: random hostcheck
host_006			4 min, 34 sec. ago		host_006 (checked by pc3239f15) DOWN: up hostcheck
host_007			19 min, 28 sec. ago		host_007 (checked by pc3239f15) OK: ok hostcheck

Host_001 Configuration:

- State information:**

Name	Value
Action URL	(null)
Active checks enabled	1
Alias	down_001
Check attempt	1
Check type	0
Config type	1
Display name	host_001
Eventhandler enabled	1
Execution time	1.11801
- Plugin output:** host_001 (checked by pc3239f15) DOWN: down hostcheck
- Perfdata:** (empty)

Cronks und Views

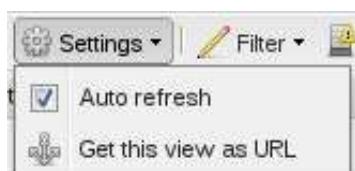
Icinga-Web erlaubt es, mehrere Cronks für verschiedene Anwendungsgebiete zu öffnen und zu verwalten. Damit können Sie Daten einsehen, Filter für unterschiedliche Views setzen oder Kommandos absenden. Die folgende Übersicht fasst die generellen Möglichkeiten zusammen (einige Cronks bieten zusätzliche Items, wie etwa das Senden von Kommandos).

Abbildung 6.16. Icinga-Web Cronk-Leiste



- Refresh
Manuelle Aktualisierung der Anzeige
- Settings
(De)aktivieren der automatischen Aktualisierung
Get this <item> by url

Abbildung 6.17. Icinga-Web Cronk-Leiste



- Filter

Modify/Remove

Commands

In Icinga-Web gibt es verschiedene Kommandos (siehe Kapitel "[External Commands](#)" für mehr Informationen), die an den Core geschickt werden können. Selektieren Sie die Einträge, die davon betroffen sein sollen, und dann das Kommando, das ausgeführt werden soll.

Host Commands

Abbildung 6.18. Icinga-Web Host-Befehle

	Process host check result
	Acknowledge host problem
	Acknowledge host problem with expiration time
	Remove host acknowledgement
	Schedule a host downtime
	Schedule a host downtime (with childs, triggered)
	Schedule a host downtime (with childs)
	Schedule a downtime for this host and its services
	Add a host comment
	Schedule next host check
	Schedule next forced host check
	Schedule a check of all services on this host
	Send custom host notification
	Enable active checks of this host
	Disable active checks of this host
	Start accepting passive checks for this host
	Stop accepting passive checks for this host
	Start obsessing over host
	Stop obsessing over host
	Enable notifications for this host
	Disable notifications for this host
	Enable notifications for all services on this host
	Disable notifications for all services on this host
	Enable checks of all services on this host
	Disable checks of all services on this host
	Enable event handler for this host
	Disable event handler for this host
	Enable flap detection for this host
	Disable flap detection for this host

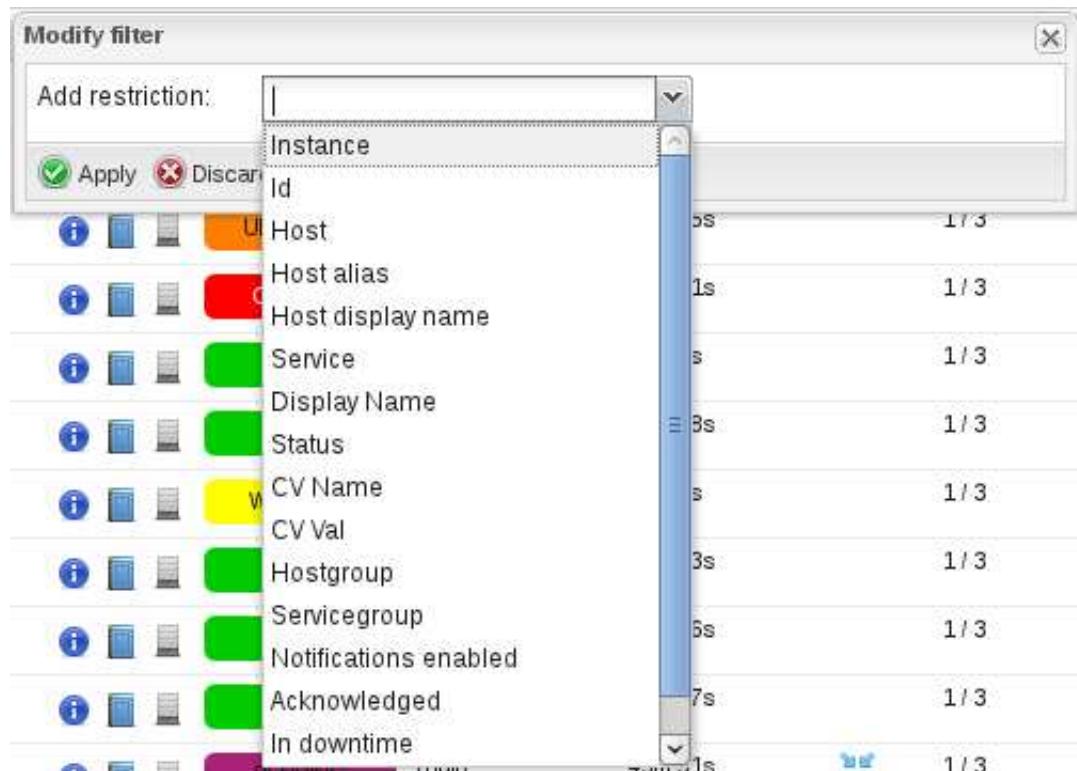
Service Commands

Abbildung 6.19. Icinga-Web Service-Befehle

	Schedule next service check
	Schedule forced service check
	Process service check result
	Acknowledge service check problem
	Acknowledge service check problem with expiration time
	Remove service acknowledgement
	Schedule a service downtime
	Add a service comment
	Send custom service notification
	Enable active checks of this service
	Disable active checks of this service
	Start accepting passive checks for this service
	Stop accepting passive checks for this service
	Start obsessing over this service
	Stop obsessing over this service
	Enable notifications for this service
	Disable notifications for this service
	Enable event handler for this service
	Disable event handler for this service
	Enable flap detection for this service
	Disable flap detection for this service

Filter

Filter erlauben Icinga-Web eigene, angepasste Ansichten (Views) die auch in eigenen persistenten Cronks gespeichert werden können. Als erstes wählen Sie bitte "Filter" und "Modify". Fügen Sie eine Restriktion aus dem Dropdown-Menü hinzu (dies ist von Cronk zu Cronk unterschiedlich). Wiederholen Sie diesen Schritt, um verschiedene Restriktionen zu setzen.

Abbildung 6.20. Icinga-Web Filtereinschränkung

Anschließend spezifizieren Sie die Bedingung(en), unter welchen der Filter die Anzeige generieren soll (contain, does not contain, is, is not). Fügen Sie zu jeder Bedingung einen Wert hinzu; Vorschläge werden eingeblendet, sobald Sie tippen.

Abbildung 6.21. Icinga-Web Filterbedingung

Wenden Sie den Filter auf den aktuellen Cronk an. Sofern benötigt, können Sie den Filter später modifizieren und weitere Restriktionen und Bedingungen hinzufügen/entfernen. Ein aktiver Filter wird rot markiert.

Abbildung 6.22. Icinga-Web Filter aktiv

Administration

Steigen Sie in die Administrationsübersicht ein, indem Sie "Admin" im oberen Menü auswählen. Danach wählen Sie "Users", "Groups", "Logs" oder "Tasks".

Abbildung 6.23. Icinga-Web Top-Menü Admin



Benutzer

Sie können Benutzer hinzufügen, löschen oder editieren.

Abbildung 6.24. Icinga-Web Benutzer-Administration

Available users					
<input type="button" value="New"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>	Page	1 of 1	<input type="button" value="First"/> <input type="button" value="Previous"/> <input type="button" value="Next"/> <input type="button" value="Last"/>	<input type="checkbox"/> Hide disabled	<input type="checkbox"/> Remove selected
ID	username	lastname	firstname	email	active
1	root	Root	Enoch	root@localhost.local	<input checked="" type="checkbox"/>
2	guest	Doe	John	john.dow@icinga.org	<input checked="" type="checkbox"/>
3	demo	user	demo	test@demo.de	<input checked="" type="checkbox"/>

Doppelklicken Sie einen Benutzer um ein neues Inlay Fenster zu öffnen, das Ihnen erlaubt, weitere Details zu spezifizieren. Dieselben Optionen stehen zur Verfügung, wenn Sie einen neuen Benutzer anlegen. Sie können die Größe des Fensters mittels Mauszeigerbewegung an den Ecken beeinflussen.

- General information
- Change password (and optional AuthKey for API)
- Permissions; z.B. zu welcher Gruppen zugehörig
- Principals für spezielle Rollen

Abbildung 6.25. Icinga-Web Benutzer editieren

Edit user

General information

User name:	guest		
Name:	John	Surname:	Doe
Email:	john.dow@icinga.org		
Disabled:	<input type="checkbox"/>		
Auth via:	internal		

Change Password

Password:	
Confirm password:	
Authkey for Api (optional):	

Meta information

Created:	2010-09-01 16:14:16
Modified:	2010-09-02 08:13:02

Permissions

Groups

appkit_admin (AppKit admin) :	<input type="checkbox"/>
appkit_user (Appkit user test) :	<input type="checkbox"/>
guest (Unauthorized Guest) :	<input type="checkbox"/>
icinga_user (The default representation of a ICINGA user) :	<input checked="" type="checkbox"/>

Principals

- + -**
- **principals**
 - **credential**
 - key icinga.demoMode

Save

Gruppen

Sie können Gruppen hinzufügen, löschen oder editieren. Die Gruppenvererbung lässt sich direkt im Gruppenbaum auf der rechten Seite anpassen.

Abbildung 6.26. Icinga-Web Gruppen-Administration

The screenshot shows the 'Available groups' section of the Icinga-Web interface. It lists four groups:

ID	groupname	description	isActive
3	appkit_admin	AppKit admin	<input checked="" type="checkbox"/>
2	appkit_user	Appkit user test	<input checked="" type="checkbox"/>
4	guest	Unauthorized Guest	<input checked="" type="checkbox"/>
1	icinga_user	The default representation of a ICINGA user	<input checked="" type="checkbox"/>

To the right, the 'Group inheritance' panel shows the hierarchy starting from 'Root':

- Root
 - appkit_user
 - guest
 - icinga_user

Doppelklicken Sie eine Gruppe, um ein neues Inlay-Fenster zu öffnen, das Ihnen erlaubt, weitere Details zu spezifizieren. Dieselben Optionen stehen zur Verfügung, wenn Sie eine neue Gruppe anlegen. Sie können die Größe des Fensters mittels Mauszeigerbewegung an den Ecken beeinflussen.

Abbildung 6.27. Icinga-Web Gruppen

The screenshot shows the 'Edit group' dialog for the 'icinga_user' group. The dialog is divided into several sections:

- General information:**
 - Group name: icinga_user
 - Description: The default representation of a ICINGA user
 - Disabled:
- Meta information:**
 - Created: 2010-09-01 18:07:29
 - Modified: 2010-09-01 18:07:29
- Permissions:**
 - Users:** A list of users assigned to this group, including demo (checked), root (checked), and guest (checked).
 - Principals:** A tree view of principal configurations:
 - principals
 - credential
 - icinga.user

A 'Save' button is located at the bottom right of the dialog.

Rollen

In der Benutzer- und Gruppenübersicht im Administrationsbereich können Sie Rollen hinzufügen/entfernen/editieren. Nachfolgend eine Liste der verfügbaren Rollen:

Abbildung 6.28. Icinga-Web Rollen

Select a principal (Press Ctrl for multiple selects)		
Principal	Description	Type
IcingaHostgroup	Limit data access to specific hostgroups	icinga
IcingaServicegroup	Limit data access to specific servicegroups	icinga
IcingaHostCustomVariablePair	Limit data access to specific custom variables	icinga
IcingaServiceCustomVariablePair	Limit data access to specific custom variables	icinga
IcingaContactgroup	Limit data access to users contact group membership	icinga
IcingaCommandRo	Limit access to commands	icinga
appkit.access	Access to login-page (which, actually, means no access)	credential
icinga.user	Access to icinga	credential
appkit.admin.groups	Access to group editor	credential
appkit.admin.users	Access to user editor	credential
appkit.admin	Access to admin panel	credential
appkit.user.dummy	Basic right for users	credential
appkit.api.access	Access to web-based api adapter	credential
icinga.demoMode	Hide features like password reset which are not wanted in demo syst	credential

Add selected principals | Clear selection

Logs

Hier können Sie verschiedene Logs betrachten, um diese als Hilfe bei Ihrer Fehlersuche zu verwenden.

Abbildung 6.29. Icinga-Web logs

Log		
Time	Message	Severity
No data to display		

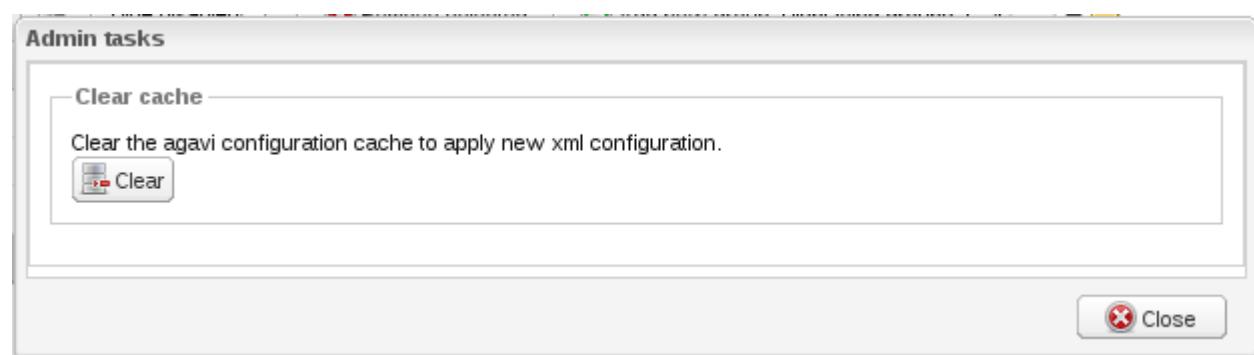
Available logs

- debug
- icinga-web

Task

Verschiedene Dinge erfordern das Löschen des Cache. Anstatt auf die Kommandozeile zu wechseln können Sie den Befehl über das "Tasks"-Menü ausführen.

Abbildung 6.30. Icinga-Web Tasks



[Zurück](#)

[Nach oben](#)

[Weiter](#)

Aktualisierung von Icinga-Web
und Icinga-Web Datenbank

[Zum Anfang](#)

Integration von PNP4Nagios in
das Icinga-Web Frontend

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Integration von PNP4Nagios in das Icinga-Web Frontend

[Zurück](#)

Kapitel 6. Die Benutzeroberflächen

[Weiter](#)

Integration von PNP4Nagios in das Icinga-Web Frontend

Mit Hilfe dieser Anleitung können Sie PNP4Nagios in das Icinga-Web-Frontend integrieren. Wenn Sie PNP4Nagios in das Icinga Classic UI integrieren möchten, lesen Sie bitte die PNP4Nagios [Dokumentation](#).

Abbildung 6.31. PNP4Nagios integriert in Icinga-Web

Instance	Service	Perfdata	Status	Last check	Info	Attempt	Output
default	SSH		OK	2010-09-01 16:06:39		1/4	SSH OK - OpenSSH_5.4 (protocol
default	Swap Usage		OK	2010-09-01 16:07:14		1/4	SWAP OK - 96% free (3779 MB o
default	Total Processes		OK	2010-09-01 16:02:52		1/4	PROCS OK: 203 processes with
default	Current Load		OK	2010-09-01 16:06:29		1/4	OK - load average: 0.55, 0.41, 0.
default	Current Users		OK	2010-09-01 16:04:07		1/4	USERS OK - 4 users currently lo
default	HTTP		WARNING	2010-09-01 16:04:44		4/4	HTTP WARNING: HTTP/1.1 403
default	PING		OK	2010-09-01 16:05:22		1/4	PING OK - Packet loss = 0%, RT
default	Root Partition		OK	2010-09-01 16:05:59		1/4	DISK OK - free space: / 21780 M

Installieren von PNP4Nagios

1. Installieren Sie PNP4Nagios wie in der PNP4Nagios-[Dokumentation](#) beschrieben.
2. Passen Sie die PNP4Nagios-Konfiguration auf Ihre Icinga-Konfiguration an. Wahrscheinlich müssen Sie hier ändern:

```
#> vi npcd.cfg
user = icinga
group = icinga
log_file = /var/log/icinga/npcd.log
perfdata_spool_dir = /var/icinga/spool/
perfdata_file = /var/icinga/perfdata.dump

#> vi process_perfdata.cfg
LOG_FILE = /var/log/icinga/perfdata.log

#> vi config.php
$conf['nagios_base'] = "/icinga/cgi-bin";
```

Erstellen Sie die Konfiguration, um die PNP4Nagios Host-Seiten in das Icinga-Web zu integrieren

1. Erstellen einer neuen Host-Grid-Ansicht:

Bitte erstellen Sie eine Kopie von icinga-host-template.xml in app/modules/Cronks/data/xml/grid unter Ihrem icinga-web Installationspfad:

```
#> cp /usr/local/icinga-web/app/modules/Cronks/data/xml/grid/icinga-host-template.xml \
/usr/local/icinga-web/app/modules/Cronks/data/xml/grid/icinga-my-host-template.xml
```

In der neuen Datei legen wir eine zusätzliche Feld- (field) Definition an:

```

<field name="pnp4nagios_host_link">
    <!-- datasource maps a data field from api call -->
    <datasource>
        <parameter name="field">HOST_NAME</parameter>
    </datasource>

    <display>
        <parameter name="visible">true</parameter>
        <parameter name="label">Perfdata</parameter>
        <parameter name="width">55</parameter>

        <parameter name="Ext.grid.Column">
            <parameter name="menuDisabled">true</parameter>
            <parameter name="fixed">true</parameter>
        </parameter>

        <parameter name="jsFunc">
            <!-- function to display column with icon in host status grid view -->
            <parameter>
                <parameter name="namespace">Cronk.grid.ColumnRenderer</parameter>
                <parameter name="function">columnImage</parameter>
                <parameter name="type">renderer</parameter>

                <parameter name="arguments">
                    <parameter name="image">images/icons/application_view_gallery.png</parameter>
                    <parameter name="css">x-icinga-grid-link</parameter>
                    <parameter name="attr">
                        <parameter name="qtip">Show host perftdata for this host</parameter>
                    </parameter>
                </parameter>
            </parameter>
        </display>

        <!-- create cell click event for the previously defined column -->
        <parameter>
            <parameter name="namespace">Cronk.grid.IcingaColumnRenderer</parameter>
            <parameter name="function">iFrameCronk</parameter>
            <parameter name="type">cellclick</parameter>
            <parameter name="arguments">
                <parameter name="title">Host perftdata for {host_name}</parameter>
                <parameter name="url"><![CDATA[ /pnp4nagios/index.php/graph?host={host_name}&srv=_HOST_ ]]></parameter>
                <parameter name="activateOnClick">true</parameter>
            </parameter>
        </parameter>
    </display>

    <filter>
        <parameter name="enabled">false</parameter>
    </filter>

    <order>
        <parameter name="enabled">false</parameter>
    </order>
</field>

```

2. Anlegen einer neuen Grid-Ansicht im "Data" Cronk-Container

Editieren von cronks.xml im Unterverzeichnis app/modules/Cronks/config/ unter Icinga-Web und hinzufügen von:

```

<cronk name="gridMyHostView">
    <ae:parameter name="module">Cronks</ae:parameter>
    <ae:parameter name="action">System.ViewProc</ae:parameter>
    <ae:parameter name="hide">false</ae:parameter>
    <ae:parameter name="description">Viewing host status in a grid including perftdata link</ae:parameter>
    <ae:parameter name="name">MyHostStatus</ae:parameter>
    <ae:parameter name="image">cronks.Stats</ae:parameter>
    <ae:parameter name="categories">data</ae:parameter>
    <ae:parameter name="ae:parameter">
        <ae:parameter name="template">icinga-my-host-template</ae:parameter>
    </ae:parameter>
</cronk>

```

Integrieren von PNP4Nagios in Icinga-Web-Serviceansichten

1. Erstellen einer neuen Service-Grid-Ansicht

Kopieren Sie das Standardtemplate "icinga-service-template.xml" von app/modules/Cronks/data/xml/grid in Ihren Icinga-Web- Installationspfad:

```
#> cp /usr/local/icinga-web/app/modules/Cronks/data/xml/grid/icinga-service-template.xml \
/usr/local/icinga-web/app/modules/Cronks/data/xml/grid/icinga-my-service-template.xml
```

In der neuen Datei legen wir eine zusätzliche Feld- (field) Definition an:

```

<field name="pnp4nagios_service_link">
    <!-- datasource maps a data field from api call -->
    <datasource>
        <parameter name="field">SERVICE_NAME</parameter>
    </datasource>

    <display>
        <parameter name="visible">true</parameter>
        <parameter name="label">Perfdata</parameter>
        <parameter name="width">55</parameter>

        <parameter name="Ext.grid.Column">
            <parameter name="menuDisabled">true</parameter>
            <parameter name="fixed">true</parameter>
        </parameter>

        <parameter name="jsFunc">
            <!-- function to display column with icon in host status grid view -->
            <parameter>
                <parameter name="namespace">Cronk.grid.ColumnRenderer</parameter>
                <parameter name="function">columnImage</parameter>
                <parameter name="type">renderer</parameter>
            </parameter>
            <parameter name="arguments">
                <parameter name="image">images/icons/application_view_gallery.png</parameter>
                <parameter name="css">x-icinga-grid-link</parameter>
                <parameter name="attr">
                    <parameter name="qtip">Show perfdata for this service</parameter>
                </parameter>
            </parameter>
        </parameter>

        <!-- create cell click event for the previously defined column -->
        <parameter>
            <parameter name="namespace">Cronk.grid.IcingaColumnRenderer</parameter>
            <parameter name="function">iFrameCronk</parameter>
            <parameter name="type">cellclick</parameter>
            <parameter name="arguments">
                <parameter name="title">Service perfdata for {service_name} on {host_name}</parameter>
                <parameter name="url"><![CDATA[/pnp4nagios/index.php/graph?host={host_name}&srv={service_name}]]></parameter>
                <parameter name="activateOnClick">true</parameter>
            </parameter>
        </parameter>
    </display>

    <filter>
        <parameter name="enabled">false</parameter>
    </filter>

    <order>
        <parameter name="enabled">false</parameter>
    </order>
</field>

```

2. Anlegen einer neuen Grid-Ansicht im "Data" Cronk-Container

Editieren von cronks.xml im Unterverzeichnis app/modules/Cronks/config/ unter Icinga-Web und hinzufügen von:

```

<cronk name="gridMyServiceView">
    <ae:parameter name="module">Cronks</ae:parameter>
    <ae:parameter name="action">System.ViewProc</ae:parameter>
    <ae:parameter name="hide">false</ae:parameter>
    <ae:parameter name="description">Viewing service status in a grid including perfdata link</ae:parameter>
    <ae:parameter name="name">MyServiceStatus</ae:parameter>
    <ae:parameter name="image">cronks.Stats2</ae:parameter>
    <ae:parameter name="categories">data</ae:parameter>
    <ae:parameter name="ae:parameter">
        <ae:parameter name="template">icinga-my-service-template</ae:parameter>
    </ae:parameter>
</cronk>

```

3. Verwenden der neuen Grid-Ansicht als Standard-Serviceansicht

Bitte sichern Sie zuerst Ihre originale Ansicht:

```
#> cp data/xml/grid/icinga-service-template.xml data/xml/grid/icinga-service-template.bak
```

dann

```
#> cp data/xml/grid/icinga-my-service-template.xml data/xml/grid/icinga-service-template.xml
```

Leeren Sie den Cache wie unten beschrieben. Die Performancegraphen sind nun in Ihren "serviceStatus"-Cronk integriert!



Anmerkung

Bitte denken Sie daran, wenn Sie eine *.xml-Datei editieren, müssen Sie danach den Cache bereinigen!

```
#> rm -f app/cache/config/*.php  
oder /path/to/clearcache.sh  
#> /usr/local/icinga-web/bin/clearcache.sh
```

Das war es auch schon, Sie sind fertig!

[Zurück](#)[Nach oben](#)[Weiter](#)[Einführung in Icinga-Web](#)[Zum Anfang](#)[Kapitel 7. Fortgeschrittene Themen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Kapitel 7. Fortgeschrittene Themen

[Zurück](#)

[Weiter](#)

Kapitel 7. Fortgeschrittene Themen

Inhaltsverzeichnis

- [Externe Befehle](#)
 - [Eventhandler](#)
 - [sprunghafte Services](#)
 - [Service- und Host-Frische-Prüfungen](#)
 - [Verteilte Überwachung](#)
 - [Redundante und Failover-Netzwerk-Überwachung](#)
 - [Erkennung und Behandlung von Status-Flattern](#)
 - [Benachrichtigungsescalationen](#)
 - [Eskalations-Bedingung](#)
 - [Bereitschafts-Rotation](#)
 - [Service- und Host-Gruppen überwachen](#)
 - [Host- und Service-Abhängigkeiten](#)
 - [Status Stalking](#)
 - [Performance-Daten](#)
 - [Geplante Ausfallzeiten](#)
 - [Benutzen des Embedded Perl Interpreters](#)
 - [Adaptive Überwachung](#)
 - [Vorausschauende Abhängigkeitsprüfungen](#)
 - [Zwischengespeicherte Prüfungen](#)
 - [Passive Host-Zustandsübersetzung](#)
 - [Service- und Host-Prüfungsplanung](#)
 - [Angepasste CGI-Kopf- und Fußzeilen](#)
 - [Objektvererbung](#)
 - [Zeitsparende Tricks für Objektdefinitionen](#)
-

[Zurück](#)

[Weiter](#)

Integration von PNP4Nagios in
das Icinga-Web Frontend

[Zum Anfang](#)

[Externe Befehle](#)



Externe Befehle

[Zurück](#)

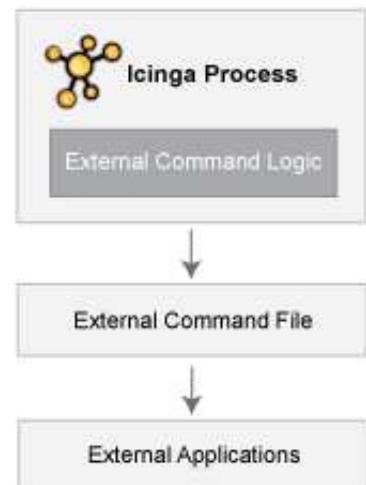
Kapitel 7. Fortgeschrittene Themen

[Weiter](#)

Externe Befehle

Einführung

Icinga kann Befehle aus externen Applikationen verarbeiten (einschließlich der CGIs) und verschiedene Aspekte seiner Überwachungsfunktionen aufgrund der Befehle verändern, die es erhält. Externe Applikationen können Befehle "einreichen", indem sie in das [command file](#) schreiben, das regelmäßig vom Icinga-Daemon verarbeitet wird.



Externe Befehle aktivieren

Damit Icinga externe Befehle verarbeitet, müssen Sie folgendes tun:

- aktivieren Sie die Prüfung auf externe Befehle mit der [check_external_commands](#)-Option.
- setzen Sie die Wiederholrate von Befehlsprüfungen mit der [command_check_interval](#)-Option.
- definieren Sie den Ort des Command-File mit der [command_file](#)-Option.
- setzen Sie korrekte Berechtigungen für das Verzeichnis, welches das External-Command-File enthält, wie in der [Schnellstartanleitung](#) beschrieben.

Wann prüft Icinga auf externe Befehle?

- in regelmäßigen Intervallen, wie sie durch die Option `command_check_interval` in der Hauptkonfigurationsdatei angegeben sind
- direkt nachdem `Eventhandler` ausgeführt werden. Das passiert zusätzlich zum regelmäßigen Zyklus von externen Befehlsprüfungen und wird getan, um unverzügliche Aktivitäten zu ermöglichen, falls ein Eventhandler Befehle an Icinga schickt.

Externe Befehle benutzen

Externe Befehle können benutzt werden, um eine Reihe von Dingen zu erreichen, während Icinga läuft. Beispiele dafür, was getan werden kann, umfassen u.a. vorübergehend Benachrichtigungen für Services und Hosts zu deaktivieren, vorübergehend Service-Prüfungen zu deaktivieren, sofortige Service-Prüfungen zu erzwingen, Kommentare für Hosts und Services hinzuzufügen usw.

Befehlsformat

Externe Befehle, die in das `command file` geschrieben werden, haben das folgende Format...

[*Zeit*] *Befehls-ID*; *Befehlsargumente*

...wobei *Zeit* die Zeit (im *time_t*-Format) ist, zu der die externe Applikation den externen Befehl an das Command-File geschickt hat. Die Werte für die *Befehls-ID* und die *Befehlsargumente* hängen davon ab, welcher Befehl an Icinga geschickt wird.

Eine komplette Liste der Befehle, die benutzt werden können, finden Sie in der [Liste der externen Befehle](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Kapitel 7. Fortgeschrittene Themen

[Zum Anfang](#)

Eventhandler

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Eventhandler

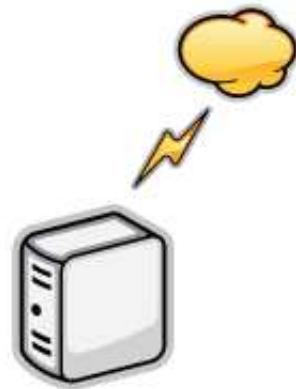
[Zurück](#)

Kapitel 7. Fortgeschrittene Themen

[Weiter](#)

Eventhandler

Einführung



Eventhandler sind optionale Systemkommandos (Scripts oder Programme), die gestartet werden, wenn ein Host- oder Service-Zustandswechsel stattfindet. Sie werden auf dem System ausgeführt, auf dem die Prüfung eingeplant (initiiert) wurde.

Ein einleuchtender Einsatz von Eventhandlern ist die Möglichkeit von Icinga, proaktiv Probleme zu beheben, bevor jemand benachrichtigt wird. Einige andere Anwendungsmöglichkeiten für Eventhandler umfassen:

- neustarten eines ausgefallenen Service
- anlegen eines Trouble-Tickets in einem Helpdesk-Systems
- eintragen von Ereignisinformationen in eine Datenbank
- Strom aus- und einschalten bei einem Host*
- etc.

* Strom durch ein automatisiertes Script bei einem Host aus- und einzuschalten, der Probleme hat, sollte wohlüberlegt sein. Betrachten Sie sorgfältig die möglichen Konsequenzen, bevor Sie automatische Reboots implementieren. :-)

Wann werden Eventhandler ausgeführt?

Eventhandler werden ausgeführt, wenn ein Service oder Host

- in einem SOFT-Problemzustand ist
- in einen HARD-Problemzustand wechselt
- aus einem SOFT- oder HARD-Problemzustand zurückkehrt

SOFT- und HARD-Zustände sind ausführlich [hier](#) beschrieben.

Eventhandler-Typen

Es gibt unterschiedliche Typen von optionalen Eventhandlern, die Sie definieren können, um Host- und Statuswechsel zu behandeln:

- Globale Host-Eventhandler
- Globale Service-Eventhandler
- Host-spezifische Eventhandler
- Service-spezifische Eventhandler

Globale Host- und Service-Eventhandler werden für *jeden* auftretenden Host- oder Service-Zustandswechsel durchgeführt, direkt vor einem möglichen Host- oder Service-spezifischen Eventhandler. Sie können globale Host- oder Service-spezifische Eventhandler durch die [global_host_event_handler](#) und [global_service_event_handler](#)-Optionen in der Hauptkonfigurationsdatei angeben.

Einzelne Hosts und Service können ihre eigenen Eventhandler haben, die ausgeführt werden, um Statuswechsel zu behandeln. Sie können einen auszuführenden Eventhandler durch die [event_handler](#)-Direktive in Ihren [Host](#)- oder [Service](#)-Definitionen angeben. Diese Host- und Service-spezifischen Eventhandler werden direkt nach dem (optionalen) globalen Host- oder Service-Eventhandler ausgeführt.

Eventhandler aktivieren

Eventhandler können durch die [enable_event_handlers](#)-Direktive in Ihrer Hauptkonfigurationsdatei programmweit aktiviert oder deaktiviert werden.

Host- und Service-spezifische Eventhandler werden durch die [event_handler_enabled](#)-Direktive in Ihrer [Host](#)- oder [Service](#)-Definition aktiviert oder deaktiviert. Host- und Service-spezifische Eventhandler werden nicht ausgeführt, wenn die globale [enable_event_handlers](#)-Option deaktiviert ist.

Eventhandler-Ausführungsreihenfolge

Wie bereits erwähnt werden globale Host- und Service-Eventhandler direkt vor Host- oder Service-spezifischen Eventhandlern ausgeführt.

Eventhandler werden bei HARD-Problemen und Erholungszuständen direkt nach dem Versand von Benachrichtigungen ausgeführt.

Eventhandler-Kommandos schreiben

Eventhandler werden wahrscheinlich Shell- oder Perl-Scripte sein, aber es ist jede Art von ausführbarer Datei denkbar, die von der Kommandozeile aus lauffähig ist. Die Scripte sollten mindestens die folgenden **Makros** als Argumente nutzen:

Für Services: **`$SERVICESTATE$`**, **`$SERVICESTATETYPE$`**, **`$SERVICEATTEMPT$`**

Für Hosts: **`$HOSTSTATE$`**, **`$HOSTSTATETYPE$`**, **`$HOSTATTEMPT$`**

Die Scripte sollten die Werte der übergebenen Parameter untersuchen und darauf basierend notwendige Aktionen ausführen. Der beste Weg, die Funktionsweise von Eventhandlern zu verstehen, ist der Blick auf ein Beispiel. Glücklicherweise finden Sie eins [hier](#).



Hinweis: Zusätzliche Eventhandler-Scripte finden Sie im *contrib/eventhandlers/-Unterverzeichnis* der Icinga-Distribution. Einige dieser Beispiel-Scripts demonstrieren die Benutzung von [externen Befehlen](#), um [redundante](#) und [verteilte](#) Überwachungsumgebungen zu implementieren.

Berechtigungen für Eventhandler-Befehle

Eventhandler werden normalerweise mit den gleichen Berechtigungen ausgeführt wie der Benutzer, der Icinga auf Ihrer Maschine ausführt. Dies kann ein Problem darstellen, wenn Sie einen Eventhandler schreiben möchten, der Systemdienste neu startet, da generell root-Rechte benötigt werden, um diese Aufgaben zu erledigen.

Idealerweise sollten Sie den Typ von Eventhandler einschätzen und dem Icinga-Benutzer gerade genug Berechtigungen gewähren, damit er die notwendigen Systembefehle ausführen kann. Vielleicht möchten Sie [sudo](#) ausprobieren, um das zu erreichen.

Service-Eventhandler-Beispiel

Das folgende Beispiel geht davon aus, dass Sie den HTTP-Server auf der lokalen Maschine überwachen und *restart-htpd* als den Eventhandler-Befehl für die HTTP-Service-Definition angegeben haben. Außerdem nehmen wir an, dass Sie die Option *max_check_attempts* für den Service auf einen Wert von 4 oder höher gesetzt haben (d.h., der Service wird viermal geprüft, bevor angenommen wird, dass es ein richtiges Problem gibt). Eine gekürzte Service-Definition könnte wie folgt aussehen...

```
define service{
    host_name           somehost
    service_description HTTP
    max_check_attempts 4
    event_handler       restart-htpd
    ...
}
```

Sobald der Service mit einem Eventhandler definiert wird, müssen wir diesen Eventhandler als Befehlsfolge definieren. Eine Beispieldefinition für *restart-htpd* sehen Sie nachfolgend. Beachten Sie die Makros in der Kommandozeile, die an das Eventhandler-Script übergeben werden - sie sind wichtig!

```
define command{
    command_name      restart-htpd
    command_line      /usr/local/icinga/libexec/eventhandlers/restart-htpd  $SERVICESTATE$ $SERVICESTATETYPE$ $SERVICEATTEMPT$
}
```

Lassen Sie uns nun das Eventhandler-Script schreiben (das ist das */usr/local/icinga/libexec/eventhandlers/restart-htpd*-Script).

```

#!/bin/sh
#
# Eventhandler-Script für den Restart des Web-Servers auf der lokalen Maschine
#
# Anmerkung: Dieses Script wird den Web-Server nur dann restarten, wenn der Service
#           dreimal erneut geprüft wurde (sich in einem "soft"-Zustand befindet)
#           oder der Web-Service aus irgendeinem Grund in einen "hard"-Zustand fällt
# In welchem Status befindet sich der Service?
case "$1" in
OK)
    # Der Service hat sich gerade erholt, also tun wir nichts...
;;
WARNING)
    # Wir kümmern uns nicht um WARNING-Zustände, denn der Dienst läuft wahrscheinlich noch...
;;
UNKNOWN)
    # Wir wissen nicht, was einen UNKNOWN-Fehler auslösen könnte, also tun wir nichts...
;;
CRITICAL)
    # Aha! Der HTTP-Service scheint ein Problem zu haben - vielleicht sollten wir den Server neu starten...
# Ist dies ein "Soft"- oder ein "Hard"-Zustand?
case "$2" in
# Wir sind in einem "Soft"-Zustand, also ist Icinga mitten in erneuten Prüfungen, bevor es in einen
# "Hard"-Zustand wechselt und Kontakte informiert werden...
SOFT)
    # Bei welchem Versuch sind wir? Wir wollen den Web-Server nicht gleich beim ersten Mal restarten,
# denn es könnte ein Ausrutscher sein!
case "$3" in
# Warte, bis die Prüfung dreimal wiederholt wurde, bevor der Web-Server restartet wird.
# Falls der Check ein viertes Mal fehlschlägt (nachdem wir den Web-Server restartet haben),
# wird der Zustandtyp auf "Hard" wechseln und Kontakte werden über das Problem informiert.
# Hoffentlich wird der Web-Server erfolgreich restartet, so dass der vierte Check zu einer
# "Soft"-Erholung führt. Wenn das passiert, wird niemand informiert, weil wir das Problem gelöst haben.
3)
    echo -n "Restart des HTTP-Service (dritter kritischer \"Soft\"-Zustand)..."
# Aufrufen des Init-Scripts, um den HTTPD-Server zu restarten
/etc/rc.d/init.d/httpd restart
;;
esac
;;
# Der HTTP-Service hat es irgendwie geschafft, in einen "Hard"-Zustand zu wechseln, ohne dass das Problem
# behoben wurde. Er hätte durch den Code restartet werden sollen, aber aus irgendeinem Grund hat es nicht
# funktioniert. Wir probieren es ein letztes Mal, okay?
# Anmerkung: Kontakte wurden bereits darüber informiert, dass es ein Problem mit dem Service gibt (solange
# Sie nicht Benachrichtigungen für diesen Service deaktiviert haben.
HARD)
    echo -n "Restart des HTTP-Service..."
# Aufrufen des Init-Scripts, um den HTTPD-Server zu restarten
/etc/rc.d/init.d/httpd restart
;;
esac
;;
esac
;;
exit 0

```

Das mitgelieferte Beispiel-Script wird versuchen, den Web-Server auf der lokalen Maschine in zwei Fällen zu restarten:

- nachdem der Service das dritte Mal erneut geprüft wurde und sich in einem kritischen "Soft"-Zustand befindet
- nachdem der Service das erste Mal in einen kritischen "Hard"-Zustand wechselt

Das Script sollte theoretisch den Web-Server restarten und das Problem beheben, bevor der Service in einen "Hard"-Problemzustand wechselt, aber wir stellen eine Absicherung bereit, falls es nicht das erste Mal funktioniert. Es ist anzumerken, dass der Eventhandler nur einmal ausgeführt wird, wenn der Service in einen HARD-Zustand wechselt. Das hält Icinga davon ab, das Script zum Restart des Web-Servers wiederholt auszuführen, wenn der Service in einem HARD-Problemzustand bleibt. Das wollen Sie nicht. :-)

Das ist alles! Eventhandler sind ziemlich einfach zu schreiben und zu implementieren, also versuchen Sie es und sehen, was Sie tun können.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Externe Befehle](#)

[Zum Anfang](#)

[sprunghafte Services](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



sprunghafte Services

[Zurück](#)

Kapitel 7. Fortgeschrittene Themen

[Weiter](#)

sprunghafte Services

Einführung

Icinga hat die Möglichkeit, zwischen "normalen" und "flüchtigen" Services zu unterscheiden. Die *is_volatile*-Option in jeder Service-Definition erlaubt Ihnen festzulegen, ob ein bestimmter Service flüchtig ist oder nicht. Für die meisten Leute wird die Mehrzahl der überwachten Services nicht-flüchtig (d.h. "normal") sein. Trotzdem können flüchtige Services sehr nützlich sein, wenn sie richtig eingesetzt werden...

Wofür sind sie nützlich?

Flüchtige Services sind nützlich zur Überwachung von...

- Dingen, die sich jedes Mal automatisch in einen "OK"-Zustand zurücksetzen, wenn sie geprüft werden
- Ereignisse wie Sicherheits-Alarme, die jedes Mal Beachtung erfordern, wenn ein Problem vorliegt (und nicht nur beim ersten Mal)

Was ist so besonders an flüchtigen Services?

Flüchtige Services unterscheiden sich von "normalen" Services in drei wichtigen Punkten. *Jedes Mal* wenn sie in einem [harten](#) nicht-OK-Zustand sind und die Prüfung einen nicht-OK-Zustand ergibt (also keine Statusänderung eintritt)...

- wird der nicht-OK-Zustand des Service protokolliert
- werden Kontakte über das Problem informiert (falls es das ist, [was zu tun ist](#)). Anmerkung: Benachrichtigungsintervalle werden bei flüchtigen Services ignoriert.
- Der [Eventhandler](#) für den Service wird ausgeführt (falls einer definiert ist)

Diese Ereignisse finden normalerweise nur für Services statt, wenn sie in einem nicht-OK-Zustand sind und gerade ein Hard-Zustandswechsel erfolgte. In anderen Worten, sie passieren nur das erste Mal, wenn ein Service in einen nicht-OK-Zustand geht. Wenn weitere Prüfungen des Service den gleichen nicht-OK-Zustand ergeben, erfolgt kein harter Zustandswechsel und keines der genannten Ereignisse wird stattfinden.



Hinweis: Wenn Sie nur an der Protokollierung interessiert sind, dann sehen Sie sich die [Stalking](#)-Option an.

Die Macht der Zwei

Wenn Sie die Möglichkeiten von flüchtigen Services und [passiven Service-Prüfungen](#) kombinieren, können Sie einige sehr nützliche Dinge tun. Beispiele hierfür umfassen u.a. die Behandlung von SNMP-Traps, Sicherheits-Alarme, usw.

Wie wäre es mit einem Beispiel... Nehmen wir an, Sie nutzen [PortSentry](#), um Portscans auf Ihrer Maschine zu erkennen und automatisch potenzielle Eindringlinge auszusperren. Wenn Sie wollen, dass Icinga über Portscans erfährt, können Sie das Folgende tun...

Icinga Konfiguration:

- Legen Sie eine Service-Definition namens *Port Scans* an und verbinden Sie diese mit dem Host, auf dem PortSentry läuft.
- Setzen Sie die *max_check_attempts*-Direktive in der Service-Definition auf 1. Dies teilt Icinga mit, sofort einen [Hard-Zustand](#) für den Service zu erzwingen, wenn ein nicht-OK-Zustand ermittelt wird.
- Setzen Sie die *active_checks_enabled*-Direktive in der Service-Definition auf 0. Dies hält Icinga davon ab, den Service aktiv zu prüfen.
- Setzen Sie die *passive_checks_enabled*-Direktive in der Service-Definition auf 1. Das erlaubt passive Prüfungen für den Service.
- Setzen Sie die *is_volatile*-Direktive in der Service-Definition auf 1.

PortSentry Konfiguration:

Editieren Sie die PortSentry-Konfigurationsdatei (`portsentry.conf`) und definieren Sie einen Befehl für die *KILL_RUN_CMD*-Direktive wie folgt:

```
KILL_RUN_CMD="/usr/local/icinga/libexec/eventhandlers/submit_check_result host_name 'Port Scans' 2 'Port scan from host $TARGET$ on port $PORT$. Host has been firewalled.'"
```

Stellen Sie sicher, *host_name* durch den Kurznamen des Hosts zu ersetzen, mit dem der Service verbunden ist.

Portscan-Script:

Erstellen Sie ein Shell-Script im `/usr/local/icinga/libexec/eventhandlers`-Verzeichnis namens *submit_check_result*. Der Inhalt des Shell-Scripts sollte ähnlich dem Folgenden sein...

```
#!/bin/sh
# Write a command to the Icinga command file to cause
# it to process a service check result
echocmd="/bin/echo"
CommandFile="/usr/local/icinga/var/rw/nagios.cmd"
# get the current date/time in seconds since UNIX epoch
datetime=`date +%s`
# create the command line to add to the command file
cmdline="[$datetime] PROCESS_SERVICE_CHECK_RESULT;$1;$2;$3;$4"
# append the command to the end of the command file
'$echocmd $cmdline >> $CommandFile'
```

Was passiert, wenn PortSentry in der Zukunft einen Portscan auf der Maschine entdeckt?

- PortSentry wird den Host ausschließen ("firewall", das ist eine Funktion der PortSentry-Software)

- PortSentry wird das *submit_check_result*-Shell-Script ausführen und ein passives Prüfergebnis an Icinga senden
- Icinga wird das external command file lesen und das passive Service-Prüfergebnis von PortSentry verarbeiten
- Icinga wird den *Port Scans*-Service in einen harten CRITICAL-Zustand versetzen und Benachrichtigungen an die Kontakte senden

Ziemlich hübsch, oder?

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Eventhandler

[Zum Anfang](#)

Service- und
Host-Frische-Prüfungen

© 2009-2011 Icinga Development Team, <http://www.icinga.org>

**Service- und Host-Frische-Prüfungen**[Zurück](#)[Kapitel 7. Fortgeschrittene Themen](#)[Weiter](#)

Service- und Host-Frische-Prüfungen

Einführung

Icinga unterstützt ein Feature, das die "Frische" (Freshness) der Host- und Service-Prüfungen überprüft. Der Zweck der Frische-Prüfung ist es, bei passiven Host- und Service-Prüfungen sicherzustellen, dass diese regelmäßig von externen Applikationen zur Verfügung gestellt werden.

Frische-Prüfungen sind sinnvoll, wenn Sie sicherstellen wollen, dass [passive Prüfungen](#) so regelmäßig empfangen werden wie Sie das erwarten. Das kann in [verteilten](#) und [Failover](#) Überwachungsumgebungen sehr sinnvoll sein.



Wie funktioniert die Frische-Prüfung?

Icinga prüft periodisch die Frische der Ergebnisse für alle Hosts und Services, bei denen Frische-Prüfungen aktiviert sind.

- ein Frische-Schwellwert wird für jeden Host oder Service berechnet.
- für jeden Host/Service wird das Alter des letzten Prüfungsergebnisses mit dem Frische-Schwellwert verglichen.
- wenn das Alter des letzten Prüfungsergebnisses größer als der Frisch-Schwellwert ist, wird das Prüfergebnis als "abgestanden" (stale) betrachtet.

- wenn das Prüfergebnis als abgestanden angesehen wird, wird Icinga eine [aktive Prüfung](#) für den Host oder Service mit dem Kommando ausführen, das in der Host- oder Service-Definition angegeben ist.



Hinweis: Eine aktive Prüfung wird ausgeführt, selbst wenn aktive Prüfungen programmweit oder auf Host- bzw. Service-spezifischer Basis deaktiviert sind.

Wenn Sie beispielsweise einen Frische-Schwellwert von 60 für einen Ihrer Services haben, wird Icinga diesen Service als abgestanden ansehen, wenn das letzte Prüfergebnis älter als 60 Sekunden ist.

Frische-Prüfungen aktivieren

Was Sie tun müssen, um Frische-Prüfungen zu aktivieren...

- aktivieren Sie Frische-Prüfungen auf programmweiter Basis mit den [check_service_freshness](#) und [check_host_freshness](#)-Direktiven.
- benutzen Sie die [service_freshness_check_interval](#)- und [host_freshness_check_interval](#)-Optionen, um Icinga mitzuteilen, wie oft es die Frische von Host- und Service-Ergebnissen prüfen soll.
- aktivieren Sie Frische-Prüfungen auf Host- und Service-spezifischer Basis, indem Sie die [check_freshness](#)-Option in Ihrer Host- und Service-Definitionen auf 1 setzen.
- konfigurieren Sie Frische-Schwellwerte, indem Sie die [freshness_threshold](#)-Option in Ihren Host- und Service-Definitionen setzen.
- konfigurieren Sie die [check_command](#)-Option in Ihren Host- oder Service-Definitionen, so dass sie ein gültiges Script enthalten, das benutzt werden kann, um den Host oder Service aktiv zu prüfen, wenn er als abgestanden angesehen wird.
- Die [check_period](#)-Option in Ihren Host- und Service-Definitionen wird benutzt, wenn Icinga festlegt, wann ein Host oder Service auf Frische geprüft werden soll, um sicherzustellen, dass es sich um ein gültiges Zeitfenster handelt.



Hinweis: Wenn Sie keinen Host- oder Service-spezifischen [freshness_threshold](#)-Wert angeben (oder ihn auf Null setzen), wird Icinga automatisch einen Schwellwert berechnen, der darauf basiert, wie oft Sie den jeweiligen Host- oder Service überwachen. Wir würden empfehlen, dass Sie explizit einen Frische-Schwellwert angeben, statt dass Icinga einen für Sie auswählt.

Beispiel

Ein Beispiel für einen Service, der eine Frische-Prüfung benötigen könnte, wäre einer, der den Status Ihrer nächtlichen Backups meldet. Vielleicht haben Sie ein externes Script, welches das Ergebnis des Backup-Jobs an Icinga meldet, sobald das Backup beendet ist. In diesem Fall werden alle Prüfungen/Ergebnisse für diesen Service durch eine externe Applikation mit Hilfe von passiven Prüfungen zur Verfügung gestellt. Um sicherzustellen, dass der Status des Backup-Jobs täglich gemeldet wird, können Sie die Frische-Prüfung für diesen Service aktivieren. Falls das externe Script das Ergebnis des Backup-Jobs nicht meldet, kann Icinga ein kritisches Ergebnis imitieren, indem man folgendes tut...

Nachfolgend, wie die Definition für den Service aussehen könnte (einige benötigte Optionen fehlen...)

```
define service{
    host_name          backup-server
    service_description ArcServe Backup Job
    active_checks_enabled 0           ; aktive Prüfungen sind NICHT aktiviert
    passive_checks_enabled 1          ; passive Prüfungen sind aktiviert (dadurch werden Ergebnisse gemeldet)
    check_freshness     1
    freshness_threshold 93600        ; 26 Stunden Schwellwert, nachdem Backups nicht immer zur gleichen Zeit beendet sind
    check_command       no-backup-report ; dieses Kommando wird nur ausgeführt, wenn der Service als "abgestanden" angesehen wird
    ...andere Optionen...
}
```

Beachten Sie, dass aktive Prüfungen für den Service deaktiviert sind. Das ist so, weil die Ergebnisse für den Service nur durch eine externe Applikation geliefert werden. Die Frische-Prüfung ist aktiviert und der Frische-Schwellwert ist auf 26 Stunden gesetzt. Das ist ein bisschen mehr als 24 Stunden, weil Backup-Jobs ab und zu länger dauern (abhängig davon, wie viele Daten zu sichern sind, wie viel Netzwerkverkehr herrscht, usw.). Das *no-backup-report*-Kommando wird nur ausgeführt, wenn die Ergebnisse des Service als abgestanden angesehen werden. Die Definition des *no-backup-report*-Kommandos könnte wie folgt aussehen...

```
define command{
    command_name      no-backup-report
    command_line      /usr/local/icinga/libexec/check_dummy 2 "CRITICAL: Results of backup job were not reported!"
```

Falls Icinga erkennt, dass das Service-Ergebnis abgestanden ist, wird es das *no-backup-report*-Kommando als eine aktive Service-Prüfung ausführen. Das führt dazu, dass das *check_dummy*-Plugin ausgeführt wird, das einen kritischen Status an Icinga meldet. Der Service wird dann in einen kritischen Zustand gehen (falls das nicht bereits der Fall ist) und wahrscheinlich wird jemand über das Problem informiert.

[Zurück](#)
[Nach oben](#)
[Weiter](#)
[sprunghafte Services](#)
[Zum Anfang](#)
[Verteilte Überwachung](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Verteilte Überwachung

[Zurück](#)[Kapitel 7. Fortgeschrittene Themen](#)[Weiter](#)

Verteilte Überwachung

Einführung

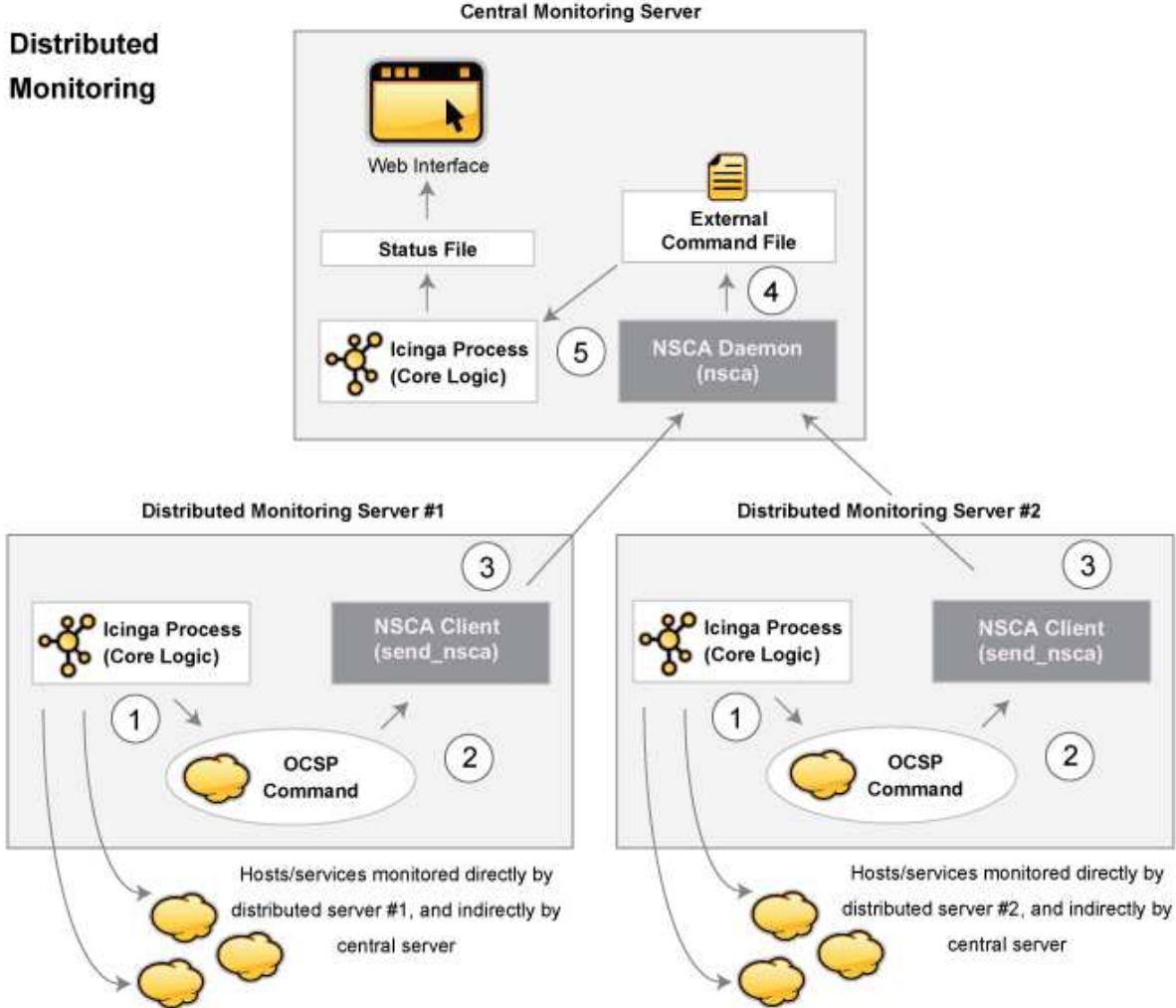
Icinga kann konfiguriert werden, so dass es verteilte Überwachung von Netzwerk-Services und Ressourcen unterstützt. Wir werden versuchen, kurz zu beschreiben, wie das erreicht werden kann...

Ziele

Das Ziel der verteilten Überwachungsumgebung, das wir beschreiben wollen, ist die Reduzierung des Overheads (CPU-Belastung, etc.) bei der Service-Prüfung von einem "zentralen" Server auf ein oder mehrere "verteilte" Server. Die meisten kleinen bis mittleren Unternehmen werden keinen wirklichen Bedarf für das Aufsetzen solch einer Umgebung haben. Wenn Sie allerdings hunderte oder sogar tausende von *Hosts* (und ein Mehrfaches an Services) mit Icinga überwachen wollen, dann kann das ziemlich wichtig werden.

Referenzdiagramm

Das folgende Diagramm soll Ihnen eine generelle Idee davon geben, wie verteilte Überwachung mit Icinga arbeitet. Wir werden uns auf die Elemente im Diagramm beziehen, während wir die Dinge erklären...



Zentraler Server vs. Verteilte Server

Beim Einrichten einer verteilten Überwachungsumgebung mit Icinga gibt es Unterschiede in der Art, wie zentrale und verteilte Server konfiguriert sind. Wir werden Ihnen zeigen, wie beide Arten von Servern konfiguriert werden und erklären, welche Auswirkungen die gemachten Änderungen auf die gesamte Überwachung haben. Für den Anfang beschreiben wir den Zweck der verschiedenen Server-Typen...

Die Funktion eines *verteilten Servers* ist es, aktiv Prüfungen für alle Services durchzuführen, die Sie für eine "Gruppe" (Cluster) von Hosts definieren. Wir benutzen den Begriff "Gruppe" locker - er meint lediglich eine willkürliche Gruppe von Hosts in Ihrem Netzwerk. Abhängig von Ihrem Netzwerk-Layout können Sie mehrere Gruppen in einem physischen Standort haben oder jede Gruppe kann durch ein WAN voneinander getrennt sein, mit einer eigenen Firewall, usw. Wichtig anzumerken ist, dass es für jede Gruppe von Hosts (wie immer Sie diese definieren mögen) einen verteilten Server gibt, auf dem Icinga läuft, und der die Services der Hosts dieser Gruppe überwacht. Ein verteilter Server enthält meistens eine simple Installation von Icinga. Es muss kein Web-Interface installiert sein, keine Benachrichtigungen versenden, keine Eventhandler-Scripts ausführen, noch etwas anderes tun außer Service-Prüfungen ausführen, wenn Sie das nicht wollen. Detaillierte Informationen zur Konfiguration eines verteilten Services gibt es später...

Der Zweck des *zentralen* Servers ist es lediglich, auf Service-Prüfungsergebnisse von einem oder mehreren verteilten Servern zu horchen. Obwohl Services ab und zu aktiv durch den zentralen Server geprüft werden, werden diese aktiven Prüfungen nur unter schlimmen Umständen ausgeführt, also lassen Sie uns im Moment sagen, dass der zentrale Server lediglich passive Prüfungen annimmt. Da der zentrale Server Ergebnisse von [passiven Service-Prüfungen](#) von einem oder mehreren verteilten Servern erhält, dient er als Mittelpunkt der gesamten Überwachungslogik (d.h., er versendet Benachrichtigungen, startet Eventhandler-Scripts, legt den Zustand von Hosts fest, enthält das Web-Interface, usw.).

Service-Prüfungsinformationen von verteilten Servern erhalten

Bevor wir näher auf Konfigurationsdetails eingehen, müssen wir wissen, wie die Service-Prüfungsergebnisse von den verteilten Servern zum zentralen Server geschickt werden. Wir haben bereits erwähnt, wie man passive Prüfungsergebnisse an den gleichen Host schickt, auf dem Icinga läuft (wie in der Dokumentation zu [passive Prüfungen](#) beschrieben), aber wir haben keinerlei Informationen darüber gegeben, wie man passive Prüfergebnisse von anderen Hosts verschickt.

Um den Versand von passiven Prüfergebnissen an einen anderen Host zu erleichtern, haben wir das [nsca-Addon](#) geschrieben. Das Addon besteht aus zwei Teilen. Das erste ist ein Client-Programm (`send_nsca`), das auf einem entfernten Host läuft und benutzt wird, um die Service-Prüfergebnisse an einen anderen Server zu senden. Das zweite Teil ist der nsca-Daemon (`nsca`), der entweder als eigenständiger Daemon oder unter `inetd` läuft und auf Verbindungen von Client-Programmen horcht. Nach dem Empfang von Service-Prüfinformationen von einem Client wird der Daemon die Prüfinformationen an Icinga (auf dem zentralen Server) weiterleiten, indem ein `PROCESS_SVC_CHECK_RESULT` zusammen mit den Prüfergebnissen in das [external command file](#) eingefügt wird. Das nächste Mal, wenn Icinga auf [externe Befehle](#) prüft, wird es die passiven Prüfergebnisse finden, die von den verteilten Servern geschickt wurden und sie verarbeiten. Einfach, oder?

Verteilte Server-Konfiguration

Also wie genau wird Icinga auf einem verteilten Server konfiguriert? Grundsätzlich ist es eine einfache Installation. Sie müssen weder ein Web-Interface installieren noch Benachrichtigungen versenden, weil dies alles vom zentralen Server aus erledigt wird.

Haupt-Konfigurationsanpassungen:

- Nur die direkt durch den verteilten Server zu überwachenden Services werden in der [Objekt-Konfigurationsdatei](#) definiert.
- Die [enable_notifications](#)-Direktive auf dem verteilten Server wird auf 0 gesetzt. Das verhindert das Versenden von Benachrichtigungen.
- Die [obsess over services](#)-Direktive auf dem verteilten Server wird aktiviert.
- Auf dem verteilten Server ist ein [ocsp command](#) definiert (wie unten beschrieben).

Damit alles zusammenkommt und ordentlich arbeitet, wollen wir, dass der verteilte Server die Ergebnisse *aller* Service-Prüfungen an Icinga meldet. Wir können [Eventhandler](#) benutzen, um Änderungen am Zustand eines Service mitzuteilen, aber das bringt's nicht. Um den verteilten Server zu zwingen, alle Prüfergebnisse zu melden, müssen Sie die [obsess_over_services](#)-Option in der Hauptkonfigurationsdatei aktivieren und ein [ocsp_command](#) bereitstellen, was nach jeder Service-Prüfung ausgeführt wird. Wir werden das ocsp-Kommando benutzen, um die Ergebnisse aller Service-Prüfungen an den zentralen Server zu senden und den `send_nsca`-Client sowie den nsca-Daemon benutzen (wie oben beschrieben), um die Übertragung zu erledigen.

Um dies zu erreichen, müssen Sie ein ocsp-Kommando wie folgt definieren:

ocsp_command=submit_check_result

Die Definition für den *submit_check_result*-Befehl sieht ungefähr so aus:

```
define command{ command_name submit_check_result command_line /usr/local/icinga/libexec/eventhandlers/submit_check_result $HOSTNAME$ '$SERVICEDESC$' '$SERVICESTATE$' '$SERVICEOUTPUT$' }
```

Die *submit_check_result* Shell-Skripte sehen ungefähr so aus (ersetzen Sie *central_server* durch die IP-Adresse des zentralen Servers):

```
#!/bin/sh
# Arguments:
# $1 = host_name (Short name of host that the service is
#   associated with)
# $2 = svc_description (Description of the service)
# $3 = state_string (A string representing the status of
#   the given service - "OK", "WARNING", "CRITICAL"
#   or "UNKNOWN")
# $4 = plugin_output (A text string that should be used
#   as the plugin output for the service checks)
#
# Convert the state string to the corresponding return code
return_code=-1
case "$3" in
    OK)
        return_code=0
        ;;
    WARNING)
        return_code=1
        ;;
    CRITICAL)
        return_code=2
        ;;
    UNKNOWN)
        return_code=-1
        ;;
esac
# pipe the service check info into the send_nsca program, which
# in turn transmits the data to the nsca daemon on the central
# monitoring server
/bin/printf "%s\t%s\t%s\t%s\n" "$1" "$2" "$return_code" "$4" | /usr/local/icinga/bin/send_nsca -H central_server -c /usr/local/icinga/etc/send_nsca.cfg
```

Das Skript oben geht davon aus, dass das *send_nsca*-Programm und die Konfigurationsdatei (*send_nsca.cfg*) in den Verzeichnissen */usr/local/icinga/bin/* und */usr/local/icinga/etc/* zu finden sind.

Das ist alles! Wir haben erfolgreich einen entfernten Host konfiguriert, auf dem Icinga als ein verteilter Überwachungs-Server läuft. Lassen Sie uns genau betrachten, was mit dem verteilten Server passiert und wie er Service-Prüfungsergebnisse an Icinga schickt (die unten skizzierten Schritte entsprechen den Zahlen im obigen Referenzdiagramm):

1. Nachdem der verteilte Server eine Service-Prüfung beendet hat, führt er den Befehl aus, den Sie mit der Variable **ocsp_command** definiert haben. In unserem Beispiel ist dies das */usr/local/icinga/libexec/eventhandlers/submit_check_result*-Script. Beachten Sie, dass die Definition für den *submit_check_result*-Befehl vier Parameter für das Script übergibt: den Namen des Hosts, der mit dem Service verbunden ist, die Service-Beschreibung, den Rückgabewert der Service-Prüfung und die Plugin-Ausgabe der Service-Prüfung.
2. das *submit_check_result*-Script übergibt die Informationen der Service-Prüfung (Host-Name, Beschreibung, Rückgabewert und Ausgabe) an das *send_nsca*-Client-Programm.
3. das *send_nsca*-Programm überträgt die Informationen der Service-Prüfung an den *nsca*-Daemon auf dem zentralen Überwachungs-Server.
4. der *nsca*-Daemon auf dem zentralen Server nimmt die Informationen der Service-Prüfung und schreibt sie in das external command file, damit Icinga sie später dort aufsammeln kann.
5. der Icinga-Prozess auf dem zentralen Server liest das external command file und verarbeitet die passiven Service-Prüfungsergebnisse, die vom verteilten Überwachungs-Server stammen.

zentrale Server-Konfiguration

Wir haben betrachtet, wie verteilte Überwachungs-Server konfiguriert werden sollten, daher wenden wir uns nun dem zentralen Server zu. Für alle wichtigen Dinge wird der zentrale so konfiguriert wie ein einzelner Server. Dessen Setup ist wie folgt:

- auf dem zentralen Server ist das Web-Interface installiert (optional, aber empfohlen)
- auf dem zentralen Server ist die [enable_notifications](#)-Direktive auf 1 gesetzt. Das aktiviert Benachrichtigungen (optional, aber empfohlen)
- auf dem zentralen Server sind [aktive Service-Prüfungen](#) deaktiviert (optional, aber empfohlen - beachten Sie die folgenden Anmerkungen)
- auf dem zentralen Server sind [external command checks](#) aktiviert (erforderlich)
- auf dem zentralen Server sind [passive Service-Prüfungen](#) aktiviert (erforderlich)

Es gibt drei andere sehr wichtige Dinge, die Sie beachten sollten, wenn Sie den zentralen Server konfigurieren:

- Der zentrale Server muss Service-Definitionen für *alle Services* haben, die auf allen verteilten Servern überwacht werden. Icinga wird passive Prüfungsergebnisse ignorieren, wenn sie nicht zu einem Service passen, den Sie definiert haben.
- Wenn Sie den zentralen Server nur benutzen, um Services zu verarbeiten, deren Ergebnisse von verteilten Hosts stammen, können Sie alle aktiven Service-Prüfungen auf programmweiter Basis durch das Setzen der [execute_service_checks](#)-Direktive auf 0 deaktivieren. Wenn Sie den zentralen Server nutzen, um selbst einige Services aktiv zu überwachen (ohne die Hilfe von verteilten Servern), dann sollten Sie die [active_checks_enabled](#)-Option der Service-Definitionen auf 0 setzen, die von den verteilten Servern überwacht werden. Das hindert Icinga daran, diese Services aktiv zu prüfen.

Es ist wichtig, dass Sie entweder alle Service-Prüfungen auf einer programmweiten Basis deaktivieren oder die [enable_active_checks](#)-Option in jeder Service-Definition deaktivieren, die von einem verteilten Server überwacht werden. Das stellt sicher, dass aktive Service-Prüfungen unter normalen Umständen niemals ausgeführt werden. Die Services werden weiterhin im normalen Prüfintervall geplant (3 Min., 5 Min., usw.), aber nicht ausgeführt. Wir werden bald erklären, warum das so ist...

Das war's! Einfach, oder?

Probleme bei passiven Prüfungen

Für alle wichtigen Dinge können wir sagen, dass sich der zentrale Server bei Überwachungen allein auf passive Prüfungen verlässt. Das Hauptproblem daran, sich komplett auf passive Prüfungen zu verlassen besteht darin, dass Icinga darauf vertrauen muss, dass jemand anders die Daten liefert. Was passiert, wenn der entfernte Host, der passive Prüfergebnisse sendet, herunterfährt oder unerreichbar wird? Wenn Icinga nicht aktiv die Services auf dem Host prüft, wie soll es wissen, wann es ein Problem gibt?

Glücklicherweise gibt es einen Weg, diese Art von Problemen zu behandeln...

Frische-Prüfung (Freshness Checking)

Icinga unterstützt ein Feature, das eine "Frische"-Prüfung für die Ergebnisse von Service-Prüfungen durchführt. Mehr Informationen über Frische-Prüfung finden Sie [hier](#). Dieses Feature sorgt für etwas Schutz gegen Situationen, in denen entfernte Hosts keine passiven Service-Prüfungen mehr an den zentralen Überwachungs-Server schicken. Der Zweck der "Frische"-Prüfung besteht darin, sicherzustellen, dass Service-Prüfungen entweder regelmäßig passiv durch verteilte Server oder aktiv durch den zentralen Server durchgeführt werden, falls dies notwendig sein sollte. Wenn die Service-Prüfergebnisse von verteilten Servern als "abgestanden" angesehen werden, kann Icinga so konfiguriert werden, um aktive Prüfungen des Service vom zentralen Überwachungs-Server aus zu erzwingen.

Wie machen Sie das? Auf dem zentralen Überwachungs-Server müssen Sie Services konfigurieren, die von verteilten Servern wie folgt überwacht werden:

- Die *check_freshness*-Option in der Service-Definition ist auf 1 zu setzen. Das aktiviert "Frische"-Prüfungen für den Service.
- Die *freshness_threshold*-Option in den Service-Definitionen sollte auf einen Wert (in Sekunden) gesetzt werden, der widerspiegelt, wie "frisch" die (von den entfernten Servern gelieferten) Ergebnisse der Service-Prüfungen sein sollten.
- Die *check_command*-Option in den Service-Definitionen sollte gültige Befehle enthalten, die genutzt werden können, um den Service aktiv vom zentralen Server aus zu prüfen.

Icinga prüft periodisch die "Frische" der Ergebnisse aller Services, für die Frische-Prüfungen aktiviert sind. Die *freshness_threshold*-Option in jeder Service-Definition wird benutzt, um festzulegen, wie "frisch" die Ergebnisse für jeden Service sein sollen. Wenn Sie z.B. diesen Wert für einen Ihrer Services auf 300 setzen, wird Icinga das Service-Ergebnis als "abgestanden" betrachten, wenn es älter als 5 Minuten (300 Sekunden) ist. Falls Sie keinen Wert für die *freshness_threshold*-Option angeben, wird Icinga automatisch einen "Frische"-Schwellwert berechnen, indem es die Werte der *check_interval*- oder der *retry_interval*-Option betrachtet (abhängig vom [Statustyp](#), in dem sich der Service befindet). Wenn die Service-Ergebnisse als "abgestanden" angesehen werden, wird Icinga den Service-Prüf-Befehl ausführen, der in der *check_command*-Option der Service-Definition angegeben ist, und dadurch den Service aktiv prüfen.

Denken Sie daran, dass Sie eine *check_command*-Option in den Service-Definitionen angeben müssen, die genutzt werden kann, um den Status des Service aktiv vom zentralen Server aus zu prüfen. Unter normalen Umständen wird dieser Prüfbefehl niemals ausgeführt (weil aktive Prüfungen auf programmweiter Ebene bzw. für den einzelnen Service deaktiviert wurden). Wenn Frische-Prüfungen aktiviert sind, wird Icinga diesen Befehl ausführen, um den Zustand des Service aktiv zu prüfen, *auch wenn aktive Prüfungen auf einer programmweiten Ebene oder Service-spezifischen Basis deaktiviert sind*.

Falls Sie es nicht schaffen, Befehle zu definieren, um aktiv einen Service vom zentralen Überwachungs-Host aus zu prüfen (oder wenn es zu einer großen Qual wird), können Sie ganz einfach bei all Ihren Services in der *check_command*-Option ein Dummy-Script angeben, das einen kritischen Status zurücklieferst. Hier ein Beispiel... Lassen Sie uns annehmen, Sie definieren einen Befehl namens 'service-is-stale' und benutzen den Befehlsnamen in der *check_command*-Option Ihrer Services. Hier nun, wie die Definition aussehen könnte...

```
define command{ command_name service-is-stale command_line /usr/local/icinga/libexec/check_dummy 2 "CRITICAL: Service results are stale" }
```

Wenn Icinga feststellt, dass das Service-Ergebnis abgestanden ist und das **service-is-stale**-Kommando aufruft, wird das */usr/local/icinga/libexec/check_dummy*-Plugin ausgeführt und der Service geht in einen kritischen Zustand. Das wird wahrscheinlich dazu führen, dass Benachrichtigungen versandt werden, so dass Sie wissen, dass es ein Problem gibt.

Host-Prüfungen durchführen

An diesem Punkt wissen Sie, wie man Service-Ergebnisse von verteilten Servern auf passive Weise erhält. Das bedeutet, der zentrale Server nicht aktiv Service-Prüfungen ausführt. Aber was ist mit Host-Prüfungen? Sie müssen sie trotzdem erledigen, aber wie?

Nachdem Host-Prüfungen normalerweise einen kleinen Teil der Überwachungsaktivität verbrauchen (sie werden nur ausgeführt, wenn es dringend notwendig ist), raten wir dazu, dass Sie die Host-Prüfungen aktiv vom zentralen Server aus durchführen. Das bedeutet, dass Sie Host-Prüfungen auf dem zentralen Server genau wie auf den verteilten Servern definieren (und auf die gleiche Weise, wie Sie das in einer normalen, nicht-verteilten Umgebung tun würden).

Passive Host-Prüfungen sind verfügbar (lesen Sie [hier](#)), so dass Sie diese in Ihrer verteilten Umgebung nutzen können, allerdings gibt es dabei ein paar Probleme. Das größte Problem besteht darin, dass Icinga Ergebnisse von passiven Host-Prüfungen (die Problemzustände DOWN und UNREACHABLE) nicht übersetzt, wenn sie verarbeitet werden. Das bedeutet, falls Ihre Überwachungs-Server eine unterschiedliche Eltern-/Kind-Host-Struktur haben (und das werden sie, wenn Ihre Überwachungs-Server an unterschiedlichen Standorten stehen), wird der zentrale Überwachungs-Server eine ungenaue Sicht Ihrer Host-Zustände haben.

Falls Sie in Ihrer verteilten Überwachungs-Umgebung passive Host-Prüfungen an einen zentralen Server senden möchten, dann stellen Sie sicher:

- dass auf dem zentralen Server [passive Host-Prüfungen](#) aktiviert sind (notwendig)
- dass auf dem verteilten Server [obsess over hosts](#) aktiviert ist.
- dass auf dem verteilten Server ein [ochp command](#) definiert ist.

Der ochp-Befehl, der zur Verarbeitung von Host-Prüfergebnissen genutzt wird, arbeitet ähnlich wie der ocsp-Befehl, der für die Verarbeitung von Service-Prüfergebnissen benutzt wird (siehe oben). Um sicherzustellen, dass passive Host-Prüfergebnisse aktuell sind, sollten Sie [Frische-Prüfungen](#) für Hosts aktivieren (ähnlich zu dem, was weiter oben für Services beschrieben wird).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Service- und
Host-Frische-Prüfungen

[Zum Anfang](#)

Redundante und
Failover-Netzwerk-Überwachung

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Redundante und Failover-Netzwerk-Überwachung

[Zurück](#)

[Kapitel 7. Fortgeschrittene Themen](#)

[Weiter](#)

Redundante und Failover-Netzwerk-Überwachung

Einführung

Dieser Abschnitt beschreibt einige Szenarien zum Implementieren von redundanten Überwachungs-Hosts auf verschiedenen Arten von Netzwerk-Layouts. Mit redundanten Hosts können Sie die Überwachung Ihres Netzwerkes aufrecht erhalten, wenn der primäre Host, auf dem Icinga läuft, ausfällt oder wenn Teile Ihres Netzwerkes unerreichbar werden.

Anmerkung: Wenn Sie gerade lernen, wie Icinga zu nutzen ist, würden wir empfehlen, Redundanz so lange nicht zu implementieren, bis Sie mit den [Voraussetzungen](#) vertraut sind. Redundanz ist ein relativ komplexes Thema und es ist noch schwieriger, es zu implementieren.

Index

[Voraussetzungen](#)

[Beispiel-Scripte](#)

[Szenario 1 - Redundante Überwachung](#)

[Szenario 2 - Failover Überwachung](#)

Voraussetzungen

Bevor Sie überhaupt daran denken können, Redundanz mit Icinga zu implementieren, müssen Sie mit folgenden Dingen vertraut werden...

- Implementieren von [EventHandlers](#) für Hosts und Services
- Erteilen von [externen Befehlen](#) an Icinga über Shell-Scripts
- Ausführen von Plugins auf entfernten Hosts mit Hilfe des [NRPE Addons](#) oder einer anderen Methode
- Überprüfen des Zustands des Icinga-Prozesses mit dem *check_nagios* Plugin

Beispiel-Scripte

Jedes dieser Beispiel-Scripte, die wir in dieser Dokumentation benutzen, finden Sie im *eventhandlers*-Unterverzeichnis der Icinga-Distribution. Vielleicht müssen Sie sie modifizieren, damit sie auf Ihrem System funktionieren...

Szenario 1 - Redundante Überwachung

Einführung

Dies ist eine einfache (und harmlose) Methode, redundante Überwachungs-Hosts zu implementieren, und es wird nur gegen eine begrenzte Anzahl von Ausfällen schützen. Komplexere Setups werden benötigt, um intelligenter Redundanz, bessere Redundanz über verschiedene Netzwerk-Segmente hinweg zu bieten.

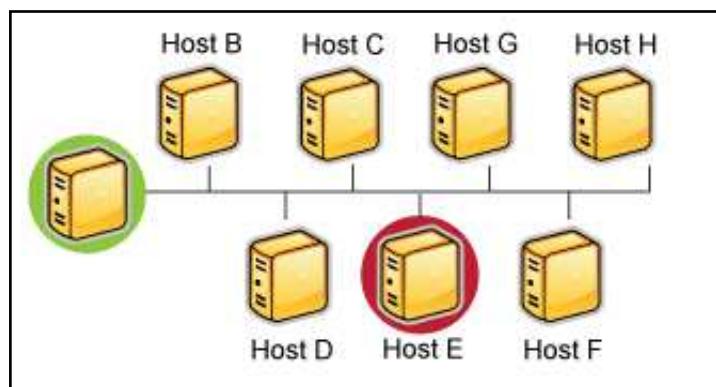
Ziele

Das Ziel dieser Art von Redundanz-Implementierung ist einfach. Sowohl der "Master"- als auch der "Slave"-Host überwachen die gleichen Hosts und Services auf dem Netzwerk. Unter normalen Umständen wird nur der "Master"-Host Benachrichtigungen an Kontakte versenden. Wir wollen, dass der "Slave"-Host die Benachrichtigung von Kontakten übernimmt, wenn:

1. der "Master"-Host, auf dem Icinga läuft, "down" ist oder...
2. der Icinga-Prozess auf dem "Master"-Host aus irgendeinem Grund stoppt

Netzwerk-Layout-Diagramm

Das untenstehende Diagramm zeigt ein sehr simples Netzwerk-Setup. Bei diesem Szenario nehmen wir an, dass auf den Hosts A und E Icinga läuft und alle gezeigten Hosts überwacht werden. Host A ist der "Master"-Host und Host E der "Slave"-Host.



anfängliche Programmeinstellungen

Auf dem Slave-Host (Host E) wird die ursprüngliche `enable_notifications`-Direktive deaktiviert, so dass dadurch der Versand von Host- oder Service-Benachrichtigungen verhindert wird. Sie sollten auch sicherstellen, dass die `check_external_commands`-Direktive deaktiviert ist. Das war einfach genug...

anfängliche Konfiguration

Als nächstes sollten wir die Unterschiede zwischen den [Objekt-Konfigurationsdatei](#) von Master- und Slave-Host(s) betrachten...

Wir gehen davon aus, dass Sie den Master-Host (Host A) so konfiguriert haben, dass er alle Services auf den gezeigten Hosts des Diagramms überwacht. Der Slave-Host (Host E) sollte die gleichen Hosts und Services überwachen, mit folgenden Zusätzen in der Konfigurationsdatei...

- Die Host-Definition für Host A (in der Host-Konfigurationsdatei von Host E) sollte einen Host-[Eventhandler](#) enthalten. Der Name für den Host-Eventhandler lautet [handle-master-host-event](#).

- Die Konfigurationsdatei auf Host E enthält einen Service, der den Status des Icinga-Prozesses auf Host A prüft. Lassen Sie uns annehmen, dass diese Prüfung das `check_nagios`-Plugin auf Host A aufruft. Das kann durch eine der in den [FAQ](#) beschriebenen Methoden erfolgen.
- Die Service-Definition für den Icinga-Prozess auf Host A sollte einen [Eventhandler](#)-Eintrag enthalten. Als Namen für diese Service-Eventhandler wählen wir **handle-master-proc-event**.

Es ist wichtig anzumerken, dass Host A (der Master-Host) keine Ahnung von Host E (dem Slave-Host) hat. In diesem Szenario besteht ganz einfach keine Notwendigkeit dazu. Natürlich können Sie von Host A Services auf Host E überwachen, aber das hat nichts mit der Implementierung von Redundanz zu tun...

Eventhandler-Befehlsdefinitionen

Wir müssen kurz innehalten und beschreiben, wie die Befehlsdefinitionen für die Eventhandler auf dem Slave-Host aussehen. Hier ist ein Beispiel...

```
define command{
    command_name handle-master-host-event
    command_line /usr/local/icinga/libexec/eventhandlers/handle-master-host-event $HOSTSTATE$ $HOSTSTATETYPE$
}
define command{
    command_name handle-master-proc-event
    command_line /usr/local/icinga/libexec/eventhandlers/handle-master-proc-event $SERVICESTATE$ $SERVICESTATETYPE$
```

Dies setzt voraus, dass Sie die Eventhandler-Skripte im Verzeichnis `/usr/local/icinga/libexec/eventhandlers` abgelegt haben. Sie können sie ablegen, wohin Sie wollen, aber dann müssen Sie die beigefügten Beispiele anpassen.

Eventhandler-Skripte

Okay, lassen Sie uns nun einen Blick darauf werden, wie die Eventhandler-Skripte aussehen...

Host-Eventhandler (**handle-master-host-event**):

```
#!/bin/sh
# Only take action on hard host states...
case "$2" in
HARD)
    case "$1" in
DOWN)
        # The master host has gone down!
        # We should now become the master host and take
        # over the responsibilities of monitoring the
        # network, so enable notifications...
        /usr/local/icinga/libexec/eventhandlers/enable_notifications
        ;;
UP)
        # The master host has recovered!
        # We should go back to being the slave host and
        # let the master host do the monitoring, so
        # disable notifications...
        /usr/local/icinga/libexec/eventhandlers/disable_notifications
        ;;
esac
;;
esac
exit 0
```

Service-Eventhandler (*handle-master-proc-event*):

```

#!/bin/sh
# Only take action on hard service states...
case "$2" in
HARD)
    case "$1" in
CRITICAL)
        # The master Icinga process is not running!
        # We should now become the master host and
        # take over the responsibility of monitoring
        # the network, so enable notifications...
        /usr/local/icinga/libexec/eventhandlers/enable_notifications
        ;;
WARNING)
UNKNOWN)
        # The master Icinga process may or may not
        # be running.. We won't do anything here, but
        # to be on the safe side you may decide you
        # want the slave host to become the master in
        # these situations...
        ;;
OK)
        # The master Icinga process running again!
        # We should go back to being the slave host,
        # so disable notifications...
        /usr/local/icinga/libexec/eventhandlers/disable_notifications
        ;;
esac
;;
esac
exit 0

```

Was tun sie für uns

Auf dem Slave-Host (Host E) sind anfänglich die Benachrichtigungen deaktiviert, so dass er keine Host- oder Service-Benachrichtigungen versendet, solange der Icinga-Prozess auf dem Master-Host (Host A) noch läuft.

Der Icinga-Prozess auf dem Slave-host (Host E) wird zum Master-Host, wenn...

- der Master-Host (Host A) "down" geht und der *handle-master-host-event*-Host-Eventhandler ausgeführt wird.
- der Icinga-Prozess auf dem Master-Host (Host A) aufhört zu arbeiten und der *handle-master-proc-event*-Service-Eventhandler ausgeführt wird.

Wenn bei dem Icinga-Prozess auf dem Slave-Host (Host E) Benachrichtigungen aktiviert sind, kann er Benachrichtigungen über jegliche Host- und Service-Probleme und Erholungen versenden. An diesem Punkt hat Host E die Verantwortlichkeiten über die Benachrichtigung von Kontakten über Host- und Service-Probleme übernommen!

Der Icinga-Prozess auf Host E wird wieder zum Host-Slave, wenn...

- sich Host A wieder erholt und der *handle-master-host-event*-Host-Eventhandler ausgeführt wird.
- sich der Icinga-Prozess auf Host A wieder erholt und den *handle-master-proc-event*-Service-Eventhandler ausführt.

Wenn bei dem Icinga-Prozess auf dem Slave-Host (Host E) Benachrichtigungen deaktiviert sind, wird er keine Benachrichtigungen mehr über Host- und Service-Probleme und Erholungen versenden. An diesem Punkt hat Host E die Verantwortlichkeiten über die Benachrichtigung von Kontakten über Host- und Service-Probleme an Host A übergeben. Alles ist wieder so, als wir angefangen haben!

Zeitverzögerungen

Redundanz bei Icinga ist in keiner Weise perfekt. Eins der offenkundigeren Probleme ist die Verzögerung zwischen dem Ausfall von Host A und der Übernahme durch Host E. Das ist bedingt durch folgende Dinge...

- die Zeit zwischen dem Ausfall des Master-Host und dem ersten Mal, dass der Slave-Host ein Problem entdeckt
- die Zeit, die benötigt wird, um festzustellen, dass der Master-Host wirklich ein Problem hat (unter Verwendung von Host- oder Service-Prüfwiederholungen auf dem Slave-Host)
- die Zeit zwischen der Ausführung des Eventhandlers und der Zeit, zu der Icinga das nächste Mal auf externe Befehle prüft

Sie können diese Verzögerung minimieren durch...

- eine hohe Frequenz von (Wiederholungs-) Prüfungen für Services auf Host E. Das kann durch die `check_interval`- und `retry_interval`-Optionen in jeder Service-Definition erreicht werden.
- eine Zahl der Host-Wiederholungsprüfungen für Host A (auf Host E), die eine schnelle Erkennung von Host-Problemen erlaubt. Das wird erreicht durch das `max_check_attempts`-Argument in der Host-Definition.
- erhöhen der Frequenz der `external command`-Prüfungen auf Host E. Dies wird erreicht durch die Anpassung der `command_check_interval`-Option in der Hauptkonfigurationsdatei.

Wenn sich Icinga auf Host A erholt, gibt es ebenfalls eine Verzögerung, bevor Host E wieder zu einem Slave-Host wird. Das wird durch folgende Dinge beeinflusst...

- die Zeit zwischen der Erholung des Master-Hosts und der Zeit, zu der der Icinga-Prozess auf Host E die Erholung erkennt
- die Zeit zwischen der Ausführung des Eventhandlers auf Host A und der Zeit, zu der der Icinga-Prozess auf Host E das nächste Mal auf externe Befehle prüft

Die genaue Verzögerung zwischen dem Übergang der Verantwortlichkeiten hängt davon ab, wieviele Services Sie definiert haben, dem Intervall, in dem Services geprüft werden, und einer Menge pures Glück. Auf jeden Falls ist es besser als nichts.

Spezialfälle

Eins sollten Sie beachten: Wenn Host A "down" geht, werden bei Host E die Benachrichtigungen aktiviert und er übernimmt die Verantwortung für das Informieren der Kontakte bei Problemen. Wenn sich Host A wieder erholt, werden bei Host E die Benachrichtigungen deaktiviert. Falls der Icinga-Prozess - wenn sich Host A erholt - auf Host A nicht sauber startet, gibt es eine Zeitspanne, während der keiner der beiden Hosts die Kontakte über Probleme informiert! Glücklicherweise berücksichtigt die Service-Prüflogik in Icinga diesen Umstand. Das nächste Mal, wenn der Icinga-Prozess auf Host E den Status des

Icinga-Prozesses auf Host A prüft, wird er feststellen, dass dieser nicht läuft. Auf Host E werden dann wieder die Benachrichtigungen aktiviert und er wird erneut die Verantwortung für die Benachrichtigung der Kontakte übernehmen.

Der exakte Wert für die Zeit, während der keiner der Hosts das Netzwerk überwacht, ist schwer zu ermitteln. Offensichtlich kann diese Zeit durch die Erhöhung der Frequenz von Service-Prüfungen (auf Host E) für Host A minimiert werden. Der Rest ist purer Zufall, aber die gesamte "Blackout"-Zeit sollte nicht allzu hoch sein.

Szenario 2 - Failover-Überwachung

Einführung

Failover-Überwachung ist ähnlich wie die redundante Überwachung (wie beschrieben in [Szenario 1](#)).

Ziele

Das grundlegende Ziel der Failover-Überwachung besteht darin, dass der Icinga-Prozess auf dem Slave-Host untätig ist, während der Icinga-Prozess auf dem Master-Host läuft. Wenn der Prozess auf dem Master-Host stoppt (oder der Host "down" geht), übernimmt der Icinga-Prozess auf dem Slave-Host die gesamte Überwachung.

Während es Ihnen die in [Szenario 1](#) beschriebene Methode erlaubt, weiterhin Benachrichtigungen zu erhalten, wenn der Master-Host "down" geht, gibt es einige Fallen. Das größte Problem besteht darin, dass der Slave-Host die gleichen Hosts und Services wie der Master *zur gleichen Zeit wie der Master* überwacht! Dies kann Probleme durch übermäßigen Traffic und Load auf den überwachten Maschinen verursachen, wenn Sie viele Services definiert haben. Hier nun, wie Sie das Problem umgehen können.

Initiale Programm-Einstellungen

Deaktivieren Sie aktive Service-Prüfungen und Benachrichtigungen auf dem Slave-Host durch die `execute_service_checks`- und die `enable_notifications`-Direktiven. Dies wird den Slave-Host davon abhalten, Services und Hosts zu überwachen und Benachrichtigungen zu versenden, während der Icinga-Prozess auf dem Master-Host noch läuft. Stellen Sie außerdem sicher, dass die `check_external_commands`-Direktive auf dem Slave-Host aktiviert ist.

Master-Prozess-Prüfungen

Setzen Sie einen cron-Job auf dem Slave-Host auf, der periodisch (sagen wir jede Minute) läuft und den Status des Icinga-Prozesses auf dem Master-Host (mit dem `check_nrpe` auf dem Slave-Host und den `nrpe daemon` und `check_nagios`-Plugins auf dem Master-Host) prüft. Das Script sollte den Return-Code des `check_nrpe`-Plugins prüfen. Falls es einen nicht-OK-Status zurückliefert, sollte das Script den entsprechenden Befehl an das `external command file` senden, um sowohl die Benachrichtigungen als auch die aktiven Service-Prüfungen zu aktivieren. Falls das Plugin einen OK-Status zurückliefert, sollte das Script Befehle an das `external command file` senden, um sowohl Benachrichtigungen als auch aktive Prüfungen zu deaktivieren.

Auf diese Weise läuft jeweils nur ein Prozess, der Hosts und Services prüft, was wesentlich effizienter ist als alles doppelt zu überwachen.

Auch von Interesse: Sie müssen *nicht* wie in [Szenario 1](#) beschrieben die Host- und Service-Handler definieren, weil die Dinge anders behandelt werden.

Zusätzliche Themen

An diesem Punkt haben Sie ein sehr einfaches Failover-Überwachungs-Setup implementiert. Trotzdem gibt es einen weiteren Punkt, den Sie berücksichtigen sollten, damit die Dinge besser laufen.

Das große Problem dabei, wie die Dinge bisher konfiguriert sind, besteht darin, dass der Slave-Host nicht den aktuellen Status von Hosts und Services kennt, wenn er die Überwachung übernimmt. Ein Weg, dieses Problem zu lösen, ist es, die [ocsp command](#)-Option auf dem Master-Host zu aktivieren und alle Service-Prüfergebnisse mit dem [nsca Addon](#) an den Slave-Host zu schicken. Der Slave-Host wird dann aktuelle Status-Informationen für alle Services haben, wenn er die Überwachung übernimmt. Weil aktive Service-Prüfungen auf dem Slave-Host nicht aktiviert sind, werden sie nicht ausgeführt. Host-Prüfungen hingegen werden nach Bedarf ausgeführt. Das bedeutet, dass sowohl Master- als auch Slave-Host Host-Prüfungen ausführen, wenn sie benötigt werden, was kein Problem darstellen sollte, weil die Mehrzahl der Überwachung Service-Prüfungen betrifft.

Das ist eigentlich alles, was das Setup betrifft.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Verteilte Überwachung](#)

[Zum Anfang](#)

[Erkennung und Behandlung von
Status-Flattern](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Erkennung und Behandlung von Status-Flattern

[Zurück](#)[Kapitel 7. Fortgeschrittene Themen](#)[Weiter](#)

Erkennung und Behandlung von Status-Flattern

Einführung

Icinga unterstützt die Erkennung von Hosts und Services, die "flattern". Flattern tritt auf, wenn Hosts oder Services zu oft den Zustand wechseln und dadurch einen Sturm von Problemen und Erholungsbenachrichtigungen erzeugen. Flattern kann auf Konfigurationsprobleme hinweisen (z.B. Schwellwerte, die zu niedrig gesetzt sind), störende Services oder wirkliche Netzwerkprobleme.

Wie Flatter-Erkennung arbeitet

Bevor wir darauf eingehen, lassen Sie uns sagen, dass es etwas schwierig war, Flatter-Erkennung zu implementieren. Wie genau legt man fest, was "zu häufig" in Bezug auf Statusänderungen für einen Host oder Service ist? Als Ethan Galstad zuerst an die Implementierung der Flatter-Erkennung gedacht hat, versuchte er Informationen zu finden, wie Flattern erkannt werden könnte/sollte. Er konnte keinerlei Informationen darüber finden, was andere benutzten (benutzen andere so etwas?), also entschied er sich für das, was er für eine sinnvolle Lösung hielt...

Sobald Icinga den Zustand eines Hosts oder Services prüft, wird es prüfen, ob dafür Flattern begonnen oder geendet hat. Es tut dies durch:

- speichern der Ergebnisse der letzten 21 Prüfungen des Hosts oder Service
- analysieren der historischen Prüfergebnisse und feststellen, wo Statusänderungen/-übergänge auftreten
- benutzen der Statusübergänge, um einen Statuswechsel-Prozentsatz (ein Maß für die Änderung) für den Statuswechsel des Hosts oder Service festzulegen
- vergleichen des Statuswechsel-Prozentwertes gegen die Flatter-Schwellwerte (hoch und niedrig)

Ein Host oder Service wird angesehen, mit dem Flatter *begonnen* zu haben, wenn der Prozentsatz das erste Mal einen *hohen* Flatter-Schwellwert überschritten hat.

Ein Host oder Service wird angesehen, das Flattern *beendet* zu haben, wenn der Prozentsatz unter einen *niedrigen* Flatter-Schwellwert sinkt (vorausgesetzt, dass er vorher geflattert hat).

Beispiel

Lassen Sie uns etwas detaillierter beschreiben, wie Flatter-Erkennung bei Services arbeitet...

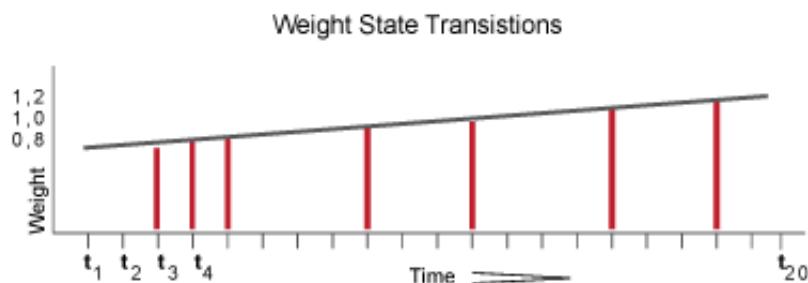
Das Bild unten zeigt eine chronologische Historie von Service-Zuständen der letzten 21 Service-Prüfungen. OK-Zustände sind in grün dargestellt, WARNING-Zustände in gelb, CRITICAL-Zustände in rot und UNKNOWN-Zustände in orange.



Die historischen Service-Prüfergebnisse werden untersucht, um festzustellen, wo Statusänderungen/-übergänge auftreten. Statusänderungen treten auf, wenn ein archivierter Status sich von den archivierten Zuständen unterscheidet, die ihm direkt vorausgehen. Da wir die Ergebnisse der letzten 21 Status-Prüfungen in dem Array ablegen, können wir bis zu 20 Statusänderungen haben. In diesem Beispiel gibt es sieben Statusänderungen, die im Bild durch blaue Pfeile gekennzeichnet sind.

Die Flatter-Erkennungslogik nutzt die Statusänderungen, um einen Gesamtprozentsatz für den Service festzulegen. Dies ist ein Maßstab für die Sprunghaftigkeit/Änderung des Service. Services, die nie den Status wechseln, haben einen Statusänderungswert von 0%, während Services, die ihren Status bei jeder Prüfung wechseln, einen Wert von 100% haben. Die meisten Services werden einen Prozentwert irgendwo dazwischen haben.

Während der Berechnung des Prozentsatzes für den Service wird der Flatter-Erkennungsalgorismus mehr Gewicht auf neuere Statusänderungen legen als auf alte. Genauer gesagt sind die Flatter-Erkennungsrouterien im Moment so ausgelegt, dass der neueste Statuswechsel 50% mehr Gewicht hat als der älteste. Das Bild unten zeigt, wie neuere Statuswechsel mehr Gewicht erhalten als ältere, während der Gesamtprozentwert für einen bestimmten Service berechnet wird.



Lassen Sie uns mit dem obigen Bild eine Berechnung der prozentualen Statusänderungen für den Service durchführen. Sie werden bemerken, dass es insgesamt sieben Statuswechsel gibt (bei t₃, t₄, t₅, t₉, t₁₂, t₁₆ und t₁₉). Ohne Gewichtung der Statuswechsel über die Zeit würde dies einen Gesamtwert von 35% ergeben:

$$(7 \text{ beobachtete Statuswechsel} / 20 \text{ mögliche Statuswechsel}) * 100 = 35 \%$$

Nachdem die Flatter-Erkennungslogik neueren Statuswechseln mehr Gewicht gibt als älteren, wird der eigentliche Wert in diesem Beispiel geringfügig kleiner sein als 35%. Lassen Sie uns annehmen, dass der gewichtete Prozentwert 31% ist...

Der errechnete Prozentwert für den Service (31%) wird dann gegen die Flatter-Schwellwerte verglichen, um zu sehen, was passiert:

- wenn der Service bisher *nicht* flattered und 31% *gleich oder größer* als der hohe Flatter-Schwellwert ist, nimmt Icinga an, dass der Service gerade angefangen hat zu flattern.
- wenn der Service *bereits* flattered und 31% *unter* dem niedrigen Flatter-Schwellwert liegt, nimmt Icinga an, dass der Service gerade aufgehört hat zu flattern.

wenn keine der beiden Bedingungen zutrifft, dann macht die Flatter-Erkennungslogik nichts weiteres mit dem Service, da er entweder (noch) nicht flattert oder bereits flattert.

Flatter-Erkennung für Services

Icinga prüft jedes Mal, wenn der Service geprüft wird (egal ob aktiv oder passiv), ob ein Service flattert.

Die Flatter-Erkennungslogik für Services arbeitet wie in dem obigen Beispiel beschrieben.

Flatter-Erkennung für Hosts

Host-Flatter-Erkennung arbeitet in einer ähnlichen Weise wie die Service-Flatter-Erkennung, mit einem wichtigen Unterschied: Icinga wird versuchen zu prüfen, ob ein Host flattert, wenn:

- der Host geprüft wird (aktiv oder passiv)
- manchmal, wenn ein Service geprüft wird, der mit dem Host verbunden ist. Genauer gesagt, wenn wenigstens x der Zeit vergangen ist, seit die letzte Flatter-Erkennung durchgeführt wurde, wobei x dem Durchschnittsintervall aller Services entspricht, die mit dem Host verbunden sind.

Warum wird das gemacht? Bei Services wissen wir, dass die minimale Zeit zwischen zwei aufeinander folgenden Flatter-Erkennungs routinen gleich dem Service-Prüfintervall sein wird. Allerdings werden Sie Hosts wahrscheinlich nicht auf einer regelmäßigen Basis überwachen, so dass es kein Prüfintervall gibt, das in der Flatter-Erkennungslogik benutzt werden kann. Außerdem ist es sinnvoll, dass die Prüfung eines Service der Erkennung eines Host-Flatters dienen sollte. Services sind Attribute eines Hosts bzw. bezogen auf Dinge, die mit dem Host verbunden sind. Auf jeden Fall ist es die beste Methode, die Ethan Galstad gefunden hat, um festzulegen, wie oft die Flatter-Erkennung auf einem Host ausgeführt werden kann.

Flatter-Erkennungsschwellwerte

Icinga benutzt verschiedene Variablen, um die Schwellwert-Prozentsätze der Statusänderungen festzulegen, die es für die Flatter-Erkennung nutzt. Für Hosts und Services gibt es hohe und niedrige *globale* und *Host- und Service-spezifische* Schwellwerte, die Sie konfigurieren können. Icinga wird die globalen Schwellwerte für die Flatter-Erkennung nutzen, wenn Sie keine Host- oder Service-spezifischen Schwellwerte angegeben haben.

Die Tabelle unten zeigt die globalen und die Host- oder Service-spezifischen Variablen, die die verschiedenen Schwellwerte kontrollieren, die bei der Flatter-Erkennung benutzt werden.

Objekt-Typ	Globale Variable	Objekt-spezifische Variablen
Host	low_host_flap_threshold	low_flap_threshold
	high_host_flap_threshold	high_flap_threshold
Service	low_service_flap_threshold	low_flap_threshold
	high_service_flap_threshold	high_flap_threshold

Zustände, die für die Flatter-Erkennung benutzt werden

Normalerweise wird Icinga die Ergebnisse der letzten 21 Prüfungen eines Hosts oder Service verfolgen, unabhängig vom Prüfergebnis (Host-/Service-Zustand), um sie für die Flatter-Erkennungslogik zu benutzen.



Hinweis: Sie können durch die `flap_detection_options`-Direktive in Ihren Host- oder Service-Definitonen verschiedene Host- oder Service-Zustände von der Nutzung in der Flatter-Erkennungslogik ausschließen. Diese Direktive erlaubt Ihnen die Angabe, welche Host- oder Service-Zustände (z.B. "UP", "DOWN", "OK", "CRITICAL") Sie für die Flatter-Erkennung benutzen wollen. Wenn Sie diese Direktive nicht nutzen wollen, werden alle Host- und Service-Zustände in der Flatter-Erkennung benutzt.

Flatter-Behandlung

Wenn bei einem Service- oder Host das erste Mal Flattern erkannt wird, wird Icinga:

1. eine Meldung protokollieren, dass der Service oder Host flattert
2. einen nicht-permanenten Kommentar zum Host oder Service hinzufügen, dass er flattert
3. eine "flapping start"-Benachrichtigung für den Host oder Service an die betreffenden Kontakte versenden
4. andere Benachrichtigungen für den Service oder Host unterdrücken (das ist einer der Filter in der [Benachrichtigungslogik](#))

Wenn ein Service oder Host aufhört zu flattern, wird Icinga:

1. eine Meldung protokollieren, dass der Service oder Host nicht mehr flattert
2. den Kommentar löschen, der zum Service oder Host hinzugefügt wurde, als dieser anfing zu flattern
3. eine "flapping stop"-Benachrichtigung für den Host oder Service an die betreffenden Kontakte versenden
4. die Blockade von Benachrichtigungen für den Service oder Host entfernen (Benachrichtigungen sind nach wie vor an die normale [Benachrichtigungslogik](#) gebunden)

Aktivieren der Flatter-Erkennung

Um die Flatter-Erkennungsmöglichkeiten in Icinga zu aktivieren, müssen Sie folgendes tun:

- setzen Sie die [enable_flap_detection](#)-Direktive auf 1.
- setzen Sie die *flap_detection_enabled*-Direktive in Ihren Host- und Service-Definitionen auf 1.

Wenn Sie die Flatter-Erkennung auf einer globalen Ebene deaktivieren wollen, setzen Sie die [enable_flap_detection](#)-Direktive auf 0.

Wenn Sie die Flatter-Erkennung nur für einige Hosts oder Services deaktivieren wollen, nutzen Sie die *flap_detection_enabled*-Direktive in den Host- oder Service-Definitionen, um das zu tun.

[Zurück](#)[Nach oben](#)[Weiter](#)

Redundante und
Failover-Netzwerk-Überwachung

[Zum Anfang](#)

Benachrichtigungsescalationen

© 2009-2011 Icinga Development Team, <http://www.icinga.org>

**Benachrichtigungseskalationen**[Zurück](#)**Kapitel 7. Fortgeschrittene Themen**[Weiter](#)

Benachrichtigungseskalationen

Einführung



Icinga unterstützt optionale Eskalation von Kontakt-Benachrichtigungen für Hosts und Services. Eskalationen von Host- oder Service-Benachrichtigungen werden erreichen durch das Definieren von [Host-Eskalationen](#) bzw. [Service-Eskalationen](#) in Ihrer/Ihren [Objekt-Konfigurationsdatei\(en\)](#).



Anmerkung: Das Beispiel, das wir unten zeigen, benutzt Service-Eskalationsdefinitionen, aber Host-Eskalationen arbeiten genau so. Außer, dass sie für Hosts sind statt für Services. :-)

Wann werden Benachrichtigungen eskaliert?

Benachrichtigungen werden eskaliert, *wenn, und nur wenn* eine oder mehrere Eskalationsdefinitionen mit der aktuellen Benachrichtigung übereinstimmen, die gerade versandt wird. Wenn eine Host- oder Service-Benachrichtigung *keine* gültige Eskalationsdefinition hat, die auf sie zutrifft, dann wird die Benachrichtigung an die Kontaktgruppe(n) verschickt, die in der Hostgroup- oder Service-Definition angegeben wurde(n). Lassen Sie uns das untenstehende Beispiel betrachten:

```
define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 3
    last_notification   5
    notification_interval 90
    contact_groups     nt-admins, managers
}
define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 6
```

```

last_notification      10
notification_interval 60
contact_groups         nt-admins, managers, everyone
}

```

Beachten Sie, dass es "Lücken" in den Benachrichtigungs-Eskalationsdefinitionen gibt. Im Besonderen werden weder die Benachrichtigungen 1 und 2 von den Eskalationen behandelt noch die Benachrichtigungen über 10. Für die ersten beiden und die Benachrichtigungen über 10 werden die *Default*-Kontaktgruppen aus der Service-Definition benutzt. Bei allen Beispielen, die wir benutzen, nehmen wir an, dass die Default-Kontaktgruppe für die Service-Definition *nt-admins* lautet.

Kontaktgruppen

Beim Definieren von Benachrichtigungs-Eskalationen ist es wichtig zu wissen, dass alle Kontaktgruppen, die Mitglieder von "niedrigeren" Eskalationen (d.h. mit niedrigeren Benachrichtigungsnummern-Bereichen) sind, auch in den "höheren" Eskalationsdefinitionen enthalten sein sollen. Das sollte passieren, um sicherzustellen, dass jeder, der über ein Problem informiert wird, *weiterhin* informiert wird, wenn ein Problem eskaliert. Beispiel:

```

define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 3
    last_notification   5
    notification_interval 90
    contact_groups     nt-admins, managers
}
define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 6
    last_notification   0
    notification_interval 60
    contact_groups     nt-admins, managers, everyone
}

```

Die erste (oder "niedrigste") Eskalationsstufe umfasst die *nt-admins* und die *managers*-Kontaktgruppe. Die letzte (oder "höchste") umfasst die *nt-admins*, *managers* und *everyone*-Kontaktgruppen. Beachten Sie, dass die *nt-admins*-Kontaktgruppe in beiden Eskalationsdefinitionen enthalten ist. Dies passiert, damit sie weiterhin per Pager informiert werden, falls noch Probleme existieren, nachdem die ersten beiden Service-Benachrichtigungen versandt wurden. Die *managers*-Kontaktgruppe erscheint zuerst in der "niedrigen" Eskalationsdefinition - sie wird das erste Mal benachrichtigt, wenn die dritte Benachrichtigung versandt wird. Wir möchten, dass die *managers*-Gruppe weiterhin informiert wird, wenn das Problem nach der fünften Benachrichtigung noch existiert, also sind sie in der "höheren" Eskalationsdefinition enthalten.

Überlappende Eskalationsbereiche

Benachrichtigungs-Eskalationsdefinitionen können Benachrichtigungs-Bereiche haben, die überlappen. Nehmen Sie das folgende Beispiel:

```

define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 3
    last_notification   5
    notification_interval 20
    contact_groups     nt-admins, managers
}

```

```
define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 4
    last_notification   0
    notification_interval 30
    contact_groups     on-call-support
}
```

Im obigen Beispiel:

- die *nt-admins-* und *managers*-Kontaktgruppen werden bei der dritten Benachrichtigung informiert
- alle drei Kontaktgruppen werden bei der vierten und fünften Benachrichtigung informiert
- nur die *on-call-support*-Kontaktgruppe wird bei der sechsten (und höheren) Benachrichtigung informiert

Erholungsbenachrichtigungen

Erholungsbenachrichtigungen unterscheiden sich geringfügig von Problembenachrichtigungen, wenn es um Eskalationen geht. Nehmen Sie das folgende Beispiel:

```
define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 3
    last_notification   5
    notification_interval 20
    contact_groups     nt-admins,managers
}
define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 4
    last_notification   0
    notification_interval 30
    contact_groups     on-call-support
}
```

Falls nach drei Problembenachrichtigungen eine Erholungsbenachrichtigung für den Service versandt wird: wer wird informiert? Die Erholung ist eigentlich die vierte Benachrichtigung, die versandt wird. Allerdings ist der Eskalationscode intelligent genug zu erkennen, dass nur die Leute, die die dritte Problembenachrichtigung erhalten haben, auch über die Erholung informiert werden. In diesem Fall würden die *nt-admins-* und *managers*-Kontaktgruppen über die Erholung informiert werden.

Benachrichtigungsintervalle

Sie können die Häufigkeit, mit der eskalierte Benachrichtigungen für einen bestimmten Host oder Service versandt werden, mit der *notification_interval*-Option in der Hostgroup- oder Service-Eskalations-Definition ändern. Beispiel:

```
define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 3
    last_notification   5
    notification_interval 45
    contact_groups     nt-admins,managers
}
define serviceescalation{
```

```

host_name          webserver
service_description HTTP
first_notification 6
last_notification   0
notification_interval 60
contact_groups     nt-admins, managers, everyone
}

```

In diesem Beispiel sehen wir, dass das Default-Benachrichtigungsintervall für den Service auf 240 Minuten eingestellt ist (das ist der Wert in der Service-Definition). Wenn die Service-Benachrichtigung bei der dritten, vierten und fünften Benachrichtigung eskaliert, wird ein Intervall von 45 Minuten zwischen den Benachrichtigungen genutzt. Bei der sechsten und folgenden Benachrichtigungen ist das Benachrichtigungsintervall 60 Minuten, wie in der zweiten Eskalationsdefinition angegeben.

Nachdem es möglich ist, überlappende Eskalationsdefinitonen für eine bestimmte Hostgruppe oder einen Service zu haben, und der Tatsache, dass ein Host Mitglied von mehreren Hostgruppen sein kann, muss Icinga eine Entscheidung treffen, was zu tun ist, wenn die Benachrichtigungs-Intervalle von Eskalationsdefinitionen überlappen. In jedem Fall, wenn es mehrere gültige Eskalationsdefinitionen für eine bestimmte Benachrichtigung gibt, wird Icinga das kleinste Benachrichtigungs-Intervall wählen. Nehmen Sie das folgende Beispiel:

```

define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 3
    last_notification   5
    notification_interval 45
    contact_groups     nt-admins, managers
}
define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 4
    last_notification   0
    notification_interval 60
    contact_groups     nt-admins, managers, everyone
}

```

Wir sehen, dass die beiden Eskalationsdefinitionen bei der vierten und fünften Benachrichtigung überlappen. Bei diesen Benachrichtigungen wird Icinga ein Benachrichtigungsintervall von 45 Minuten benutzen, weil dies das kleinste Intervall aller vorhandenen gültigen Eskalationsdefinitionen für diese Benachrichtigungen ist.

Eine letzte Anmerkung zu Benachrichtigungsintervallen, die Intervalle von 0 behandelt. Ein Intervall von 0 bedeutet, dass Icinga lediglich eine Benachrichtigung für die erste gültige Benachrichtigung während der Eskalationsdefinition versendet. Alle folgenden Benachrichtigungen für die Hostgruppe oder den Service werden unterdrückt. Nehmen Sie dieses Beispiel:

```

define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 3
    last_notification   5
    notification_interval 45
    contact_groups     nt-admins, managers
}
define serviceescalation{
    host_name          webserver
    service_description HTTP
    first_notification 4

```

```

last_notification      6
notification_interval 0
contact_groups         nt-admins, managers, everyone
}
define serviceescalation{
host_name              webserver
service_description     HTTP
first_notification      7
last_notification       0
notification_interval   30
contact_groups          nt-admins, managers
}

```

In dem obigen Beispiel werden maximal vier Problembenachrichtigungen zu diesem Service versandt. Das ist so, weil das Benachrichtigungsintervall 0 in der zweiten Eskalationsdefinition angibt, dass nur eine Benachrichtigung versandt werden soll (beginnend mit der vierten und diese einschließend) und folgende Benachrichtigungen unterdrückt werden sollen. Deshalb hat die dritte Eskalationsdefinition keinerlei Auswirkungen, denn es wird nie mehr als vier Benachrichtigungen geben.

Zeitfenster-Beschränkungen

Unter normalen Umständen können Eskalationen zu jeder Zeit benutzt werden, zu der Benachrichtigungen für einen Host oder Service versandt werden. Dieses "Benachrichtigungs-Zeitfenster" ist festgelegt durch die *notification_period*-Direktive in der [Host](#)- oder [Service](#)-Definition.

Sie können optional Eskalationen durch die *escalation_period*-Direktive in der Host- oder Service-Eskalationsdefinition beschränken, so dass sie lediglich während bestimmter Zeitspannen benutzt werden. Wenn Sie die *escalation_period*-Direktive benutzen, um eine [Zeitspanne](#) zu definieren, während der die Eskalation benutzt werden kann, wird sie nur zu dieser Zeit benutzt. Wenn Sie keine *escalation_period*-Direktive angeben, kann die Eskalation zu jeder Zeit innerhalb des "Benachrichtigungs-Zeitfensters" des Hosts oder Service benutzt werden.



Anmerkung: eskalierte Benachrichtigungen unterliegen weiterhin den normalen Zeitbeschränkungen, die durch die *notification_period*-Direktive in einer Host- oder Service-Definition festgelegt wurden, so dass die Zeitspanne, die Sie in einer Eskalationsdefinition angeben, ein Teil des größeren "Benachrichtigungs-Zeitfensters" sein sollte.

Status-Beschränkungen

Wenn Sie die Eskalationsdefinition beschränken wollen, damit sie nur benutzt wird, während sich der Host oder Service in einem bestimmten Zustand befindet, so können Sie die *escalation_options*-Direktive in der Host- oder Service-Eskalationsdefinition benutzen. Wenn Sie die *escalation_options*-Direktive nicht verwenden, werden die Eskalationen in jedem Status der Hosts oder Services benutzt.

[Zurück](#)
[Nach oben](#)
[Weiter](#)

Erkennung und Behandlung von
Status-Flattern

[Zum Anfang](#)

Eskalations-Bedingung

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Eskalations-Bedingung

[Zurück](#)
[Kapitel 7. Fortgeschrittene Themen](#)
[Weiter](#)

Eskalations-Bedingung

Einleitung

Ab Icinga 1.0.1 ist ein Patch implementiert, so dass Sie nun eine Eskalations-Bedingung definieren können (ähnlich wie `escalation_options [w,u,c,r]`). Eine Eskalation mit einer definierten Bedingung wird nur dann eskalieren, wenn der aktuelle Zustand eines bestimmten Hosts/Services mit der angegebenen Bedingung übereinstimmt. Ein mögliches Anwendungsbeispiel könnte das folgende Szenario sein:

Stellen Sie sich zwei verschiedene Eskalationen für den selben Service `foo` vor. Eine soll nur dann eskalieren, wenn der Service `bar` OK ist, die andere soll eskalieren, wenn `bar` CRITICAL oder WARNING ist. Nun stellen Sie sich vor, dass `foo` der zentrale Service einer Firma ist und der Administrator sofort reagieren muss, wenn der Service down ist. `bar` könnte ein Service sein, der angibt, ob der Admin im Büro oder zu Hause ist. Die Eskalation würde wie folgt reagieren:

- wenn der Administrator im Büro ist, dann sende zuerst eine e-Mail, nach 5 Minuten dann eine SMS
- wenn der Administrator zu Hause ist, dann sende zuerst eine SMS und nach 30 Minuten eine zweite SMS an den Administrator sowie eine SMS an den Abteilungsleiter

Dies kann erreicht werden, ohne Icinga neu zu starten bzw. die Konfiguration neu zu laden.

Syntax

Die Direktive `escalation_condition` ist komplett optional und kann sowohl für Host- als auch für Service-Eskalationen definiert werden. Die Syntax lautet:

```
escalation_condition <condition> ( [ & / | ] <condition> )
```

wobei `<condition>` entweder `host hostname = [u,d,o]` oder `service hostname.service_description = [w,u,c,o]` ist.

Wie Sie sehen können, akzeptiert `escalation_condition` eine Liste von einer oder mehreren Bedingungen, die durch "&" (logisches UND) oder "|" (logisches ODER) getrennt sind. Die Bedeutungen von `[w,u,c,o,d]` unterscheiden sich geringfügig von denen, die bei `escalation_options` benutzt werden:

- w = WARNING
- u = UNKNOWN
- c = CRITICAL
- o = OK für Services oder UP für hosts (man könnte an ONLINE denken)
- d = Down für hosts

Beispiel

```
define serviceescalation {
    host_name           localhost
    service_description HTTP
    first_notification 5
    contact_groups      admins, managers
    escalation_condition host linux=d | service linux.SSH=w,c
}
```

Diese Beispiel-Eskalation würde eskalieren, wenn der HOST 'linux' DOWN ist oder der Service 'linux.SSH' WARNING oder CRITICAL.

[Thanks to: Vitali Voroth, DECOIT GmbH * <http://www.decoit.de>]

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Benachrichtigungseskalationen

[Zum Anfang](#)

Bereitschafts-Rotation

© 2009-2011 Icinga Development Team, <http://www.icinga.org>

**Bereitschafts-Rotation**[Zurück](#)[Kapitel 7. Fortgeschrittene Themen](#)[Weiter](#)**Bereitschafts-Rotation****Einführung**

Admins müssen oft genug Pager, Mobiltelefonanrufe usw. beantworten, wenn sie es am wenigsten gebrauchen können. Keiner mag es, morgens um 4 Uhr geweckt zu werden. Allerdings ist es oft besser, das Problem mitten in der Nacht zu lösen als den Zorn eines unglücklichen Chefs zu spüren, wenn Sie am nächsten Morgen um 9 Uhr ins Büro kommen.

Für die glücklichen Admins, die ein Team von Gurus haben, die die Verantwortlichkeiten bei der Beantwortung von Alarmen teilen können, gibt es oft Bereitschaftspläne. Mehrere Admins werden oft abwechselnd Benachrichtigungen an Wochenenden, Nächten, Urlauben usw. entgegennehmen.

Wir werden Ihnen zeigen, wie Sie [Zeitfenster](#)-Definitionen erstellen können, die die meisten Bereitschafts-Benachrichtigungen behandeln werden. Diese Definitionen werden keine menschlichen Dinge berücksichtigen, die unweigerlich auftreten werden (Admins, die sich krank melden, Tausch von Schichten, oder Pager, die ins Wasser fallen), aber sie werden es Ihnen erlauben, eine grundlegende Struktur in Ihre Aufteilung zu bringen, die für die meiste Zeit funktionieren wird.

Szenario 1: Urlaub und Wochenenden

Zwei Admins - John und Bob - sind verantwortlich für die Bearbeitung von Icinga-Alarmen. John erhält alle Benachrichtigungen an Wochentagen (und Nächten) - außer im Urlaub - und Bob erhält Benachrichtigungen während der Wochenenden und Urlaube. Glücklicher Bob. Hier

nun, wie Sie diese Art der Rotation mit Zeitfenstern definieren...

Definieren Sie zuerst ein Zeitfenster, das Bereiche für Urlaube enthält:

```
define timeperiod{
    name          holidays
    timeperiod_name
    january 1      00:00-24:00 ; Neujahr
    2008-03-23     00:00-24:00 ; Ostern (2008)
    2009-04-12     00:00-24:00 ; Ostern (2009)
    monday -1 may   00:00-24:00 ; Memorial Day (Letzter Montag im Mai)
    july 4          00:00-24:00 ; Unabhängigkeitstag
    monday 1 september 00:00-24:00 ; Labour Day (1. Montag im September)
    thursday 4 november 00:00-24:00 ; Thanksgiving (4. Donnerstag im November)
    december 25      00:00-24:00 ; Weihnachten
    december 31      17:00-24:00 ; Silvester (ab 17:00 Uhr)
}
```

Als nächstes definieren Sie ein Zeitfenster für Johns Bereitschaftszeiten, das die Wochentage und Nächte während der Woche enthält, aber die Daten/Zeiten im Urlaubs-Zeitfenster ausschließt:

```
define timeperiod{
    timeperiod_name john-oncall
    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
    exclude         holidays           ; Exclude holiday dates/times defined elsewhere
}
```

Sie können nun dieses Zeitfenster in Johns Kontaktdefinition referenzieren:

```
define contact{
    contact_name      john
    ...
    host_notification_period  john-oncall
    service_notification_period john-oncall
}
```

Definieren Sie ein neues Zeitfenster für Bobs Bereitschaftszeiten, das die Wochenenden und die Daten/Zeiten der o.g. holiday-Zeitfenster enthält:

```
define timeperiod{
    timeperiod_name bob-oncall
    friday          00:00-24:00
    saturday        00:00-24:00
    use             holidays           ; Also include holiday date/times defined elsewhere
}
```

Sie können nun auf dieses Zeitfenster in Bobs Kontaktdefinition referenzieren:

```
define contact{
    contact_name      bob
    ...
    host_notification_period  bob-oncall
    service_notification_period bob-oncall
}
```

Szenario 2: Abwechselnde Tage

In diesem Szenario wechseln sich John und Bob täglich mit der Bearbeitung von Alarmen ab - unabhängig davon, ob es sich um Wochenenden, Wochentage oder Urlaub handelt.

Definieren Sie ein Zeitfenster, wann John Benachrichtigungen erhalten soll. Angenommen, der heutige Tag ist der 1. August 2009 und John beginnt heute mit der Bearbeitung von Benachrichtigungen, dann würde die Definition wie folgt aussehen:

```
define timeperiod{
    timeperiod_name          john-oncall
    2009-08-01 / 2           00:00-24:00      ; Alle zwei Tage, beginnend am 1. August 2009
}
```

Nun definieren Sie ein Zeitfenster, wann Bob Benachrichtigungen erhalten soll. Bob erhält Benachrichtigungen an den Tagen, an denen John keine erhält, also beginnt seine erste Bereitschaft morgen (2. August 2009).

```
define timeperiod{
    timeperiod_name          bob-oncall
    2009-08-02 / 2           00:00-24:00      ; Alle zwei Tage, beginnend am 2. August 2009
}
```

Nun müssen Sie diese Zeitfenster-Definitionen in den Kontaktdefinitionen von John und Bob referenzieren.

```
define contact{
    contact_name             john
    ...
    host_notification_period john-oncall
    service_notification_period john-oncall
}

define contact{
    contact_name             bob
    ...
    host_notification_period bob-oncall
    service_notification_period bob-oncall
}
```

Szenario 3: Abwechselnde Wochen

In diesem Szenario wechseln sich John und Bob jede Woche mit der Bearbeitung von Alarmen ab. John bearbeitet Alarne von Montag bis Sonntag in der einen Woche und Bob bearbeitet Alarne in den nächsten sieben Tagen. Dies wiederholt sich immer wieder.

Definieren Sie ein Zeitfenster, wann John Benachrichtigungen erhalten soll. Angenommen, heute ist Montag, der 27. Juli 2009 und John bearbeitet Benachrichtigungen in dieser Woche (beginnend mit heute), würde die Definition wie folgt aussehen:

```
define timeperiod{
    timeperiod_name john-oncall
    2009-07-27 / 14 00:00-24:00      ; alle 14 days (zwei Wochen), beginnend am 27. Juli 2009
    2009-07-28 / 14 00:00-24:00      ; jeden zweiten Dienstag, beginnend am 28. Juli 2009
    2009-07-29 / 14 00:00-24:00      ; jeden zweiten Mittwoch, beginnend am 29. Juli 2009
    2009-07-30 / 14 00:00-24:00      ; jeden zweiten Donnerstag, beginnend am 30. Juli 2009
    2009-07-31 / 14 00:00-24:00      ; jeden zweiten Freitag, beginnend am 31. Juli 2009
    2009-08-01 / 14 00:00-24:00      ; jeden zweiten Samstag, beginnend am 1. August 2009
    2009-08-02 / 14 00:00-24:00      ; jeden zweiten Sonntag, beginnend am 2. August 2009
}
```

Nun definieren Sie ein Zeitfenster, wann Bob Benachrichtigungen erhalten soll. Bob erhält Benachrichtigungen in den Wochen, in denen John keine bekommt, also startet seine erste Bereitschaft am nächsten Montag (3. August 2009).

```
define timeperiod{
    timeperiod_name bob-oncall
    2007-08-03 / 14 00:00-24:00      ; Every 14 days (two weeks), starting Monday, August 3th, 2009
    2007-08-04 / 14 00:00-24:00      ; Every other Monday starting August 4th, 2009
    2007-08-05 / 14 00:00-24:00      ; Every other Tuesday starting August 5th, 2009
    2007-08-06 / 14 00:00-24:00      ; Every other Wednesday starting August 6th, 2009
    2007-08-07 / 14 00:00-24:00      ; Every other Thursday starting August 7th, 2009
    2007-08-08 / 14 00:00-24:00      ; Every other Friday starting August 8th, 2009
    2007-08-09 / 14 00:00-24:00      ; Every other Saturday starting August 9th, 2009
}
```

Nun müssen Sie diese Zeitfenster-Definitionen in den Kontaktdefinitionen von John und Bob referenzieren.

```
define contact{
    contact_name          john
    ...
    host_notification_period   john-oncall
    service_notification_period   john-oncall
}

define contact{
    contact_name          bob
    ...
    host_notification_period   bob-oncall
    service_notification_period   bob-oncall
}
```

Szenario 4: Urlaubstage

In diesem Szenario bearbeitet John Benachrichtigungen an allen Tagen außer an denen, an denen er frei hat. Er hat frei an einigen festen Tagen im Monat ebenso wie an einigen geplanten Urlaubszeiten. Bob bearbeitet Benachrichtigungen, wenn John Urlaub hat oder nicht im Büro ist.

Definieren Sie zuerst ein Zeitfenster, das die Bereiche für Johns Urlaubstage und freie Tage enthält:

```
define timeperiod{
    name                  john-out-of-office
    timeperiod_name       john-out-of-office
    day 15                00:00-24:00           ; 15. Tag jeden Monats
    day -1                00:00-24:00           ; Letzter Tag jeden Monats (28., 29., 30., oder 31.)
    day -2                00:00-24:00           ; Vorletzter Tag jeden Monats (27., 28., 29., oder 30.)
    january 2              00:00-24:00           ; 2. Januar jeden Jahres
    june 1 - july 5        00:00-24:00           ; Jaehrlicher Camping-Trip (1. Juni - 5. Juli)
    2009-11-01 - 2009-11-10 00:00-24:00           ; Urlaub auf den Virgin Islands (1.-10. November 2009)
}
```

Als nächstes definieren Sie ein Zeitfenster für Johns Bereitschaftszeiten, das die Daten/Zeiten im o.g. Zeitfenster ausschließt:

```
define timeperiod{
    timeperiod_name       john-oncall
    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
    exclude               john-out-of-office ; Exclude dates/times John is out
}
```

Sie können nun dieses Zeitfenster in Johns Kontaktdefinition referenzieren:

```
define contact{
    contact_name                john
    ...
    host_notification_period    john-oncall
    service_notification_period john-oncall
}
```

Definieren Sie ein neues Zeitfenster für Bobs Bereitschaftszeiten, das die Zeiten von Johns Abwesenheiten enthält:

```
define timeperiod{
    timeperiod_name             bob-oncall
    use                         john-out-of-office      ; Include holiday date/times that John is out
}
```

Sie können nun dieses Zeitfenster in Bobs Kontaktdefinition referenzieren:

```
define contact{
    contact_name                bob
    ...
    host_notification_period    bob-oncall
    service_notification_period bob-oncall
}
```

Andere Szenarien

Es gibt eine Menge von anderen Bereitschafts-Benachrichtigungs-Szenarien, die Sie haben könnten. Die Datumsausschluss-Direktive in den [Zeitfenster-Definitionen](#) ist in der Lage, die meisten Datums- und Zeitbereiche abzubilden, die Sie brauchen könnten, also betrachten Sie die verschiedenen Formate, die Sie benutzen können. Wenn Sie einen Fehler bei der Erstellung von Zeitfenster-Definitionen machen, dann sollten Sie darauf achten, jemand anderem mehr Bereitschaftszeit zu geben. :-)

[Zurück](#)
[Nach oben](#)
[Weiter](#)
[Eskalations-Bedingung](#)
[Zum Anfang](#)
[Service- und Host-Gruppen
überwachen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Service- und Host-Gruppen überwachen

[Zurück](#)
[Kapitel 7. Fortgeschrittene Themen](#)
[Weiter](#)

Service- und Host-Gruppen überwachen

Einführung

Einige Leute haben gefragt, wie man Gruppen (Cluster) von Hosts und Services überwacht, also möchten wir hier schreiben, wie man das macht. Es ist ziemlich geradeaus, also hoffentlich sind die Dinge einfach zu verstehen...

Zuerst benötigen wir eine Definition, was wir mit "Cluster" meinen. Der einfachste Weg, dies zu verstehen, ist mit einem Beispiel. Lassen Sie uns annehmen, Ihr Unternehmen hat fünf Hosts, die redundante DNS-Services für Ihr Unternehmen zur Verfügung stellt. Wenn einer ausfällt, ist das keine große Katastrophe, weil die verbleibenden Server weiterhin die Namensauflösung bereitstellen. Wenn Sie mit der Überwachung der Verfügbarkeit der DNS-Server betraut sind, werden Sie fünf Server überwachen wollen. Das ist das, was wir als *Service-Cluster* ansehen würden. Der Service-Cluster besteht aus fünf einzelnen DNS-Services, die Sie überwachen wollen. Obwohl Sie jeden einzelnen Service überwachen wollen, wird Ihr Hauptaugenmerk eher auf dem Gesamtstatus des DNS-Service-Clusters liegen als auf der Verfügbarkeit eines einzelnen Service.

Wenn Ihre Organisation eine Gruppe von Hosts hat, die eine Hochverfügbarkeitslösung darstellt, würden wir dies als *Host-Cluster* bezeichnen. Wenn ein bestimmter Host ausfällt, wird ein anderer einspringen, um die Aufgaben des ausgefallenen zu übernehmen. Als eine Randbemerkung: Sehen Sie sich das [High-Availability Linux Project](#) für Informationen zur Redundanz von Hosts und Services mit Linux an.

Angriffsplan

Es gibt mehrere Wege, wie Sie eventuell Service- oder Host-Gruppen überwachen können. Wir werden die Methode beschreiben, von der wir glauben, dass sie die Einfachste ist. Service- oder Host-Cluster überwachen umfasst zwei Dinge:

- überwachen einzelner Cluster-Elemente
- überwachen des Clusters als eine gesamte Einheit

Das Überwachen von einzelnen Host- oder Service-Cluster-Elementen ist einfacher als Sie denken. Eigentlich tun Sie es wahrscheinlich schon. Bei Service-Clustern sollten Sie sicherstellen, dass Sie jedes Service-Element des Clusters überwachen. Wenn Sie ein Cluster aus fünf DNS-Servern haben, dann stellen Sie sicher, dass Sie fünf einzelne Service-Definitionen haben (z.B. mit dem `check_dns`-Plugin). Bei Host-Clustern stellen Sie sicher, dass Sie entsprechende Host-Definitionen für jedes Mitglied des Clusters haben (Sie müssen auch mindestens einen Service auf jedem Host überwachen). **Wichtig:** Sie können die

Benachrichtigungen für die einzelnen Cluster-Elemente deaktivieren (Host- oder Service-Definitionen). Obwohl keine Benachrichtigungen für die einzelnen Elemente versandt werden, bekommen Sie trotzdem eine visuelle Anzeige des einzelnen Host- oder Service-Zustands in der [Status CGI](#). Das ist nützlich bei der genauen Erkennung der Quelle von Problemen im Cluster in der Zukunft.

Die Überwachung des gesamten Clusters kann mit Hilfe der bereits im Cache verfügbaren Ergebnisse der Cluster-Elemente erfolgen. Auch wenn Sie alle Elemente des Clusters erneut prüfen könnten, um den Cluster-Status zu ermitteln: warum sollten Sie Bandbreite und Ressourcen vergeuden, wenn bereits die Ergebnisse im Cache vorliegen? Wo werden die Ergebnisse abgelegt? Ergebnisse für Cluster-Elemente sind im [Status-File](#) zu finden (vorausgesetzt, dass Sie jedes Element überwachen). Das *check_cluster*-Plugin ist genau für den Zweck ausgelegt, um Host- und Service-Zustände im Status-File zu prüfen. **Wichtig:** Auch wenn Sie Benachrichtigungen für einzelne Elemente des Clusters nicht aktiviert haben, möchten Sie sie vielleicht für den Gesamtstatus des Clusters aktivieren.

Das *check_cluster*-Plugin benutzen

Das *check_cluster*-Plugin ist dafür ausgelegt, den Gesamtstatus eines Host- oder Service-Clusters durch die Prüfung der Statusinformationen jedes einzelnen Host- oder Service-Cluster-Elements zu ermitteln.

noch mehr... Das *check_cluster*-Plugin finden Sie im contrib-Verzeichnis der Nagios-Plugins unter <http://sourceforge.net/projects/nagiosplug/>.

Service-Cluster überwachen

Nehmen wir an, dass Sie drei DNS-Server haben, die redundante Dienste in Ihrem Netzwerk bereitstellen. Zuerst müssen Sie jeden einzelnen DNS-Server überwachen, bevor Sie sie als Cluster überwachen können. Wir nehmen an, dass Sie bereits drei einzelne Services haben (die alle "DNS Service" heißen), die mit Ihren DNS-Hosts verbunden sind ("host1", "host2" und "host3" genannt).

Um die Services als einen Cluster zu überwachen, müssen Sie einen neuen "Cluster"-Service erstellen. Bevor Sie das tun, sollten Sie ein Service-Cluster-Prüfbefehl konfigurieren. Lassen Sie uns annehmen, dass Sie einen Befehl namens *check_service_cluster* wie folgt definieren:

```
define command{
    command_name      check_service_cluster
    command_line      /usr/local/icinga/libexec/check_cluster --service -l $ARG1$ -w $ARG2$ -c $ARG3$ -d $ARG4$
}
```

Nun müssen Sie den "Cluster"-Service erstellen und den *check_service_cluster*-Befehl benutzen, den Sie gerade als Cluster-Prüfbefehl erstellt haben. Das folgende Beispiel gibt einen Hinweis, wie das zu tun ist. Es generiert einen CRITICAL-Alarm, wenn zwei oder mehr Services im Cluster in einem nicht-OK-Zustand sind und einen WARNING-Alarm, wenn nur einer der Services in einem nicht-OK-Zustand ist. Wenn jedes der einzelnen Service-Mitglieder des Clusters OK sind, wird auch die Cluster-Prüfung einen OK-Status zurückliefern.

```
define service{
    ...
    check_command  check_service_cluster!"DNS Cluster"!0!1!$SERVICESTATEID:host1:DNS Service$,,$SERVICESTATEID:host2:DNS Service$,,$SERVICESTATEID:host3:DNS Service$
    ...
}
```

Es ist wichtig anzumerken, dass wir eine Komma-separierte Liste von *on-demand* Service-Zustands-Makros an das \$ARG4\$-Makro des Cluster-Prüfbefehls übergeben. Das ist wichtig! Icinga wird diese On-Demand-Makros mit den aktuellen Service-Status-IDs (numerischen Werten statt Zeichenketten) der einzelnen Mitglieder des Clusters füllen.

Host-Cluster überwachen

Host-Cluster zu überwachen ist ziemlich ähnlich zur Überwachung von Service-Clustern. Offenkundig besteht der Hauptunterschied darin, dass Hosts überwacht werden und nicht Services. Um den Status eines Host-Clusters zu überwachen, müssen Sie einen Service definieren, der das *check_cluster*-Plugin benutzt. Der Service sollte *nicht* mit einem der Hosts im Cluster verbunden werden, weil dies Probleme mit Benachrichtigungen für den Cluster erzeugt, wenn der Host "down" geht. Eine gute Idee könnte es sein, den Service mit dem Host zu verbinden, auf dem Icinga läuft. Wenn der Host, auf dem Icinga läuft, "down" geht, dann funktioniert auch Icinga nicht mehr und dann können Sie auch nichts mehr tun (es sei denn, Sie hätten eine [redundante Host-Überwachung](#) eingerichtet)...

Wie auch immer, lassen Sie uns annehmen, dass Sie einen *check_host_cluster*-Befehl wie folgt definiert haben:

```
define command{
    command_name      check_host_cluster
    command_line      /usr/local/icinga/libexec/check_cluster --host -l $ARG1$ -w $ARG2$ -c $ARG3$ -d $ARG4$
}
```

Sagen wir, dass Sie drei Hosts ("host1", "host2" und "host3" genannt) in Ihrem Host-Cluster haben. Wenn Icinga einen WARNING-Alarm generieren soll, wenn einer der Host im Cluster nicht UP ist bzw. einen CRITICAL-Alarm, wenn zwei oder mehr Hosts nicht UP sind, dann sollte der Service, um das Host-Cluster zu überwachen, ungefähr so aussehen:

```
define service{
    ...
    check_command  check_host_cluster!"Super Host Cluster"!0!1;$HOSTSTATEID:host1$,HOSTSTATEID:host2$,HOSTSTATEID:host3$
    ...
}
```

Es ist wichtig anzumerken, dass wir eine Komma-separierte Liste von *on-demand* Host-Zustands-Makros an das \$ARG4\$-Makro des Cluster-Prüfbefehls übergeben. Das ist wichtig! Icinga wird diese On-Demand-Makros mit den aktuellen Host-Status-IDs (numerischen Werten statt Zeichenketten) der einzelnen Mitglieder des Clusters füllen.

Das war's! Icinga wird regelmäßig den Status des Host-Clusters prüfen und Benachrichtigungen an Sie versenden, wenn der Status nicht OK ist (vorausgesetzt, dass Sie Benachrichtigungen für den Service aktiviert haben). Beachten Sie, dass Sie höchstwahrscheinlich die Benachrichtigungen in den einzelnen Host-Definitionen deaktivieren werden, wenn der Host "down" geht. Denken Sie daran, dass Sie der Status der einzelnen Hosts weniger interessiert als der Gesamtstatus des Clusters. Abhängig von Ihrem Netzwerk-Layout und von dem, was Sie erreichen wollen, möchten Sie vielleicht die Benachrichtigungen für UNREACHABLE-Zustände bei den Host-Definitionen aktiviert lassen.

[Zurück](#)
[Nach oben](#)
[Weiter](#)
[Bereitschafts-Rotation](#)
[Zum Anfang](#)
[Host- und Service-Abhängigkeiten](#)

**Host- und Service-Abhängigkeiten**[Zurück](#)[Kapitel 7. Fortgeschrittene Themen](#)[Weiter](#)

Host- und Service-Abhängigkeiten

Einführung

Service- und Hostabhängigkeiten sind ein fortgeschrittenes Feature von Icinga, dass Ihnen die Kontrolle über Hosts und Services erlaubt basierend auf dem Status von einem oder mehreren anderen Hosts oder Services. Wir werden erklären, wie Abhängigkeiten arbeiten, zusammen mit den Unterschieden zwischen Host- und Service-Abhängigkeiten.

Überblick Service-Abhängigkeiten

Es gibt ein paar Dinge, die Sie über Service-Abhängigkeiten wissen sollten:

1. ein Service kann von einem oder mehreren Services abhängig sein
2. ein Service kann von Services abhängig sein, die nicht mit dem gleichen Host verbunden sind
3. Service-Abhängigkeiten werden nicht vererbt (solange es nicht explizit konfiguriert wurde)
4. Service-Abhängigkeiten können benutzt werden, um Service-Prüfungen auszuführen und Service-Benachrichtigungen können unter verschiedenen Umständen unterdrückt werden (OK, WARNING, UNKNOWN und/oder CRITICAL-Zustände)
5. Service-Abhängigkeiten sind ggf. nur während bestimmter [Zeitfenster](#) gültig

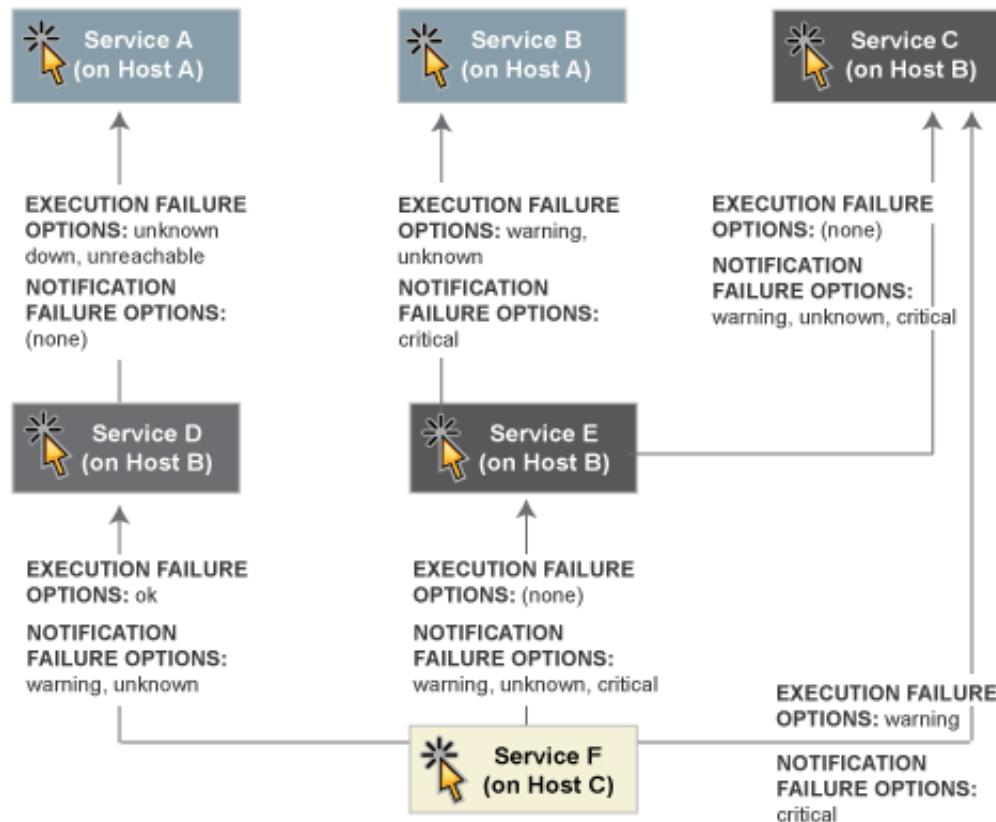
Service-Abhängigkeiten definieren

Zuerst die Grundlagen. Sie erstellen Service-Abhängigkeiten durch Hinzufügen von [Service-Abhängigkeitsdefinitionen](#) in der/n [Objekt-Konfigurationsdatei\(en\)](#). In jeder Definition geben Sie den *abhängigen* Service an, den Service, von dem sie *abhangen* und die Kriterien (falls vorhanden), die die Ausführung und Benachrichtungsabhängigkeiten fehlschlagen lassen (diese werden später beschrieben).

Sie können mehrere Abhängigkeiten für einen bestimmten Service erstellen, aber Sie müssen eine eigene Service-Abhängigkeitsdefinition anlegen für jede Abhängigkeit, die Sie erstellen.

Beispiel Service-Abhängigkeiten

Das folgende Bild zeigt ein beispielhaftes Logik-Layout von Service-Benachrichtigungen und Ausführungsabhängigkeiten. Verschiedene Services sind abhängig von anderen Services bzgl. Benachrichtigungen und Prüfausführung.



In diesem Beispiel wurde die Abhangigkeitsdefinition fur *Service F* auf *Host C* wie folgt aussehen:

```

define servicedependency{
    host_name          Host B
    service_description Service D
    dependent_host_name Host C
    dependent_service_description Service F
    execution_failure_criteria o
    notification_failure_criteria w,u
}
define servicedependency{
    host_name          Host B
    service_description Service E
    dependent_host_name Host C
    dependent_service_description Service F
    execution_failure_criteria n
    notification_failure_criteria w,u,c
}
define servicedependency{
    host_name          Host B
    service_description Service C
    dependent_host_name Host C
    dependent_service_description Service F
    execution_failure_criteria w
    notification_failure_criteria c
}
  
```

Die anderen im obigen Bild gezeigten Abhangigkeitsdefinitionen wurden wie folgt definiert:

```

define servicedependency{
    host_name          Host A
    service_description Service A
    dependent_host_name Host B
    dependent_service_description Service D
    execution_failure_criteria u
}
  
```

```

        notification_failure_criteria    n
    }
define servicedependency{
    host_name                      Host A
    service_description            Service B
    dependent_host_name           Host B
    dependent_service_description Service E
    execution_failure_criteria   w,u
    notification_failure_criteria c
}
define servicedependency{
    host_name                      Host B
    service_description            Service C
    dependent_host_name           Host B
    dependent_service_description Service E
    execution_failure_criteria   n
    notification_failure_criteria w,u,c
}

```

Wie Service-Abhangigkeiten getestet werden

Bevor Icinga eine Service-Prufung ausfuhrt oder Benachrichtigungen fur einen Service versendet, wird es prufen, ob der Service irgendwelche Abhangigkeiten hat. Wenn es keine Abhangigkeiten gibt, wird die Prufung ausgefuhrt oder die Benachrichtigung versandt, wie es sein sollte. Falls der Service ein oder mehrere Abhangigkeiten *hat*, wird Icinga jeden Abhangigkeitseintrag wie folgt prufen:

1. Icinga erhalt den aktuellen Status * des Services, von dem der Eintrag *abhangig* ist.
2. Icinga vergleicht den Status des Services, von dem der Eintrag *abhangig* ist, gegen die Ausführungs- oder Benachrichtigungsfehleroptionen in der Abhangigkeitsdefinition (je nachdem, welche zu dieser Zeit relevant ist).
3. wenn der aktuelle Status des Services, von dem der Eintrag *abhangig* ist, mit einer der Fehleroptionen ubereinstimmt, dann wird die Abhangigkeit als fehlerhaft angesehen und Icinga verlasst die Abhangigkeits-Prfschleife.
4. wenn der aktuelle Status des Services, von dem der Eintrag *abhangig* ist, nicht mit einer der Fehleroptionen ubereinstimmt, dann wird die Abhangigkeit als korrekt durchlaufen angesehen und Icinga wird fortfahren und den nachsten Abhangigkeitseintrag prufen.

Dieser Ablauf wird ausgefuhrt, bis entweder alle Abhangigkeiten fur diesen Service gepruft wurden oder eine Abhangigkeitsprufung fehlschagt.

 Anmerkung: * Bitte beachten Sie, dass Icinga per Default den aktuellsten **Hard-Status** des/r Services benutzt, von dem der Eintrag *abhangig* ist, wenn es die Abhangigkeitsprufungen durchfuhrt. Wenn Icinga den aktuellsten Status des/r Services benutzen soll (egal, ob es sich um einen Hard- oder Soft-Zustand handelt), dann aktivieren Sie die **soft_state_dependencies**-Option.

Ausführungsabhangigkeiten

Ausführungsabhangigkeiten werden benutzt, um einzuschrnen, wann **aktive Prufungen** eines Service ausgefuhrt werden konnen. **Passive Prufungen** werden durch Ausführungsabhangigkeiten nicht eingeschrnetzt.

Wenn *alle* der Ausführungsabhangigkeitstests fur den Service *erfolgreich* durchlaufen wurden, wird Icinga die Prufung fur den Service durchfuhren, wie es das normal tun wurde. Wenn auch nur einer der Ausführungsabhangigkeiten fur einen Service fehlschagt, wird Icinga

vorergehend die Ausführung von Prufungen fur diesen (abhangigen) Service verhindern. Irgendwann in der Zukunft konnen die Ausführungsabhangigkeitstests fur den Service erfolgreich durchlaufen werden. Wenn dies geschieht, wird Icinga mit der Prufung des Service beginnen, wie es das normal tun wurde. Mehr Informationen uber die Logik der Prufungsplanung finden Sie [hier](#).

Im obigen Beispiel waren die Tests der Ausführungsabhangigkeiten fur **Service E** fehlgeschlagen, wenn **Service B** in einem WARNING- oder UNKNOWN-Zustand ist. Falls dies der Fall ist, wurde die Service-Prufung nicht ausgefuhrt und die Prufung wurde fr eine (mogliche) Ausführung zu einem spateren Zeitpunkt geplant.

Benachrichtigungsabhangigkeiten

Wenn *alle* der Benachrichtigungsabhangigkeitstests fur den Service *erfolgreich* durchlaufen wurden, wird Icinga Benachrichtigungen fur den Service versenden, wie es das normal tun wurde. Wenn auch nur einer der Benachrichtigungsabhangigkeiten fur einen Service fehlschagt, wird Icinga vorergehend die Benachrichtigungen fur diesen (abhangigen) Service unterdrucken. Irgendwann in der Zukunft konnen die Benachrichtigungsabhangigkeitstests fur den Service erfolgreich durchlaufen werden. Wenn dies geschieht, wird Icinga mit dem Versand von Benachrichtigungen fur diesen Service beginnen, wie es das normal tun wurde. Mehr Informationen uber die Benachrichtigungslogik finden Sie [hier](#).

Im obigen Beispiel waren die Tests der Benachrichtigungsabhangigkeiten fur **Service F** fehlgeschlagen, wenn **Service C** in einem CRITICAL-Zustand *und/oder* **Service D** in einem WARNING- oder UNKNOWN-Zustand *und/oder* **Service E** in einem WARNING-, UNKNOWN- oder CRITICAL-Zustand ist. Falls dies der Fall ist, wurden keine Benachrichtigungen versandt werden.

Abhangigkeitsvererbung

Wie bereits erwahnt werden Service-Abhangigkeiten *nicht* per Default vererbt. Im obigen Beispiel sehen Sie, dass Service F von Service E abhangig ist. Trotzdem erbt er nicht automatisch die Abhangigkeiten von Service E zu Service B und Service C. Um Service F von Service C abhangig zu machen, mussen wir eine weitere Service-Abhangigkeitsdefinition hinzufugen. Es gibt keine Abhangigkeitsdefinition fur Service B, also ist Service F *nicht* abhangig von Service B.

Wenn Sie Service-Abhangigkeiten vererbbar machen *wollen*, mussen Sie die *inherits_parent*-Direktive in der [Service-Abhangigkeits](#)-Definition benutzen. Wenn diese Direktive aktiviert ist, bedeutet das, dass der Abhangige die Abhangigkeiten des Service erbt, von dem er abhangt (auch als Master-Service bezeichnet). Mit anderen Worten, wenn der Master-Service von anderen Services abhangt und eine von diesen Abhangigkeiten fehlschagt, wird auch dieser Service fehlschlagen.

Stellen Sie sich fur das obige Beispiel vor, Sie mochten eine neue Abhangigkeit fur Service F hinzufugen, um ihn von Service A abhangig zu machen. Sie konnen eine neue Abhangigkeitsdefinition erzeugen, die Service F als den *abhangigen* Service und Service A als den *Master-Service* angibt (d.h. der Service, auf den F *angewiesen* ist). Sie konnen alternativ die Abhangigkeitsdefinition der Services D und F verndern, die dann wie folgt aussehen:

```
define servicedependency{
    host_name                  Host B
    service_description         Service D
    dependent_host_name        Host C
    dependent_service_description Service F
    execution_failure_criteria o
    notification_failure_criteria n
    inherits_parent             1
}
```

Weil die `inherits_parent`-Direktive aktiviert ist, werden die Abhängigkeiten zwischen den Services A und D getestet, wenn die Abhängigkeiten zwischen den Services F und D getestet werden.

Abhängigkeiten können mehrere Vererbungsebenen haben. Wenn bei der Abhängigkeitsdefinition zwischen A und D die `inherits_parent`-Direktive aktiviert ist und Service A von einem anderen Service abhängig ist (sagen wir Service G), dann wäre Service F von den Services D, A und G abhängig (jeder mit möglicherweise unterschiedlichen Kriterien).

Host-Abhängigkeiten

Wie Sie vielleicht erwarten, arbeiten Host-Abhängigkeiten in einer ähnlichen Weise wie Service-Abhängigkeiten. Der Unterschied ist, dass sie für Hosts gelten und nicht für Services.



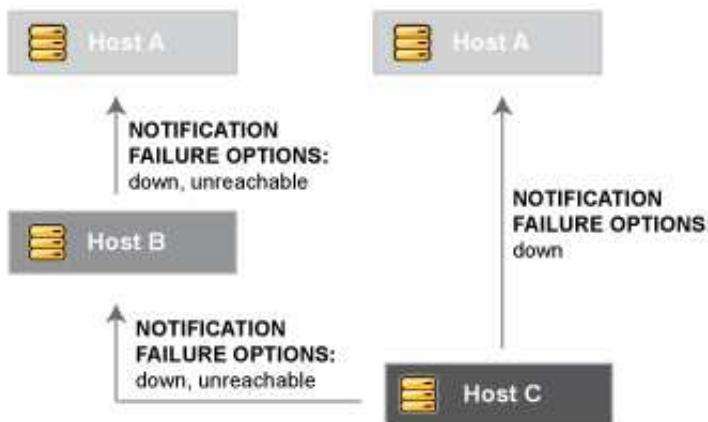
Hinweis: Verwechseln Sie Host-Abhängigkeiten nicht mit Eltern/Kind-Beziehungen. Sie sollten in den meisten Fällen Eltern/Kind-Beziehungen (mit Hilfe der `parents`-Direktive in den Host-Definitionen) benutzen statt Host-Abhängigkeiten. Eine Beschreibung, wie Eltern/Kind-Beziehungen arbeiten, finden Sie in der Dokumentation zur [Netzwerkerreichbarkeit](#).

Hier die Grundlagen zu Host-Abhängigkeiten:

1. ein Host kann von einem oder mehreren Hosts abhängig sein
2. Host-Abhängigkeiten werden nicht vererbt (solange es nicht explizit konfiguriert wurde)
3. Host-Abhängigkeiten können genutzt werden, um Host-Prüfungen auszuführen und Host-Benachrichtigungen unter bestimmten Umständen zu unterdrücken (UP, DOWN- und/oder UNREACHABLE-Zustände)
4. Host-Abhängigkeiten sind ggf. nur während bestimmter [Zeitfenster](#) gültig

Beispiel Host-Abhängigkeiten

Das folgende Bild zeigt ein Beispiel für das logische Layout von Benachrichtigungsabhängigkeiten. Verschiedene Hosts sind bzgl. Benachrichtigungen abhängig von anderen Hosts.



Im obigen Beispiel würden die Abhängigkeitsdefinitionen für *Host C* wie folgt aussehen:

```

define hostdependency{
    host_name                      Host A
    dependent_host_name            Host C
    notification_failure_criteria d
}
define hostdependency{
    host_name                      Host B
    dependent_host_name            Host C
    notification_failure_criteria d,u
}

```

Wie bei Service-Abhangigkeiten werden Host-Abhangigkeiten nicht vererbt. Im Beispielbild sehen Sie, dass Host C nicht die Host-Abhangigkeiten von Host B erbt. Um Host C von Host A abhangig zu machen, muss eine neue Host-Abhangigkeitsdefinition erstellt werden.

Host-Benachrichtigungsabhangigkeiten arbeiten in einer ahnlichen Weise wie Service-Benachrichtigungsabhangigkeiten. Wenn *alle* der Benachrichtigungsabhangigkeitstests fur den Host *erfolgreich* durchlaufen wurden, wird Icinga Benachrichtigungen fur den Host versenden, wie es das normal tun wurde. Wenn auch nur einer der Benachrichtigungsabhangigkeiten fur einen Host fehlschagt, wird Icinga vorergehend die Benachrichtigungen fur diesen (abhangigen) Host unterdrucken. Irgendwann in der Zukunft konnen die Benachrichtigungsabhangigkeitstests fur den Host erfolgreich durchlaufen werden. Wenn dies geschieht, wird Icinga mit dem Versand von Benachrichtigungen fur diesen Host beginnen, wie es das normal tun wurde. Mehr Informationen uber die Benachrichtigungslogik finden Sie [hier](#).

[Zuruck](#)
[Nach oben](#)
[Weiter](#)
[Service- und Host-Gruppen
uberwachen](#)
[Zum Anfang](#)
[Status Stalking](#)

 2009-2011 Icinga Development Team, <http://www.icinga.org>



Status Stalking

[Zurück](#)

Kapitel 7. Fortgeschrittene Themen

[Weiter](#)

Status Stalking

Einführung

Statusverfolgung ("state stalking") ist ein Feature, welches wahrscheinlich von den meisten Benutzern nicht eingesetzt wird. Wenn es aktiviert ist, erlaubt es Ihnen die Protokollierung von Änderungen bei Service- und Host-Prüfungen, selbst wenn sich der Zustand von Host oder Service nicht ändert. Wenn die Verfolgung für einen bestimmten Host oder Service aktiviert ist, wird Icinga den Host oder Service sehr genau beobachten und jede Änderung protokollieren, die es in der Ausgabe der Prüfergebnisse sieht. Wie Sie sehen werden, kann es sehr hilfreich bei der späteren Analyse der Log-Files sein.

Wie funktioniert es?

Unter normalen Umständen wird das Ergebnis einer Host- oder Service-Prüfung nur protokolliert, wenn der Host oder Service seit der letzten Prüfung seinen Zustand geändert hat. Es gibt wenige Ausnahmen, aber normalerweise ist das die Regel.

Wenn Sie die Verfolgung für einen oder mehrere Zustände eines bestimmten Hosts oder Service aktivieren, wird Icinga die Ergebnisse der Host- oder Service-Prüfung protokollieren, wenn sich die Ausgaben der Prüfung von den Ausgaben der letzten Prüfung unterscheiden. Nehmen Sie das folgende Beispiel von acht aufeinander folgenden Prüfungen eines Service:

Service Check #:	Service State:	Service Check Output:	Logged Normally	Logged With Stalking
x	OK	RAID array optimal	-	-
x+1	OK	RAID array optimal	-	-
x+2	WARNING	RAID array degraded (1 drive bad, 1 hot spare rebuilding)	✓	✓
x+3	CRITICAL	RAID array degraded (2 drives bad, 1 host spare online, 1 hot spare rebuilding)	✓	✓
x+4	CRITICAL	RAID array degraded (3 drives bad, 2 hot spares online)	-	✓
x+5	CRITICAL	RAID array failed	-	✓
x+6	CRITICAL	RAID array failed	-	-
x+7	CRITICAL	RAID array failed	-	-

Bei dieser Reihenfolge von Prüfungen würden Sie normalerweise nur zwei Einträge dieser Katastrophe sehen. Der erste würde bei Prüfung x+2 auftreten, wenn der Service von einem OK- in einen WARNING-Zustand wechselt. Der zweite Log-Eintrag würde bei Service-Prüfung x+3 auftreten, wenn der Service von einem WARNING- in einen CRITICAL-Zustand wechselt.

Aus welchem Grund auch immer möchten Sie die komplette Geschichte dieser Katastrophe in Ihren Log-Dateien sehen. Vielleicht, um Ihrem Manager zu zeigen, wie schnell die Situation außer Kontrolle geriet, vielleicht nur, um bei ein paar Bier in der Kneipe darüber zu lachen...

Wenn Sie die Verfolgung dieses Services für CRITICAL-Zustände aktiviert haben, hätten Sie zusätzlich zu den Ereignissen x+2 und x+3 auch noch x+4 und x+5 protokolliert. Warum ist das so? Mit aktiverter Verfolgung hätte Icinga die Ausgaben jeder Service-Prüfung untersucht, um Veränderungen zur Ausgabe der vorherigen Prüfung festzustellen. Wenn sich die Ausgaben unterscheiden und sich der Zustand des Service zwischen den beiden Prüfungen nicht verändert hat, würde die Ausgabe der neueren Prüfung protokolliert.

Ein ähnliches Beispiel für die Verfolgung könnte ein Service sein, der Ihren Web-Server prüft. Wenn das check_http-Plugin das erste Mal einen WARNING-Zustand wegen eines 404-Fehlers zurückliefert und bei folgenden Prüfungen einen WARNING-Zustand wegen eines nicht gefundenen Musters zurückliefert, dann möchten Sie das vielleicht wissen. Wenn Sie die Statusverfolgung für WARNING-Zustände nicht aktiviert haben, würde nur das erste WARNING-Ereignis (der 404-Fehler) protokolliert und Sie hätten keine Ahnung (beim Blick auf archivierte Protokolle), dass weitere WARNING-Zustände nicht auf dem 404-Fehler zurückzuführen sind, sondern dass bestimmte Textmuster nicht in der untersuchten Web-Seite zu finden waren.

Sollte ich die Verfolgung aktivieren?

Zuerst müssen Sie entscheiden, ob Sie wirklich Bedarf zur Untersuchung archivierter Protokolldaten haben, um die genaue Ursache eines Problems zu finden. Sie können entscheiden, dass Sie dieses Feature für ein paar Hosts oder Services brauchen, aber nicht für alle. Sie können auch feststellen, dass Sie die Verfolgung nur für einige Host- oder Service-Zustände brauchen, statt für alle. Sie könnten z.B. entscheiden, dass Sie die Verfolgung nur für die WARNING- und CRITICAL-Zustände eines Service benötigen, aber nicht OK- und UNKNOWN-Zustände.

Die Entscheidung, die Verfolgung für einen bestimmten Host oder Service zu aktivieren, hängt auch vom Plugin ab, das Sie zur Prüfung des Hosts oder Service einsetzen.

Wie aktiviere ich die Verfolgung?

Sie können die Verfolgung für Hosts und Services durch die *stalking_options*-Direktive in den [Host- und Service-Definitionen](#) aktivieren.

Wie unterschieden sich Verfolgung und "flüchtige Services"?

[Flüchtige Services](#) (volatile services) sind ähnlich, aber sie verursachen Benachrichtigungen und führen Eventhandler aus. Die Verfolgung dient lediglich der Protokollierung.

Risiken

Sie sollten beachten, dass es einige potenzielle Fallstricke bei der Aktivierung von Verfolgungen gibt. Sie beziehen sich alle auf die Berichtsfunktionen, die in verschiedenen [CGIs](#) zu finden sind (Histogramm, Alarmübersicht, usw.). Weil die Statusverfolgung zusätzliche Alarmeinträge schreibt, werden die Berichte eine erhöhte Anzahl von Alarmen anzeigen.

Als generelle Regel würden wir empfehlen, dass Sie die Verfolgung für Hosts und Services *nicht* aktivieren, ohne gründlich darüber nachgedacht zu haben. Auf jeden Fall ist sie da, wenn Sie sie brauchen.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Host- und
Service-Abhängigkeiten

[Zum Anfang](#)

Performance-Daten

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Performance-Daten

[Zurück](#)

Kapitel 7. Fortgeschrittene Themen

[Weiter](#)

Performance-Daten

Einführung

Icinga ist ausgelegt, dass [Plugins](#) neben den normalen Statusinformationen optional auch Performance-Daten zurückliefern können, die Sie außerdem zur weiteren Verarbeitung an externe Applikationen übergeben können. Eine Beschreibung der verschiedenen Typen von Performance-Daten wie auch Informationen darüber, wie diese Daten verarbeitet werden, finden Sie im Folgenden...

Type von Performance-Daten

Es gibt zwei grundlegende Kategorien von Performance-Daten, die von Icinga erhalten werden können:

1. Prüf-Performance-Daten
2. Plugin-Performance-Daten

Prüf-Performance-Daten sind interne Daten, die sich auf die aktuelle Ausführung einer Host- oder Service-Prüfung beziehen. Dies kann Dinge wie die Service-Prüfverzögerung enthalten (service check latency, d.h., wie viel Zeit von der geplanten Ausführung bis zu eigentlichen Ausführung verging) oder die Anzahl der Sekunden, die die Ausführung einer Host- oder Service-Prüfung dauerte. Dieser Typ von Performance-Daten ist für alle ausgeführten Prüfungen verfügbar. Die [\\$HOSTEXECUTIONTIME\\$](#)- und [\\$SERVICEEXECUTIONTIME\\$](#)-Makros können benutzt werden, um die Anzahl der Sekunden zu ermitteln, die eine Host- oder Service-Prüfung dauerte und die [\\$HOSTLATENCY\\$](#)- und [\\$SERVICELATENCY\\$](#)-Makros können zur Ermittlung der "Verspätung" einer regulär geplanten Host- oder Service-Prüfung genutzt werden.

Plugin-Performance-Daten sind externe Daten, die spezifisch für das Plugin sind, das die Host- oder Service-Prüfung ausführt. Plugin-spezifische Daten können Dinge wie Prozentsatz des Paketverlustes, freie Plattenplatz, Prozessor-Load, Anzahl der gegenwärtigen Benutzer usw. umfassen - generell jede Art von Metrik, die das Plugin misst, wenn es ausgeführt wird. Plugin-spezifische Performance-Daten sind optional und werden ggf. nicht von allen Plugins unterstützt. Plugin-spezifische Performance-Daten (falls verfügbar) werden durch die [\\$HOSTPERFDATA\\$](#)- und [\\$SERVICEPERFDATA\\$](#)-Makros bereit gestellt. Lesen Sie weiter, um mehr Informationen darüber zu erhalten, wie Plugins Performance-Daten an Icinga zur Bereitstellung durch die [\\$HOSTPERFDATA\\$](#)- und [\\$SERVICEPERFDATA\\$](#)-Makros zurückliefern können.

Plugin-Performance-Daten

Als Minimum müssen Icinga-Plugins eine einzelne Zeile mit menschlich lesbarem Text zurückliefern, die den Status eines Typs von Messdaten enthält. Zum Beispiel könnte das check_ping-Plugin eine Textzeile wie die folgende zurückliefern:

```
PING ok - Packet loss = 0%, RTA = 0.80 ms
```

Bei dieser einfachen Art von Ausgabe ist die gesamte Textzeile in den \$HOSTOUTPUT\$- oder \$SERVICEOUTPUT\$-[Makros](#) verfügbar (abhängig davon, ob dieses Plugin als Host- oder Service-Prüfung benutzt wurde).

Plugins können in ihrer Ausgabe optionale Performance-Daten zurückliefern, indem nach dem normalen, menschlich lesbaren Text ein Pipe-Symbol (!) folgt und danach eine Zeichenkette, die ein oder mehrere Performance-Daten-Metriken enthält. Lassen Sie uns das check_ping-Plugin als Beispiel nehmen und annehmen, dass es um die Ausgabe von Performance-Daten-Metriken für den Prozentsatz von Paketverlusten (percent packet loss) und durchschnittlicher Umlaufzeit (average round trip time) erweitert wurde. Die Beispielausgabe des Plugins könnte wie folgt aussehen:

```
PING ok - Packet loss = 0%, RTA = 0.80 ms | percent_packet_loss=0, rta=0.80
```

wenn Icinga dieses Plugin-Ausgabeformat sieht, wird es die Ausgabe in zwei Teile aufteilen:

1. alles vor dem Pipe-Symbol wird als "normale" Ausgabe des Plugins angesehen und im \$HOSTOUTPUT\$- oder \$SERVICEOUTPUT\$-Makro gespeichert
2. alles nach dem Pipe-Symbol wird als Plugin-spezifische Ausgabe angesehen und in den \$HOSTPERFDATA\$- oder \$SERVICEPERFDATA\$-Makros gespeichert.

Im obigen Beispiel würde das \$HOSTOUTPUT\$- oder das \$SERVICEOUTPUT\$-Makro "PING ok - Packet loss = 0%, RTA = 0.80 ms" enthalten (ohne Anführungszeichen) und das \$HOSTPERFDATA\$- oder das \$SERVICEPERFDATA\$-Makro würde "percent_packet_loss=0, rta=0.80" enthalten (ohne Anführungszeichen).

Icinga kann mehrere Zeilen Performance-Daten (ebenso wie normale Textausgaben) von Plugins entgegennehmen, wie in der [plugin API documentation](#) beschrieben.

 **Anmerkung:** der Icinga-Daemon verarbeitet Plugin-Performance-Daten nicht direkt, so dass es ihm egal ist, wie die Performance-Daten aussehen. Es gibt daher eigentlich keine Beschränkungen des Formats oder des Inhalts der Performance-Daten. Wenn Sie allerdings ein externes Addon benutzen, um die Performance-Daten zu verarbeiten (z.B. PNP oder PerfParse), erwartet das Addon die Performance-Daten möglicher Weise in einem bestimmten Format. Prüfen Sie die Dokumentation des Addon auf weitere Informationen.

Performance-Daten verarbeiten

Wenn Sie die Performance-Daten, die von den Plugins und in Icinga verfügbar sind, müssen Sie folgendes tun:

1. aktivieren Sie die [process_performance_data](#)-Option.
2. konfigurieren Sie Icinga so, dass Performance-Daten in Dateien geschrieben und/oder durch Befehle verarbeitet wird.

Lesen Sie weiter, um Informationen darüber zu erhalten, wie Performance-Daten durch das Schreiben in Dateien oder die Ausführung von Befehlen verarbeitet werden.

Performance-Daten verarbeiten durch Befehle

Der flexibelste Weg, um Performance-Daten zu verarbeiten, besteht darin, Icinga Befehle ausführen zu lassen (die Sie angeben), um die Daten zu verarbeiten oder sie umzulenken, damit sie später von externen Applikationen verarbeitet werden. Die Befehle, die Icinga ausführt, um Host- und Service-Performance-Daten zu verarbeiten, werden durch die [host_perfdata_command](#)- und [service_perfdata_command](#)-Optionen festgelegt.

Eine Beispiel-Befehlsdefinition, die Service-Prüf-Performance-Daten zur späteren Verarbeitung durch eine andere Applikation in eine Textdatei umleitet, finden Sie nachfolgend:

```
# ACHTUNG: diese Definition funktioniert NICHT mit PNP!
define command{
    command_name    store-service-perfdata
    command_line    /bin/echo -e "$LASTSERVICECHECKS:$HOSTNAME:$SERVICEDESC:$SERVICESTATE:$SERVICEATTEMPTS:$SERVICESTATETYPE:$SERVICEEXECUTIONTIME:$SERVICELATENCY:$SERVICEOUTPUT:$SERVICEPERFDATA" >> /usr/local/icinga/var/service-perfdata.dat
```



Hinweis: Diese Methode, obwohl flexibel, erzeugt einen relativ hohen CPU-Overhead. Wenn Sie Performance-Daten für viele Hosts und Services verarbeiten, dann ist es vielleicht besser, diese Daten in eine Datei zu schreiben. Diese Methode wird im nächsten Abschnitt beschrieben.

Performance-Daten in Dateien schreiben

Sie können Icinga mit Hilfe der [host_perfdata_file](#)- und [service_perfdata_file](#)-Optionen anweisen, die Host- und Service-Performance-Daten direkt in Textdateien auszugeben. Das Format, in dem Host- und Service-Performance-Daten in diese Dateien geschrieben wird, wird durch die [host_perfdata_file_template](#)- und [service_perfdata_file_template](#)-Optionen festgelegt.

Eine Beispiel-Dateiformatvorlage für Performance-Daten könnte wie folgt aussehen:

```
# ACHTUNG: diese Definition funktioniert NICHT mit PNP!
service_perfdata_file_template=[SERVICEPERFDATA]\$TIMET\$\$HOSTNAME\$\$SERVICEDESC\$\$SERVICEEXECUTIONTIME\$\$SERVICELATENCY\$\$SERVICEOUTPUT\$\$SERVICEPERFDATA$
```

Per Default werden die Textdateien im "append"-Modus ("anhängen") eröffnet. Wenn Sie den Modus auf "write" ("schreiben") oder "non-blocking read/write" ("nicht-blockierendes Lesen/Schreiben", nützlich beim Schreiben in Pipes) ändern, können Sie die [host_perfdata_file_mode](#)- und [service_perfdata_file_mode](#)-Optionen nutzen.

Zusätzlich können Sie Icinga mit den [host_perfdata_file_processing_command](#)- und [service_perfdata_file_processing_command](#)-Optionen anweisen, periodisch Befehle auszuführen, um regelmäßig die Performance-Daten-Dateien zu verarbeiten (z.B., um sie zu rotieren). Das Intervall, in dem diese Befehle ausgeführt werden, ist durch die [host_perfdata_file_processing_interval](#)- und [service_perfdata_file_processing_interval](#)-Optionen festgelegt.

[Zurück](#)
[Nach oben](#)
[Weiter](#)
[Status Stalking](#)
[Zum Anfang](#)
[Geplante Ausfallzeiten](#)



Geplante Ausfallzeiten

[Zurück](#)
[Kapitel 7. Fortgeschrittene Themen](#)
[Weiter](#)

Geplante Ausfallzeiten

Einführung

Icinga erlaubt Ihnen, Termine für geplante Ausfallzeiten (downtime) von Hosts und Services zu vergeben, die Sie überwachen. Das ist nützlich, wenn Sie bereits wissen, dass Sie einen Server für einen Upgrade oder etwas Ähnliches herunterfahren müssen.



Ausfallzeit einplanen

Sie können eine Ausfallzeit für Hosts und Services über das [extinfo CGI](#) einplanen (wenn Sie Host- oder Service-Informationen ansehen). Klicken Sie auf den Link "Schedule downtime for this host/service", um die Ausfallzeit zu planen.

Sobald Sie die Ausfallzeit für einen Host oder Service einplanen, wird Icinga für diesen Host oder Service einen Kommentar hinzufügen, der anzeigt, dass für diese Periode eine Ausfallzeit geplant ist. Wenn die Zeit vorüber ist, wird Icinga diesen Kommentar automatisch löschen. Cool, oder?

Feste und flexible Ausfallzeiten

Wenn Sie über das Web-Interface eine Ausfallzeit einplanen, werden Sie gefragt, ob sie fest oder flexibel sein soll. Hier eine Erklärung, wie sich "fest" und "flexibel" unterscheiden:

"Feste" Ausfallzeiten starten und stoppen genau zu den Zeiten, die Sie bei der Planung festgelegt haben. Okay, das war einfach genug...

"Flexible" Ausfallzeiten sind gedacht für Zeiten, wenn Sie wissen, dass ein Host oder Service für X Minuten (oder Stunden) nicht verfügbar sein wird, aber Sie nicht genau wissen, wann das sein wird. Wenn Sie flexible Ausfallzeiten planen, wird Icinga die geplante Ausfallzeit irgendwann zwischen den Start- und Endzeiten beginnen, die Sie angegeben haben. Die Ausfallzeit wird solange dauern, wie Sie das bei der Planung angegeben haben. Dabei wird angenommen, dass der Host oder Service, für den Sie eine flexible Ausfallzeit geplant haben, ausfällt (oder unerreichbar wird) oder zwischen der angegebenen Start- und Endezeit in einen nicht-OK-Zustand wechselt. Die Zeit, zu der der Host oder Service in einen Problemzustand wechselt, legt die Zeit fest, zu der Icinga tatsächlich die Ausfallzeit startet. Die Ausfallzeit wird die angegebene Zeitspanne dauern, auch wenn sich der Host oder Service vor der definierten Zeit erholt. Das wird aus gutem Grund getan. Wie wir alle wissen, denken Sie vielleicht, dass Sie ein Problem gelöst haben, aber müssen den Server doch noch zehnmal neu starten, bevor es wirklich funktioniert. Geschickt, oder?

ausgelöste Ausfallzeiten

Während des Planens von Host- oder Service-Ausfallzeiten haben Sie die Möglichkeit, sie zu "ausgelösten" Ausfallzeiten (triggered downtime) zu machen. Was ist eine ausgelöste Ausfallzeit, fragen Sie? Bei ausgelösten Ausfallzeiten wird der Start der Ausfallzeit durch den Start einer anderen geplanten Host- oder Service-Ausfallzeit ausgelöst. Dies ist sehr nützlich, wenn Sie Ausfallzeiten für eine große Zahl von Hosts oder Services planen und die Startzeit der Auszeit von der Startzeit eines anderen Ausfallzeiteintrags abhängt. Wenn Sie zum Beispiel eine flexible Ausfallzeit für einen bestimmten Host planen (weil er zur Wartung heruntergefahren wird), könnten Sie ausgelöste Ausfallzeiten für alle "Kinder" des Hosts planen.

Wie geplante Ausfallzeiten Benachrichtigungen beeinflussen

Wenn sich ein Host oder Service in einer Phase geplanter Ausfallzeit befindet, wird Icinga keine normalen Benachrichtigungen für den Host oder Service versenden. Allerdings wird es eine "DOWNTIMESTART"-Benachrichtigung für den Host oder Service versenden, die jeden Admin darüber informiert, dass sie nachfolgend keine Problemalarme erhalten werden.

Wenn die geplante Ausfallzeit vorbei ist, wird Icinga wieder normale Benachrichtigungen für den Host oder Service versenden. Eine "DOWNTIMEEND"-Benachrichtigung wird an die Admins versandt, dass die geplante Ausfallzeit vorüber ist und dass sie wieder normale Alarne erhalten werden.

Wenn die geplante Auszeit vorzeitig abgebrochen wird (bevor sie endet), wird eine "DOWNTIMECANCELLED"-Benachrichtigung an die betroffenen Admins versandt.

Überlappende geplante Ausfallzeiten

Ich mag es, dieses als das "Oh Mist, es funktioniert nicht"-Syndrom zu bezeichnen. Sie wissen, wovon wir sprechen. Sie fahren einen Server herunter, um einen "Routine"-Hardware-Upgrade zu machen, nur um später festzustellen, dass die OS-Treiber nicht funktionieren, das RAID-Array hochgegangen ist oder Laufwerkskopien fehlgeschlagen und Ihre Original-Platten jetzt nutzlos sind. Moral der Geschichte ist, dass jede Routinearbeit an einem Server durchaus drei- oder viermal länger dauern kann, als Sie ursprünglich geplant haben...

Nehmen wir das folgende Szenario:

1. Sie planen eine Auszeit für Host A an einem Montag von 19:30 Uhr bis 21:30 Uhr
2. Sie fahren den Server am Montag gegen 19:45 Uhr herunter, um einen Platten-Upgrade durchzuführen
3. nachdem Sie eineinhalb Stunden mit SCSI-Fehlern und Treiberinkompatibilitäten verschwendet haben, können Sie endlich den Server starten
4. um 21:15 Uhr stellen Sie fest, dass eine Ihrer Partitions nirgends auf der Platte zu finden ist
5. da Sie wissen, dass es eine lange Nacht wird, gehen Sie zurück und planen eine zusätzliche Auszeit für Host A von Montag 21:20 Uhr bis Dienstagmorgen 1:30 Uhr

Wenn Sie überlappende Ausfallzeiten für einen Host oder Service planen (in diesem Fall waren die Zeiten von 19:40 Uhr bis 21:30 Uhr und 21:20 bis 1:30 Uhr), wird Icinga warten, bis die letzte Periode geplanter Ausfallzeiten vorüber ist, bevor Benachrichtigungen zu diesem Host oder Service versandt werden. In diesem Beispiel werden Benachrichtigungen für Host A bis Dienstagmorgen 1:30 Uhr unterdrückt.

[Zurück](#)[Nach oben](#)[Weiter](#)[Performance-Daten](#)[Zum Anfang](#)[Benutzen des Embedded Perl
Interpreters](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Benutzen des Embedded Perl Interpreters

[Zurück](#)

Kapitel 7. Fortgeschrittene Themen

[Weiter](#)

Benutzen des Embedded Perl Interpreters

Einführung

Icinga kann für die Unterstützung eines eingebetteten Perl-Interpreters (embedded perl interpreter) kompiliert werden. Dies erlaubt es Icinga, Perl-Plugins effizienter als sonst auszuführen, also mag es interessant sein, wenn Sie sich viel auf Plugins verlassen, die in Perl geschrieben sind.

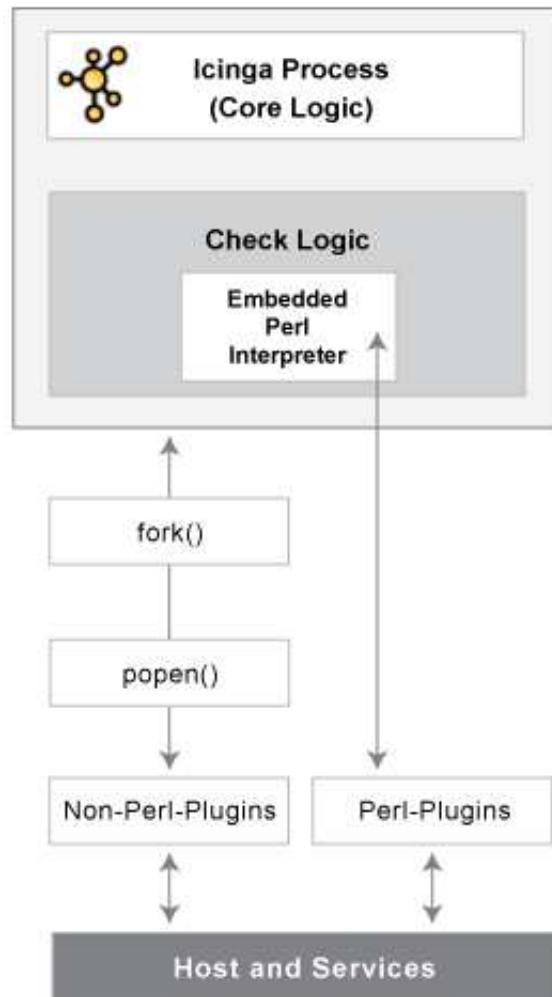
Ohne den eingebetteten Perl-Interpreter führt Icinga Perl- (und andere) Plugins durch "forking" und ausführen als einen externen Befehl aus. Wenn der eingebettete Perl-Interpreter benutzt wird, kann Icinga Perl-Plugins durch einen einfachen Library-Call ausführen.



Hinweis: Der Perl-Interpreter arbeitet mit allen Perl-Skripten, die Icinga ausführt - nicht nur Plugins. Diese Dokumentation behandelt den eingebetteten Perl-Interpreter in Verbindung mit Plugins für Host- und Service-Prüfungen, aber sie trifft genauso auf andere Arten von Perl-Skripten zu, die Sie vielleicht für andere Arten von Befehlen benutzen (z.B. Benachrichtigungs-Skripte, Eventhandler-Skripte usw.).

Stephen Davis hat den originalen eingebetteten Perl-Interpreter-Code vor einigen Jahren beigetragen. Stanley Hopcroft war die erste Person, die geholfen hat, den eingebetteten Perl-Interpreter-Code zu verbessern und die die Vor- und Nachteile bei der Benutzung kommentiert hat. Er hat auch verschiedene hilfreiche Hinweise zur Erstellung von Perl-Plugins gegeben, die sauber mit dem eingebetteten Interpreter arbeiten.

Es sollte angemerkt werden, dass sich "ePN" - wie in dieser Dokumentation benutzt - auf "embedded Perl Icinga" bezieht, oder wenn Sie das bevorzugen, Icinga mit eingebettetem Perl-Interpreter.



Vorteile

Einige Vorteile von ePN (embedded Perl Icinga) umfassen:

- Icinga wird viel weniger Zeit bei der Ausführung Ihrer Perl-Plugins verbringen, weil es nicht länger "fork"t, um das Plugin auszuführen (das Laden des Perl-Interpreters entfällt). Statt dessen führt es das Plugin durch einen Library-Call durch.
- es reduziert die Systembelastung durch Perl-Plugins und/oder erlaubt es Ihnen, mehr Prüfungen mit Perl-Plugins durchzuführen, als Ihnen das sonst möglich wäre. Mit anderen Worten haben Sie weniger Anreiz, Plugins in anderen Sprachen wie z.B. C/C++, oder Expect/TCL zu schreiben, die bei den Entwicklungszeiten eine Zehnerpotenz langsamer angesehen werden als Perl (wobei sie auch zehnmal schneller ablaufen, von TCL mal abgesehen).
- Wenn Sie kein C-Programmierer sind, dann können Sie trotzdem eine Menge mit Perl erledigen, ohne dass es Icinga viel langsamer macht. Beachten Sie, dass ePN Ihre Plugins nicht schneller macht (außer, dass es die Ladezeit eliminiert). Wenn Sie schnelle Plugins wollen, dann berücksichtigen Sie Perl XSUBs (XS) oder C, *nachdem* Sie sicher sind, dass Sie Ihr Perl getuned haben und dass Sie einen angemessenen Algorithmus haben (Benchmark.pm ist *unbezahlbar* für den Vergleich von Perl-Sprachelementen).
- ePN zu benutzen ist eine exzellente Gelegenheit, mehr über Perl zu lernen.

Nachteile

Die Nachteile von ePN (embedded Perl Icinga) sind ziemlich die gleichen wie bei Apache mod_perl (d.h. Apache mit einem eingebetteten Interpreter) verglichen mit einem schlanken Apache:

- Ein Perl-Programm, das *wunderbar* mit schlichtem Icinga arbeitet, muss *nicht* mit ePN funktionieren. Möglicherweise müssen Sie Ihre Plugins modifizieren, damit sie funktionieren.



Wichtig

Bitte beachten Sie, dass Ihr Plugin ggf. "Memory-Leaks" enthält, so dass es wichtig ist, dass Sie es mit new_mini_epn testen/verifizieren (das Sie im contrib-Verzeichnis finden).

- Perl-Plugins sind unter ePN schwieriger zu debuggen als unter schlichtem Icinga.
- Ihr ePN wird eine größere SIZE (Speichernutzung) haben als ein schlankes Icinga.
- Einige Perl-Konstrukte können nicht genutzt werden oder mögen sich anders verhalten als Sie das erwarten würden.
- Sie sollte sich bewusst sein, dass es 'mehr als einen Weg gibt, es zu tun' und ggf. einen Weg wählen, der weniger attraktiv oder offensichtlich ist.
- Sie werden mehr Perl-Know-How benötigen (aber nichts sehr esoterisches oder Zeug über Perl-Interna - außer, wenn Ihre Plugins XSUBS benutzen).

Benutzung des eingebetteten Perl-Interpreters

Wenn Sie den eingebetteten Perl-Interpreter benutzen wollen, um Ihre Perl-Plugins und Scripts auszuführen, dann lesen Sie hier, was Sie tun müssen:

1. Kompilieren Sie Icinga mit Unterstützung für den eingebetteten Perl-Interpreter (Anweisungen s.u.).
2. aktivieren Sie die `enable_embedded_perl`-Option in der Hauptkonfigurationsdatei.
3. setzen Sie die `use_embedded_perl_implicitly`-Option entsprechend Ihren Anforderungen. Diese Option legt fest, ob der Perl-Interpreter per Default für einzelne Perl-Plugins und Scripts benutzt werden sollte.
4. Optional sollten Sie bei verschiedenen Perl-Plugins und Scripts die Ausführung durch den eingebetteten Perl-Interpreter aktivieren oder deaktivieren. Das kann nützlich sein, wenn bestimmte Perl-Scripte Probleme bei der Ausführung mit dem Perl-Interpreter haben. Beachten Sie die Anweisungen weiter unten für mehr Informationen, wie das zu tun ist.

Icinga mit eingebettetem Perl kompilieren

Wenn Sie den eingebetteten Perl-Interpreter benutzen möchten, müssen Sie zuerst Icinga mit der Unterstützung dafür kompilieren. Um dies zu tun, starten Sie einfach das `configure`-Script zusätzlich mit der `-enable-embedded-perl` -Option. Wenn Sie aktivieren wollen, dass der Perl-Interpreter intern kompilierte Scripts in einem Cache ablegen soll, dann nutzen Sie die `--with-perlcache` -Option. Beispiel:

```
./configure --enable-embedded-perl --with-perlcache otheroptions...
```

Sobald Sie das `configure`-Script mit den neuen Optionen ausgeführt haben, müssen Sie Icinga erneut kompilieren.

Paket-Bauer können eine andere Option nutzen, um das Verzeichnis der `p1.pl` Datei anzugeben:

```
./configure --with-p1-file-dir=<path>
```

Plugin-spezifische Benutzung des Perl-Interpreters

Beginnend mit Icinga 1.6 können Sie angeben, welche Perl-Plugins oder Scripts mit dem eingebetteten Perl-Interpreter ablaufen sollen und welche nicht. Das ist besonders dann nützlich, wenn Sie Perl-Scripte haben, die nicht sauber mit dem Perl-Interpreter laufen.

Um Icinga *explizit* mitzuteilen, ob der Perl-Interpreter benutzt werden soll oder nicht, fügen Sie Ihrem Perl-Script/Plugin einen der folgenden Einträge hinzu..

Um Icinga mitzuteilen, den Perl-Interpreter für ein bestimmtes Script zu nutzen, fügen Sie dem Perl-Script diese Zeile hinzu:

```
# icinga: +epn
```

Um Icinga mitzuteilen, den Perl-Interpreter für ein bestimmtes Script NICHT zu nutzen, fügen Sie dem Perl-Script diese Zeile hinzu:

```
# icinga: -epn
```

Eine der beiden Zeilen muss innerhalb der ersten zehn Zeilen stehen, damit sie von Icinga erkannt wird.



Anmerkung

"icinga: +/- epn" wird seit Icinga 1.2.1 unterstützt. Vorher mussten Sie "nagios: +/-epn" benutzen, was aus Kompatibilitätsgründen weiterhin unterstützt wird.



Hinweis: Wenn Sie nicht *explizit* die oben genannte Methode nutzen, um Icinga mitzuteilen, den Perl-Interpreter für ein einzelnes Plugin zu nutzen, wird Icinga eine Entscheidung für Sie treffen. Dieser Entscheidungsprozess wird von der `use_embedded_perl_implicitly`-Variable kontrolliert. Wenn der Wert auf 1 gesetzt ist, werden alle Perl-Plugins/Scripts (bei denen nicht explizit der ePN aktiviert/deaktiviert ist) mit dem Perl-Interpreter ausgeführt. Wenn der Wert auf 0 gesetzt ist, werden sie NICHT mit dem Perl-Interpreter ausgeführt.

Plugins für die Nutzung mit Embedded Perl entwickeln

Informationen über die Entwicklung von Plugins zur Nutzung mit dem eingebetteten Perl-Interpreter finden Sie [hier](#).

[Zurück](#)[Nach oben](#)[Weiter](#)[Geplante Ausfallzeiten](#)[Zum Anfang](#)[Adaptive Überwachung](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Adaptive Überwachung

[Zurück](#)

Kapitel 7. Fortgeschrittene Themen

[Weiter](#)

Adaptive Überwachung

Einführung

Icinga erlaubt Ihnen, verschiedene Befehle und Host- und Service-Prüfattribute während der Laufzeit zu verändern. Wir bezeichnen diese Möglichkeit als "adaptive Überwachung". Bitte beachten Sie, dass diese adaptiven Überwachungs-Features in Icinga wahrscheinlich von 99% aller Benutzer nicht genutzt werden, aber sie erlauben Ihnen einige nette Dinge.

Was kann verändert werden?

Die folgenden Service-Prüfattribute können während der Laufzeit verändert werden:

- Prüfbefehl (und Befehlsparameter)
- Prüfintervall
- max. Prüfversuche
- Prüfzeitfenster
- Eventhandler-Befehl (und Befehlsparameter)

Die folgenden Host-Prüfattribute können während der Laufzeit verändert werden:

- Prüfbefehl (und Befehlsparameter)
- Prüfintervall
- max. Prüfversuche
- Prüfzeitfenster
- Eventhandler-Befehl (und Befehlsparameter)

Die folgenden globalen Attribute können während der Laufzeit verändert werden:

- Globaler Host-Eventhandler-Befehl (und Befehlsparameter)
- Globaler Service-Eventhandler-Befehl (und Befehlsparameter)

Externe Befehle für adaptive Überwachung

Um globale oder Host- bzw. Service-spezifische Attribute während der Laufzeit zu ändern, müssen Sie über das [external command file](#) den entsprechenden [externen Befehl](#) an Icinga senden. Eine vollständige Liste von externen Befehlen, die zur adaptiven Überwachung benutzt werden können, finden Sie in der [Liste der externen Befehle](#).



Anmerkungen:

- Bei der Änderung von Prüfbefehlen, Prüfzeitfenstern oder Eventhandler-Befehlen ist es wichtig anzumerken, dass die neuen Werte für diese Optionen vor dem Neustart von Icinga definiert werden müssen. Jede Anfrage, die einen Befehl oder ein Zeitfenster auf einen Wert ändert, der beim Start nicht definiert war, wird ignoriert.
- Sie können Befehlsparameter zusammen mit dem tatsächlichen Befehlsnamen angeben - trennen Sie einfach die einzelnen Parameter vom Befehlsnamen (und voneinander) durch Ausrufezeichen (!). Mehr Informationen, wie Parameter in Befehlsdefinitionen während der Laufzeit verarbeitet werden, finden Sie in der Dokumentation zu [Makros](#).

[Zurück](#)[Nach oben](#)[Weiter](#)

Benutzen des Embedded Perl
Interpreters

[Zum Anfang](#)

Vorausschauende
Abhängigkeitsprüfungen



Vorausschauende Abhängigkeitsprüfungen

[Zurück](#)
[Kapitel 7. Fortgeschrittene Themen](#)
[Weiter](#)

Vorausschauende Abhängigkeitsprüfungen

Einführung

Host- und Service-[Abhängigkeiten](#) können definiert werden, um Ihnen größere Kontrolle darüber zu geben, wann Prüfungen ausgeführt und wann Benachrichtigungen versandt werden. Da Abhängigkeiten benutzt werden, um grundlegende Aspekte des Überwachungsprozesses zu kontrollieren, ist es wichtig sicherzustellen, dass die Status-Informationen in der Abhängigkeitslogik so aktuell wie möglich sind.

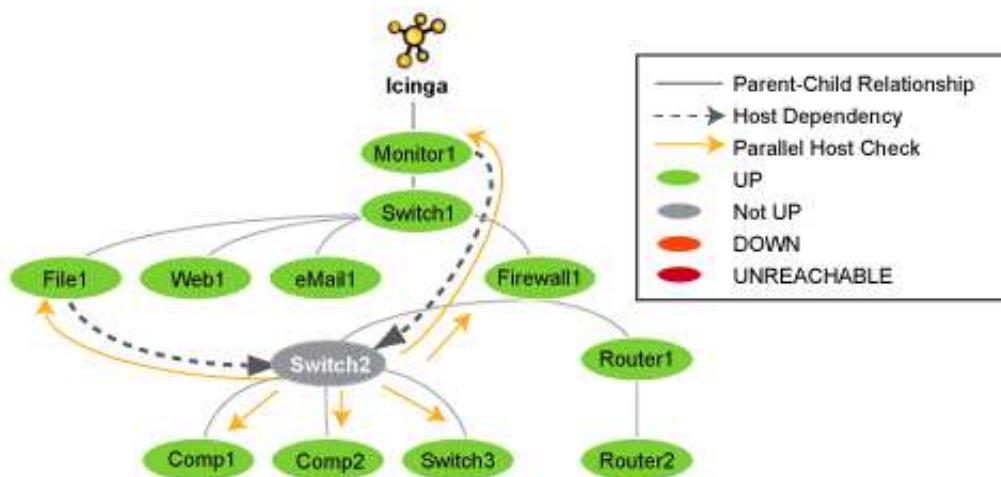
Icinga erlaubt Ihnen, vorausschauende Abhängigkeitsprüfungen für Hosts und Services zu aktivieren, um sicherzustellen, dass die Abhängigkeitslogik die aktuellsten Status-Informationen hat, wenn Entscheidungen darüber getroffen werden müssen, ob Benachrichtigungen verschickt werden oder um aktive Prüfungen für einen Host oder Service zu erlauben.

Wie arbeiten vorausschauende Prüfungen?

Das nachfolgende Bild zeigt ein einfaches Diagramm von Hosts, die von Icinga überwacht werden, zusammen mit ihren Eltern/Kindbeziehungen und Abhängigkeiten.

Der *Switch2*-Host in diesem Beispiel hat gerade den Status von UP in einen Problemzustand gewechselt. Icinga muss feststellen, ob der Host DOWN oder UNREACHABLE ist, also wird es parallele Prüfungen für die direkten Eltern (*Firewall1*) und Kinder (*Comp1*, *Comp2*, und *Switch3*) auslösen. Das ist eine normale Funktion der [Host-Erreichbarkeits](#)-Logik.

Sie werden auch feststellen, dass *Switch2* von *Monitor1* und *File1* in Bezug auf Benachrichtigungen oder Prüfausführung abhängt (welches davon ist unwichtig für dieses Beispiel). Wenn vorausschauende Host-Abhängigkeitsprüfungen aktiviert sind, wird Icinga parallele Prüfungen von *Monitor1* und *File1* sowie gleichzeitig für die direkten Eltern und Kinder von *Switch2* auslösen. Icinga tut dies, weil es weiß, dass es die Abhängigkeitslogik in der nahen Zukunft prüfen muss (z.B. für Zwecke der Benachrichtigung) und es will sicherstellen, dass es die aktuellsten Statusinformationen für die Hosts hat, die an der Abhängigkeit beteiligt sind.



So arbeiten vorausschauende Abhängigkeitsprüfungen. Einfach, oder?

👉 Anmerkung: Vorausschauende Service-Abhängigkeitsprüfungen arbeiten in einer ähnlichen Weise wie oben beschrieben. Außer natürlich, dass sie mit Services arbeiten statt mit Hosts.

Vorausschauende Prüfungen aktivieren

Vorausschauende Abhängigkeitsprüfungen verursachen ziemlich wenig Overhead, also würden wir empfehlen, dass Sie diese aktivieren. In den meisten Fällen werden die Vorteile, aktuelle Informationen für die Abhängigkeitslogik zu haben, den zusätzlichen Overhead durch diese Prüfungen mehr als ausgleichen.

Vorausschauende Abhängigkeitsprüfungen zu aktivieren ist einfach:

- Vorausschauende Host-Abhängigkeitsprüfungen werden durch die `enable_predictive_host_dependency_checks`-Option kontrolliert.
- Vorausschauende Service-Abhängigkeitsprüfungen werden durch die `enable_predictive_service_dependency_checks`-Option kontrolliert.

Cached Checks

Vorausschauende Abhängigkeitsprüfungen sind Prüfungen nach Bedarf und daher den Regeln von [cached checks](#) unterworfen. Cached checks können Ihnen Performance-Verbesserungen liefern, wenn Icinga darauf verzichtet, eine Host- oder Serviceprüfung durchzuführen, wenn es statt dessen ein relativ aktuelles Prüfungsergebnis nutzen kann. Mehr Informationen über cached checks finden Sie [hier](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Adaptive Überwachung](#)

[Zum Anfang](#)

[Zwischengespeicherte Prüfungen](#)



Zwischengespeicherte Prüfungen

[Zurück](#)

Kapitel 7. Fortgeschrittene Themen

[Weiter](#)

Zwischengespeicherte Prüfungen

Einführung



Die Leistung der Überwachungslogik von Icinga kann mit Hilfe von zwischengespeicherten Prüfungen (cached checks) nennenswert gesteigert werden. Zwischengespeicherte Prüfungen erlauben es Icinga, auf die Ausführung einer Host- oder Service-Prüfung zu verzichten, wenn es feststellt, dass ein recht aktuelles Prüfergebnis ausreicht.

Nur für Prüfungen nach Bedarf

Regelmäßig eingeplante Host- und Service-Prüfungen werden keine Leistungssteigerung durch zwischengespeicherte Prüfungen erfahren. Zwischengespeicherte Prüfungen sind nur sinnvoll zur Steigerung von Host- und Service-Prüfungen nach Bedarf. Geplante Prüfungen sorgen dafür, dass Host- und Service-Zustände regelmäßig aktualisiert werden, was in der Zukunft dazu führen kann, dass die Ergebnisse als zwischengespeicherte Prüfungen genutzt werden können.

Zur Erinnerung: Host-Prüfungen nach Bedarf treten auf...

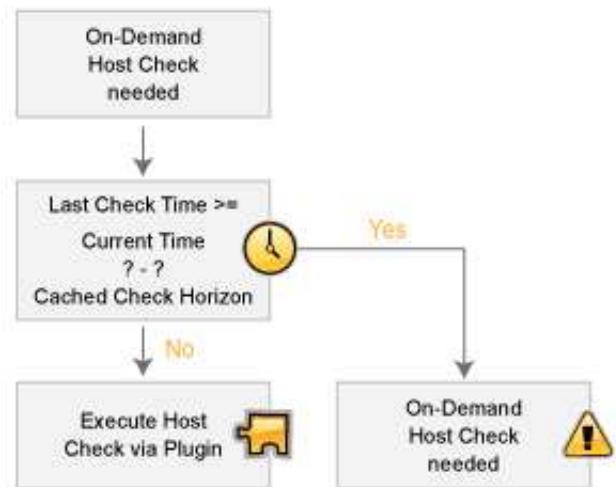
- wenn ein mit einem Host verbundener Service den Status wechselt
- wenn nötig als Teil der [Host-Erreichbarkeits-Logik](#)
- wenn nötig für [vorausschauende Host-Abhängigkeitsprüfungen](#)

und Service-Prüfungen nach Bedarf treten auf...

- wenn nötig für vorausschauende Service-Abhängigkeitsprüfungen

 Hinweis: Solange Sie keinen Gebrauch von Service-Abhängigkeiten machen, wird Icinga nicht in der Lage sein, zwischengespeicherte Prüfungen zur Leistungssteigerung von Service-Prüfungen zu nutzen. Keine Bange - das ist normal. Zwischengespeicherte Host-Prüfungen sorgen für große Leistungssteigerungen und jeder sollte dort einen Vorteil sehen.

Wie Zwischenspeicherung arbeitet



Wenn Icinga eine Host- oder Service-Prüfung nach Bedarf durchführen muss, wird es eine Festlegung treffen, ob es ein zwischengespeichertes Ergebnis benutzen kann oder ob es wirklich eine Prüfung durchführen muss. Es tut dies, indem es schaut, ob die letzte Prüfung für den Host oder Service innerhalb der letzten X Minuten erfolgte, wobei X der zwischengespeicherte Host- oder Service-Horizont ist.

Wenn die letzte Prüfung innerhalb des Zeitfensters erfolgte, das durch die cached-check-horizon-Variable angegeben ist, wird Icinga das Ergebnis der letzten Host- oder Service-Prüfung nutzen und *nicht* eine neue Prüfung ausführen. Wenn der Host oder Service noch nicht geprüft wurde oder die letzte Prüfung außerhalb des cached-check-horizon-Zeitfensters liegt, wird Icinga durch ein Plugin eine neue Host- oder Service-Prüfung durchführen.

Was dies wirklich bedeutet

Icinga führt Prüfungen nach Bedarf durch, weil es den aktuellen Status eines Hosts oder Service *in diesem Moment* wissen muss. Durch die Nutzung von zwischengespeicherten Prüfungen lassen Sie Icinga glauben, dass die kürzlichen Prüfungsergebnisse für die Ermittlung des aktuellen Zustands von Hosts "gut genug" sind und dass es nicht hergehen muss und erneut den Zustand für den Host oder Service prüfen muss.

Die cached-check-horizon-Variable teilt Icinga mit, wie aktuell Prüfergebnisse sein müssen, um zuverlässig den jetzigen Status eines Hosts oder Services darzustellen. Bei einem cached-check-horizon-Wert von 30 Sekunden sagen Sie Icinga, dass die Prüfung des Zustands eines Host innerhalb der letzten 30 Sekunden erfolgt sein muss, um noch als aktueller Zustand dieses Hosts angesehen zu werden.

Die Anzahl von zwischengespeicherten Prüfergebnissen, die Icinga nutzen kann, im Verhältnis zu der Anzahl von Prüfungen nach Bedarf, kann als die cached-check "Treffer"-Rate bezeichnet werden. Durch die Erhöhung des cached-check-horizon-Wertes bis zum regulären Prüfintervall des Hosts können Sie theoretisch eine Trefferrate von 100% erreichen. In diesem Fall würden alle Prüfungen nach Bedarf zwischengespeicherte Prüfergebnisse benutzen. Was für eine Leistungssteigerung! Aber ist es das wirklich? Wahrscheinlich nicht.

Die Zuverlässigkeit von zwischengespeicherten Prüfergebnissen nimmt mit der Zeit ab. Höhere Trefferraten erfordern, dass vorherige Prüfergebnisse für längere Zeit als "gültig" angesehen werden. Dinge können sich schnell in jedem Netzwerk-Szenario ändern, und es gibt keine Garantie dafür, dass es bei einem Server auf einmal brennt, der vor 30 Sekunden fehlerfrei funktionierte. Das ist der Kompromiss: Zuverlässigkeit gegen Geschwindigkeit. Wenn der cached-check-horizon-Wert groß ist, riskieren Sie, dass Sie unzuverlässige Prüfergebnisse in der Überwachungslogik haben.

Icinga wird letztendlich den korrekten Status aller Hosts und Services ermitteln, so dass es lediglich für eine kurze Zeit mit inkorrekt Informationen arbeitet, selbst wenn sich die zwischengespeicherten Prüfergebnisse als unzuverlässig herausstellen sollten. Selbst kurze Zeiten von unzuverlässigen Statusinformationen können sich für Admins als Ärgernis erweisen, wenn sie Benachrichtigungen über Probleme bekommen, die nicht länger existieren.

Es gibt keinen Standard-cached-check-horizon-Wert oder keine Trefferrate, die für jeden Icinga-Benutzer akzeptierbar wäre. Einige Leute möchten einen kleinen horizon-Zeitfenster und eine niedrige Trefferrate während andere ein größeres Zeitfenster und eine höhere Trefferrate bevorzugen (mit einer kleineren Zuverlässigkeitsrate). Einige Leute möchten vielleicht ganz auf zwischengespeicherte Prüfungen verzichten, um eine hundertprozentige Zuverlässigkeitsrate zu erhalten. Verschiedene horizon-Zeitfenster auszuprobieren und ihren Einfluss auf die Zuverlässigkeit von Statusinformationen zu sehen ist vielleicht das einzige Bedürfnis, das ein einzelner Benutzer hat, um den "richtigen" Wert für seine Situation zu finden.

Konfigurationsvariablen

Die folgenden Variablen legen die Zeitfenster fest, in denen ein vorangegangenes Prüfergebnis als ein zwischengespeichertes Prüfergebnis genutzt werden kann:

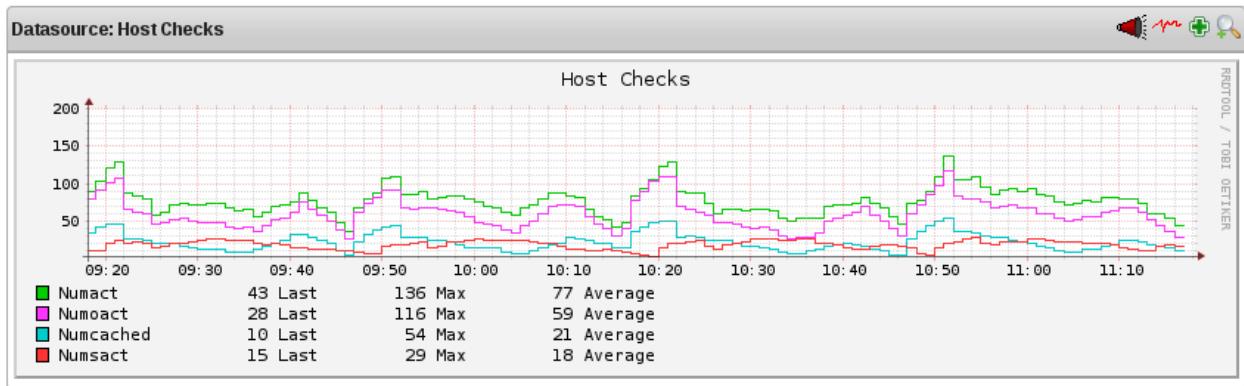
- Die `cached_host_check_horizon`-Variable kontrolliert zwischengespeicherte Host-Prüfungen.
- Die `cached_service_check_horizon`-Variable kontrolliert zwischengespeicherte Service-Prüfungen.

Zwischenspeichereffektivität optimieren

Um den größten Nutzen aus zwischengespeicherten Prüfungen zu ziehen, sollten Sie:

- regelmäßige Host-Prüfungen einplanen
- z.B. PNP4Nagios benutzen, um grafische Auswertungen von 1) Prüfungen nach Bedarf und 2) zwischengespeicherten Prüfungen zu erstellen
- die cached-check-horizon-Variable Ihren Anforderungen entsprechend anpassen

Sie können regelmäßige Prüfungen für Ihre Hosts durch einen größeren Wert als 0 in der `check_interval`-Option in Ihren [Host-Definitionen](#) einplanen. Wenn Sie das tun, sollten Sie die `max_check_attempts`-Option auf einen Wert größer als 1 setzen, oder es wird ein Performance-Problem geben. Das potentielle Performance-Problem ist [hier](#) genauer beschrieben.



Ein guter Weg, um den richtigen Wert für die cached-check-horizon-Optionen zu ermitteln, besteht im Vergleich der Anzahl von Prüfungen nach Bedarf gegen die Anzahl, in denen zwischengespeicherte Ergebnisse benutzt werden. Das [icingastats](#)-Dienstprogramm kann Informationen über zwischengespeicherte Prüfungen erzeugen, die dann mit [PNP4Nagios](#) dargestellt werden können. Ein Beispiel-Diagramm, das zwischengespeicherte Prüfungen gegen solche nach Bedarf darstellt, sehen Sie oben.

Bei der Testumgebung, aus der dieser Graph stammt, gab es...

- insgesamt 110 Hosts, die alle in regelmäßigen Abständen geprüft wurden
- ein durchschnittliches (regelmäßig geplantes) Host-Prüfintervall von 30 Minuten
- ein [cached_host_check_horizon](#) von 15 Sekunden

Das Diagramm zeigt, wie viele regelmäßig geplante Host-Prüfungen im Vergleich zu zwischengespeicherten Host-Prüfungen erfolgt sind. In diesem Beispiel wurden alle fünf Minuten ein Durchschnitt von 77 Host-Prüfungen durchgeführt. 59 von diesen (76%) sind Prüfungen nach Bedarf.

Es zeigt auch, wie viele zwischengespeicherte Host-Prüfungen während der Zeit aufgetreten sind. In diesem Beispiel waren es im Durchschnitt 21 Host-Prüfungen alle fünf Minuten.

Erinnern Sie sich, dass zwischengespeicherte Prüfungen nur für Prüfungen nach Bedarf verfügbar sind. Basierend auf den 5-Minuten- Durchschnitten der Graphen sehen wir, dass Icinga in 21 von 59 Fällen ein zwischengespeichertes Ergebnis benutzen kann, wenn Prüfungen nach Bedarf auszuführen sind. Das scheint nicht viel zu sein, aber diese Graphen stellen eine kleine Überwachungsumgebung dar. Bedenken Sie, dass 21 von 59 fast 36% sind und Sie können sich vorstellen, wie dies die Host-Prüf-Performance in großen Umgebungen steigern kann. Der Prozentsatz könnte größer sein, wenn der Wert der [cached_host_check_horizon](#)-Variablen erhöht wird, aber das würde die Zuverlässigkeit der zwischengespeicherten Host-Statusinformation verringern.

Sobald Sie ein paar Stunden oder Tage mit PNP4Nagios-Graphen haben, sollten Sie sehen, wie viele Host- und Service-Prüfungen mit Hilfe von Plugins ausgeführt werden gegen die, die zwischengespeicherte Prüfergebnisse benutzen. Nutzen Sie diese Informationen, um die cached-check-horizon-Variablen entsprechend für Ihre Situation anzupassen. Überwachen Sie weiterhin die PNP4Nagios-Graphen, um zu sehen, wie die Änderung der horizon-Variablen die zwischengespeicherten Prüf-Statistiken beeinflusst. Ändern und wiederholen Sie, falls erforderlich.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Vorausschauende
Abhängigkeitsprüfungen

[Zum Anfang](#)

Passive
Host-Zustandsübersetzung

© 2009-2011 Icinga Development Team, <http://www.icinga.org>

**Passive Host-Zustandsübersetzung**[Zurück](#)[Kapitel 7. Fortgeschrittene Themen](#)[Weiter](#)

Passive Host-Zustandsübersetzung

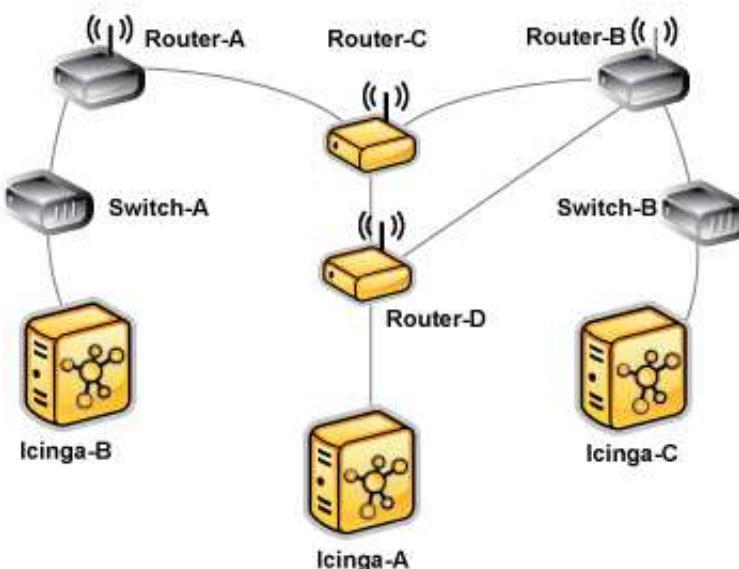
Einführung

Wenn Icinga passive Host-Prüfungen von entfernten Quellen erhält (d.h. andere Icinga-Instanzen in verteilten oder Failover-Umgebungen), gibt der von der entfernten Quelle gelieferte Host-Status ggf. nicht genau den aus Icinga-Sicht zutreffenden Zustand wieder. Weil verteilte und Failover-Überwachungs-Installationen ziemlich identisch sind, ist es wichtig einen Mechanismus anzubieten, um exakte Host-Zustände zwischen verschiedenen Icinga-Instanzen sicherzustellen.

Verschiedene Sichten

Das folgende Bild zeigt eine vereinfachte Sicht für ein Failover-Überwachungsaufbau.

- *Icinga-A* ist der primäre Überwachungsserver, der aktiv alle Switches und Router überwacht.
- *Icinga-B* und *Icinga-C* sind Backup-Überwachungsserver, die passive Prüfergebnisse von *Icinga-A* erhalten.
- Sowohl *Router-C* als auch *Router-D* sind fehlerhaft und daher offline.



In welchem Status sind *Router-C* und *Router-D* gerade? Die Antwort hängt davon ab, welche Icinga-Instanz Sie fragen.

- *Icinga-A* sieht *Router-D* als DOWN und *Router-C* als UNREACHABLE
- *Icinga-B* sollte *Router-C* als DOWN und *Router-D* als UNREACHABLE sehen
- *Icinga-C* sollte beide Router als DOWN sehen.

Jede Icinga-Instanz hat eine unterschiedliche Sicht des Netzwerks. Die Backup-Überwachungsserver sollten nicht blind passive Host-Zustände vom primären Überwachungsserver akzeptieren oder Sie werden inkorrekte Informationen über den aktuellen Zustand des Netzwerks haben.

Ohne die Übersetzung von passiven Host-Prüfergebnissen vom primären Überwachungsserver (*Icinga-A*) würde *Icinga-C* den *Router-D* als UNREACHABLE sehen, obwohl dieser vom eigenen Standpunkt aus eigentlich DOWN ist. Ähnliches gilt für die DOWN/UNREACHABLE-Zustände von *Router-C* und *Router-D* (vom Standpunkt von *Icinga-A* aus), die aus Sicht von *Icinga-B* umgedreht werden sollten.

 Anmerkung: Es kann einige Situationen geben, in denen Sie nicht möchten, dass Icinga die DOWN/UNREACHABLE-Zustände von entfernten Quellen in ihre "korrekten" Zustände vom Standpunkt der lokalen Icinga-Instanz aus umsetzt. Zum Beispiel möchten Sie vielleicht in verteilten Überwachungsumgebungen, dass die zentrale Icinga-Instanz weiß, wie verteilte Instanzen ihre jeweiligen Teile des Netzwerks sehen.

Status-Übersetzung aktivieren

Per Default wird Icinga *nicht* automatisch die DOWN/UNREACHABLE-Zustände von passiven Prüfergebnissen übersetzen. Sie müssen dieses Feature aktivieren, wenn Sie es benötigen und nutzen wollen.

Die automatische Übersetzung von passiven Host-Prüfzuständen wird durch die `translate_passive_host_checks`-Variable kontrolliert. Durch die Aktivierung wird Icinga automatisch DOWN- und UNREACHABLE-Zustände von entfernten Quellen in die korrekten Zustände für die lokale Instanz übersetzen.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Zwischengespeicherte Prüfungen](#)

[Zum Anfang](#)

Service- und
Host-Prüfungsplanung

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Service- und Host-Prüfungsplanung

[Zurück](#)[Kapitel 7. Fortgeschrittene Themen](#)[Weiter](#)

Service- und Host-Prüfungsplanung

Einführung

Es gab eine Menge Fragen dazu, wie Service-Prüfungen in bestimmten Situationen geplant werden, außerdem wie sich Planung und eigentliche Ausführung unterscheiden und wie die Ergebnisse verarbeitet werden. Wir werden versuchen, ein bisschen mehr ins Detail zu gehen, wie dies alles funktioniert...

Konfigurationsoptionen

Es gibt es verschiedene Konfigurationsoptionen, die beeinflussen, wie Service-Prüfungen geplant, ausgeführt und verarbeitet werden. Als Anfang enthält jede [Service-Definition](#) drei Optionen, die festlegen, wann und wie jede einzelne Service-Prüfung geplant und ausgeführt wird. Diese drei Optionen sind:

- *check_interval*
- *retry_interval*
- *check_period*

Es gibt außerdem vier Konfigurationsoptionen in der [Hauptkonfigurationsdatei](#), die Service-Prüfungen beeinflussen. Dies sind:

- *service_inter_check_delay_method*
- *service_interleave_factor*
- *max_concurrent_checks*
- *check_result_reaper_frequency*



Anmerkung

Die letzte Direktive betrifft auch Host-Prüfungen.

Wir werden nun mehr ins Detail gehen, wie all diese Optionen die Service-Prüfungsplaung beeinflussen. Lassen Sie uns zuerst betrachten, wie Services beim ersten (Neu-)Start von Icinga eingeplant werden...

Initiale Planung

Wenn Icinga (neu) startet, wird es versuchen, die initialen Prüfungen aller Services in einer Art und Weise so zu planen, dass die Load auf dem lokalen und den entfernten Hosts minimiert wird. Dies wird durch die Verteilung und das Verschachteln der Services erreicht. Die Verteilung von Service-Prüfungen (auch als inter-check-delay bekannt) wird benutzt, um die Last des lokalen Icinga-Servers zu minimieren/auszugleichen und die Verschachtelung wird benutzt, um die Last auf entfernten Hosts zu minimieren/auszugleichen. Sowohl inter-check-relay als auch Verschachtelungsfunktion werden nachfolgend erläutert.

Selbst wenn die Service-Prüfungen initial geplant werden, um die Last auf lokalen und entfernten Hosts auszubalancieren, werden die Dinge dem eintretenden Chaos nachgeben und ein wenig zufällig werden. Gründe dafür sind u.a., dass Services nicht alle mit dem gleichen Intervall geprüft werden, dass die Ausführung einiger Services länger dauert als andere, dass Host- und/oder Service-Probleme das Timing von ein oder mehreren Services verändern können, etc. Wenigstens versuchen wir, die Dinge gut zu beginnen. Hoffentlich hält die initiale Planung die Last auf dem lokalen und den entfernten Hosts im Laufe der Zeit relativ ausgeglichen...



Anmerkung

Wenn Sie die initiale Service-Prüfungs-Planungsinformationen ansehen möchten, dann starten Sie Icinga mit der `-s` Kommandozeilenoption. Dabei werden Ihnen grundlegende Planungsinformationen (inter-check-Verzögerung, Verschachtelungsfaktor, erste und letzte Service-Prüfzeit, etc., angezeigt) und es wird ein neues Status-Log angezeigt, das die genauen Zeiten darstellt, zu denen alle Services initial eingeplant werden. Weil diese Option das Status-Log überschreibt, sollte Sie sie nicht nutzen, solange eine weitere Icinga-Instanz läuft. Icinga wird *nicht* die Überwachung starten, wenn diese Option benutzt wird.

Inter-Check-Verzögerung (inter-check delay)

Wie bereits erwähnt, versucht Icinga die Last auf dem Rechner, auf dem Icinga läuft, auszugleichen, indem die Service-Prüfungen verteilt werden. Der Abstand zwischen aufeinander folgenden Service-Prüfungen wird "inter-check delay" genannt. Durch die Angabe eines Werts für die Variable `service_inter_check_delay_method` in der Hauptkonfigurationsdatei können Sie festlegen, wie diese Verzögerung berechnet wird. Wir werden erläutern, wie die "schlaue" Berechnung arbeitet, weil dies die Einstellung ist, die Sie für die normale Verarbeitung benutzen sollten.

Wenn Sie die Einstellung "smart" bei der Variable `service_inter_check_delay_method` angeben, wird Icinga den Wert für die inter-check-Verzögerung wie folgt berechnen:

$$\text{inter-check-Verzögerung} = (\text{durchschnittl. Check-Intervall für alle Services}) / (\text{Gesamtzahl der Services})$$

Nehmen wir ein Beispiel. Sagen wir, Sie haben 1.000 Services mit einem normalen Prüfintervall von fünf Minuten (natürlich werden einige Services mit anderen Intervallen geprüft, aber wir vereinfachen an dieser Stelle...). Die gesamte Check-Intervall-Zeit ist 5.000 ($1.000 * 5$). Das bedeutet, dass das durchschnittliche Check-Intervall für jeden Service fünf Minuten ist ($5.000 / 1.000$). Aufgrund dieser Information wissen wir, dass wir (im Durchschnitt) 1.000 Prüfungen pro fünf Minuten benötigen. Das heißt, dass wir eine inter-check-Verzögerung von 0,005 Minuten ($5 / 1000$, also etwa 0,3 Sekunden) benutzen sollten, wenn die Services das erste Mal verteilt werden. Durch die Verteilung alle 0,3 Sekunden können wir erreichen, dass Icinga jede Sekunde drei Service-Prüfungen einplant und/oder ausführt. Durch die gleichmäßige Aufteilung über die Zeit können wir hoffen, dass die Last auf dem lokalen Rechner, auf dem Icinga läuft, in etwa gleich bleibt.

Service-Verschachtelung (service interleaving)

Wie oben erläutert hilft die inter-check-Verzögerung dabei, die Last auf dem lokalen Host auszugleichen. Was ist aber mit entfernten Hosts? Ist es notwendig, die Last auf entfernten Hosts auszugleichen? Warum? Ja, es ist wichtig, und ja, Icinga kann dabei helfen. Wenn Sie eine große Zahl von Services auf einem entfernten Host überwachen und die Prüfungen nicht verteilt wären, dann könnte der entfernte Host denken, dass er das Opfer einer SYN-Attacke wurde, wenn es viele offene Verbindungen auf dem gleichen Port gibt. Außerdem ist es nett, wenn man versucht, die Last auf den Hosts auszugleichen/zu minimieren...

Durch die Angabe eines Werts für die Variable `service_interleave_factor` in der Hauptkonfigurationsdatei können beeinflussen, wie dieser Faktor berechnet wird. Wir werden erläutern, wie die "schlaue" Berechnung arbeitet, weil dies die Einstellung ist, die Sie für die normale Verarbeitung nutzen sollten. Sie können, natürlich, einen Wert vorgeben, anstatt ihn von Icinga berechnen zu lassen. Außerdem ist zu beachten, dass die Verschachtelung bei einem Wert von 1 praktisch deaktiviert ist.

Wenn Sie die Einstellung "smart" bei der Variable `service_interleave_factor` angeben, wird Icinga den Wert für den Verschachtlungsfaktor wie folgt berechnen:

$$\text{interleave factor} = \text{ceil} (\text{Gesamtzahl der Services} / \text{Gesamtzahl der Hosts})$$

Nehmen wir ein Beispiel. Sagen wir, Sie haben insgesamt 1.000 Services und 150 Hosts, die Sie überwachen. Icinga würde einen Verschachtlungsfaktor von 7 berechnen ($1000 / 150 = 6,6$; aufgerundet 7). Das bedeutet, dass Icinga bei der initialen Planung die erste Service-Prüfung einplant, die es findet, dann die nächsten sechs überspringt, den nächsten einplant, usw... Dieser Prozess wird wiederholt, bis alle Service-Prüfungen eingeplant sind. Weil die Services nach dem Namen des Hosts sortiert sind (und damit eingeplant werden), mit dem sie verbunden sind, wird dies helfen, die Last auf entfernten Hosts zu minimieren/auszugleichen.

Die folgenden Bilder zeigen, wie Service-Prüfungen eingeplant werden, wenn sie nicht verschachtelt werden (`service_interleave_factor=1`) und wenn sie mit einem Wert von 4 für `service_interleave_factor` verschachtelt werden.

nicht verschnebbare Prüfungen						
Rund	Service	Status	Last Check	Duration	Attempt	Technical Information
	Process	Process	Process	Process	Process	Process
100	Process1/Load	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Scsi_C_Files_Scan	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Read Memory_Use	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Memory_Monitor	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	FileIO	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
100	Process1/Load	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Physical_Memory_Use	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Read_Memory_Use	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Memory_Monitor	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Analysis_Process_U	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
100	Process1/Load	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Physical_Memory_Use	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Read_Memory_Use	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Scsi_C_Files_Scan	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Analysis_Process_U	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
100	Process1/Load	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Physical_Memory_Use	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Read_Memory_Use	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Scsi_C_Files_Scan	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001
	Analysis_Process_U	PENDING	N/A	04:05:15m	0/0	Searched check scheduled for Wk4 Aug 12/04/2001

consecutively, in the order they appear in the list. This is referred to as non-interleaved scheduling and is a result of setting the service interleaving factor to 1.

every 4th check in this list is
only scheduled. This is due to
the automatically calculated
leave factor in this example
was 4.

1

service checks are interleaved, based on remote servers by this given time are somewhat balanced.

Host	Condition	Status	Last Check	Duration	Average	Previous Information
OK	Amazon Processor Load	OK	06/01/2001 22:09:05	04:04:190	10%	Last OK: 5 min ago. 95% 5-95 avg:14.2%, 10 min avg:14.8%
OK	Amazon C/Fr Status	OK	06/01/2001 22:09:05	04:04:190	10%	Serial num: 737 MB Serial: 10000000000000000000000000000000
OK	Amazon Memory Usage	OK	06/01/2001 22:09:05	04:04:190	10%	Series check failed at Wed Aug 12 23:27:21 2001
Physical Mexico User	PING	N/A		04:04:190	10%	PING - failed. Last OK: Fri Jul 10 20:30:40
OK	PINGS	OK	06/01/2001 22:09:05	04:04:190	10%	PING - failed. Last OK: RTA < 0.20 ms
OK	Amazon Processor Use	OK	06/01/2001 22:12:12	04:04:190	10%	Partial memory: 743 MB Serial: 10000000000000000000000000000000
OK	Amazon Memory Usage	OK	06/01/2001 22:12:12	04:04:190	10%	Series check failed at Wed Aug 12 23:27:21 2001
OK	Amazon C/Fr Status	OK	06/01/2001 22:12:12	04:04:190	10%	Serial num: 2596 MB Serial: 10000000000000000000000000000000
OK	Amazon Processor Load	OK	06/01/2001 22:12:12	04:04:190	10%	Last OK: 5 min ago. 95% 5-95 avg:15.0%, 10 min avg:15.8%
OK	Physical Mexico User	OK	06/01/2001 22:12:12	04:04:190	10%	PING - failed. Last OK: GATE 5.10 - 14.00 ms
OK	Amazon Processor Use	OK	06/01/2001 22:09:05	04:04:190	10%	Series check failed at Wed Aug 12 23:27:21 2001
Physical Mexico User	PING	N/A		04:04:190	10%	Pinged metric: 9531 / 50 MB Serial: 10000000000000000000000000000000
OK	Amazon C/Fr Status	OK	06/01/2001 22:09:05	04:04:190	10%	Serial num: 762 MB Serial: 20000000000000000000000000000000
OK	Amazon Memory Usage	OK	06/01/2001 22:09:05	04:04:190	10%	Series check failed at Wed Aug 12 23:27:21 2001
OK	Physical Mexico User	OK	06/01/2001 22:09:05	04:04:190	10%	Pinged metric: 3057 / 50 MB Serial: 10000000000000000000000000000000

Now service checks have
ended after three "passes."

Maximale Zahl gleichzeitiger Service-Prüfungen

Um Icinga davon abzuhalten, all Ihre CPU-Ressourcen zu verbrauchen, können Sie die maximale Zahl von gleichzeitigen Service-Prüfungen beschränken, die zu einer beliebigen Zeit laufen können. Dies wird durch die Option `max_concurrent_checks` in der Hauptkonfigurationsdatei festgelegt.

Gut daran ist, dass Sie mit dieser Einstellung Icingas CPU-Nutzung beeinflussen können. Schlecht ist, dass Service-Prüfungen ins Hintertreffen geraten können, wenn dieser Wert zu niedrig eingestellt ist. Wenn es Zeit wird, eine Service-Prüfung auszuführen, wird Icinga sicherstellen, dass nicht mehr als x Service-Prüfungen ausgeführt werden bzw. darauf warten, dass die Prüfergebnisse verarbeitet werden (wobei x die Anzahl der Prüfungen ist, die Sie über die Option `max_concurrent_checks` angegeben haben). Falls diese Grenze erreicht ist, wird Icinga die Ausführung von anstehenden Prüfungen aufschieben, bis einige der vorherigen Prüfungen beendet sind. Also wie kann man einen geeigneten Wert für die Option `max_concurrent_checks` festlegen?

Zuerst müssen Sie einige Dinge wissen...

- die inter-check-Verzögerung, die Icinga benutzt, um die initialen Service-Prüfungen einzuplanen (nutzen Sie die Option `-s`, um den Wert zu kontrollieren)
 - die Häufigkeit (in Sekunden) von "reaper events", wie sie in der `check_result_reaper_frequency`-Variable in der Hauptkonfigurationsdatei angegeben ist
 - eine Vorstellung der durchschnittlichen Zeit, die Service-Prüfungen wirklich zur Ausführung benötigen (die meisten Plugins haben einen Timeout von 10 Sekunden, so dass der Durchschnitt wahrscheinlich niedriger liegt)

Dann benutzen Sie die folgende Berechnung, um einen geeigneten Wert für die maximale Zahl von gleichzeitig erlaubten Prüfungen zu errechnen...

max. Anzahl gleichzeitiger Prüfungen = ceil(max(check result reaper frequency , average check execution time) / inter-check delay)

Die errechnete Zahl sollte einen guten Ausgangspunkt für die *max_concurrent_checks*-Variable bieten. Es kann sein, dass Sie diesen Wert noch ein wenig erhöhen müssen, falls Service-Prüfungen nach wie vor nicht zur geplanten Zeit ausgeführt werden oder verringern Sie, falls Icinga zu viel CPU-Zeit beansprucht.

Nehmen wir an, dass Sie 875 Services, jeder mit einem durchschnittlichen Intervall von zwei Minuten. Das bedeutet, dass die inter-check-Verzögerung etwa 0,137 Sekunden ist. Wenn Sie die "check result reaper frequency" auf zehn Sekunden einstellen, können Sie einen ungefähren Wert für die maximale Zahl von gleichzeitigen Prüfungen wie folgt berechnen (wir nehmen an, dass die durchschnittliche Ausführungszeit für Service-Prüfungen kleiner als zehn Sekunden ist) ...

max. Zahl gleichzeitiger Prüfungen = ceil(10 / 0.137)

In diesem Fall ist der berechnete Wert 73. Das ergibt Sinn, denn Icinga wird etwas mehr als sieben neue Service-Prüfungen pro Sekunde ausführen und es wird Service-Prüfergebnisse nur alle zehn Sekunden verarbeiten. Das bedeutet, dass es zu einer beliebigen Zeit nur etwas mehr als 70 Service-Prüfungen gibt, die ausgeführt werden bzw. deren Ergebnisse verarbeitet werden. In diesem Fall würden wir wahrscheinlich empfehlen, den Wert für die Zahl der gleichzeitigen Prüfungen auf 80 zu erhöhen, weil es Verzögerungen gibt, wenn Icinga Service-Prüfergebnisse verarbeitet bzw. andere Dinge tut. Sie werden offensichtlich ein wenig testen und verändern müssen, damit alles reibungslos funktioniert, aber mit diesen Informationen sollten Sie ein paar generelle Richtlinien an der Hand haben...

Zeitbeschränkungen

Die Option *check_period* legt den **Zeitraum** fest, in dem Icinga Service-Prüfungen ausführen kann. Falls die Zeit, zu der eine Prüfung für einen bestimmten Service ausgeführt werden sollen, nicht innerhalb des angegebenen Zeitraum liegt, wird die Prüfung *nicht* ausgeführt, und zwar unabhängig vom Status des Service. Statt dessen wird Icinga die Service-Prüfung für die nächste gültige Zeit des Zeitraums einplanen. Wenn die Prüfung gestartet werden kann (d.h. die Zeit ist gültig innerhalb des Zeitraums), wird die Service-Prüfung ausgeführt.



Anmerkung

Auch wenn eine Service-Prüfung nicht zu einer bestimmten Zeit ausgeführt werden kann, könnte Icinga sie trotzdem *einplanen*. Das wird höchstwahrscheinlich während der initialen Planung von Services passieren, aber es kann auch in anderen Fällen passieren. Das bedeutet nicht, dass Icinga die Prüfung ausführen wird. Wenn es Zeit wird, die Prüfung tatsächlich *auszuführen*, wird Icinga kontrollieren, ob die Prüfung zur angegebenen Zeit gestartet werden kann. Falls nicht, wird Icinga die Service-Prüfung nicht ausführen, sondern sie zu einer späteren Zeit einplanen. Lassen Sie sich nicht verwirren! Die Planung und Ausführung von Service-Prüfungen sind zwei unterschiedliche (wenn auch, zusammenhängende) Dinge.

Normale Planung

In einer idealen Welt hätten Sie keine Netzwerkprobleme. Aber wenn das so wäre, dann hätten Sie auch kein Netzwerküberwachungsprogramm. Wie auch immer, wenn die Dinge reibungslos laufen und ein Service in einem OK-Zustand ist, nennen wir das "normal". Service-Prüfungen werden normalerweise in der Häufigkeit geplant, die in der Option *check_interval* angegeben ist. Das war's. Einfach, oder?

Planung bei Problemen

Also, was passiert, wenn es Probleme mit einem Service gibt? Nun, eins der Dinge ist, dass sich die Service-Prüfungsplanung ändert. Wenn Sie die Option `max_attempts` auf einen Wert größer als eins gesetzt haben, wird Icinga die Prüfung erneut einplanen, bevor es entscheidet, dass ein wirkliches Problem existiert. Während der Service erneut geprüft wird (bis zu `max_attempts` Mal), wird er als in einem "soft"-Status befindlich angesehen (wie [hier](#) beschrieben) und die Service-Prüfungen werden mit einer Häufigkeit eingeplant, die in der Option `retry_interval` angegeben ist.

Wenn Icinga den Service `max_attempts` Mal erneut eingeplant hat und er immer noch in einem nicht-OK Status ist, wird Icinga den Service in einen "Hard"-Status versetzen, Benachrichtigungen an Kontakte versenden (falls zutreffend) und weitere Prüfungen des Service wieder mit der Häufigkeit planen, die in der Option `check_interval` festgelegt ist.

Wie immer gibt es Ausnahmen der Regel. Wenn eine Service-Prüfung zu einem nicht-OK-Status führt, wird Icinga den mit diesem Service verbundenen Host prüfen, um festzustellen, ob er "up" ist oder nicht (siehe die Anmerkung [unten](#) zu Informationen, wie dies passiert). Wenn der Host nicht "up" ist (also "down" oder "unreachable"), wird Icinga den Service sofort in einen harten nicht-OK-Status versetzen und die Zahl der aktuellen Versuche auf 1 zurücksetzen. Da der Service in einem harten nicht-OK-Status ist, wird die Service-Prüfung mit der normalen Häufigkeit geplant, die in der Option `check_interval` angegeben ist, statt des Wertes aus der Option `retry_interval`.

Host-Prüfungen

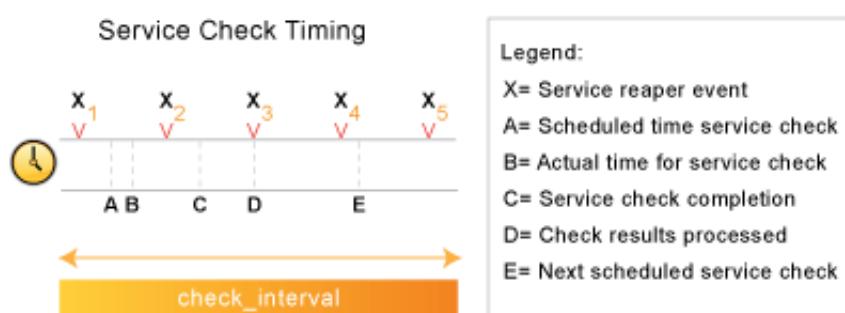
Ein Fall, wo Icinga den Status eines Hosts prüft, ist, wenn ein Service-Prüfung einen nicht-OK-Zustand ergibt. Icinga prüft den Host, um zu entscheiden, ob der Host "up" ist oder nicht, bzw. ob der Host "up", "down" oder "unreachable" ist. Wenn die erste Host-Prüfung einen nicht-OK-Zustand ergibt, wird Icinga Host-Prüfungen wie bei den Services durchführen.

Planungsverzögerungen

Es sollte erwähnt werden, dass Service-Prüfungsplanung und -ausführung geschieht, so gut es geht. Individuelle Service-Prüfungen werden in Icinga als Ereignisse mit niedriger Priorität angesehen, so dass sie verzögert werden können, wenn Ereignisse mit höherer Priorität ausgeführt werden müssen. Beispiel von Ereignissen mit hoher Priorität umfassen Log-Datei-Rotationen, externe Befehlsprüfungen und Prüfergebnis-Ernteereignisse.

Planungsbeispiel

Die Planung von Service-Prüfungen, ihre Ausführung und die Verarbeitung ihrer Ergebnisse können ein bisschen schwierig zu verstehen sein, deshalb schauen wir uns ein einfaches Beispiel an. Betrachten wir das folgende Diagramm - wir werden uns darauf beziehen, während wir die Dinge erklären.



Zuallererst sind X_n Prüfergebnis-Ernteereignisse, die in der Häufigkeit geplant werden, die durch die Option `check_result_reaper_frequency` in der Hauptkonfigurationsdatei angegeben ist. Prüfergebnis-Ernteereignisse übernehmen die Arbeit, Service-Prüfergebnisse zu sammeln und zu verarbeiten. Sie dienen als die Kernlogik für Icinga, starten Host-Prüfungen, Ereignisbehandlung-Routinen und Benachrichtigungen, wenn das notwendig ist.

Für das Beispiel hier ist die Ausführung eines Service für den Zeitpunkt **A** geplant. Allerdings kam Icinga in der Ereigniswarteschlange ins Hintertreffen, so dass die Prüfung erst zum Zeitpunkt **B** ausgeführt wird. Die Service-Prüfung endet zum Zeitpunkt **C**, so dass die Differenz zwischen den Punkten **C** and **B** die Laufzeit der Prüfung ist.

Die Ergebnisse der Service-Prüfungen werden nicht sofort nach der Prüfung verarbeitet. Statt dessen werden die Ergebnisse für eine spätere Verarbeitung durch einen Prüfergebnis-Ernteereignis gespeichert. Das nächste Prüfergebnis-Ernteereignis findet zum Zeitpunkt **D** statt, so dass dies ungefähr die Zeit ist, zu der die Ergebnisse verarbeitet werden (die tatsächliche Zeit kann später als **D** sein, weil ggf. andere Service-Prüfergebnisse vor diesem Service verarbeitet werden).

Zu der Zeit, zu der das Prüfergebnis-Ernteereignis die Service-Prüfergebnisse verarbeitet, wird es die nächste Service-Prüfung einplanen und in Icingsas Ereigniswarteschlange stellen. Wir nehmen an, dass die Service-Prüfung einen OK-Zustand ergibt, so dass die nächste Prüfung zum Zeitpunkt **E** nach der ursprünglich geplanten Prüfzeit geplant wird, mit einem zeitlichen Abstand, der in der `check_interval`-Option angegeben ist. Beachten Sie, dass der Service *nicht* erneut eingeplant wird basierend auf der Zeit, zu der er tatsächlich ausgeführt wird! Es gibt eine Ausnahme (es gibt immer eine, oder?) - falls die Zeit, zu der die Service-Prüfung tatsächlich ausgeführt wird (Punkt **B**) nach der nächsten Service-Prüfzeit liegt (Punkt **E**), wird Icinga das durch das Anpassen der nächsten Prüfzeit ausgleichen. Das wird gemacht, damit Icinga nicht verrückt wird beim Versuch, mit den Service-Prüfungen Schritt zu halten, wenn eine hohe Last auftritt. Außerdem, wie sinnvoll ist es, etwas in der Vergangenheit zu planen...?

Service-Definitionsoptionen, die die Planung beeinflussen

Jede Service-Definition enthält eine `check_interval`- und eine `retry_interval`-Option. Hoffentlich klärt das Folgende, was diese zwei Optionen tun, wie sie mit der `max_check_attempts`-Option in der Service-Definition zusammenwirken, und wie sie die Planung des Service beeinflussen.

Zuallererst gibt die `check_interval`-Option das Intervall an, in dem der Service unter "normalen" Umständen geprüft wird. "Normale" Umstände bedeutet, wenn sich der Service in einem "OK"- oder einem **harten** nicht-OK-Zustand befindet.

Wenn ein Service das erste Mal von einem OK- in einen nicht-OK-Zustand wechselt, gibt Ihnen Icinga die Möglichkeit, das Intervall temporär zu verkleinern oder zu vergrößern, in dem nachfolgende Prüfungen für diesen Service ausgeführt werden. Wenn der Service-Zustand das erste Mal wechselt, wird Icinga bis zu `max_check_attempts`-1 Versuche durchführen, bevor es entscheidet, dass es sich um ein richtiges Problem handelt. Während die Prüfungen wiederholt werden, werden sie gemäß der `retry_interval`-Option neu eingeplant, was schneller oder langsam als die `check_interval`-Option ist. Während der Service erneut geprüft wird (bis zu `max_check_attempts`-1 mal), ist der Service in einem **soft-Zustand**. Wenn der Service `max_check_attempts`-1 mal geprüft wurde und sich immer noch in einem nicht-OK-Zustand befindet, wird der Service in einen **hard-Zustand** wechseln und wird nachfolgend wieder mit der normalen Rate eingeplant, die in der `check_interval`-Option festgelegt ist.

Als Randbemerkung, wenn Sie einen Wert von 1 für die `max_check_attempts`-Option definieren, wird der Service niemals mit dem Intervall geprüft, das in der `retry_interval`-Option angegeben ist. Statt dessen wird er sofort in einen **hard-Zustand** wechseln und anschließend mit dem in der Option `check_interval` festgelegten Intervall geprüft.

TODO

Host-Prüfungs-Direktiven

Die meisten der o.g. Informationen treffen auch auf Host-Prüfungen zu.

Dieser Abschnitt wird aktualisiert. Voraussichtlich gibt es mehr Informationen in einer der nächsten Ausgaben...

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Passive
Host-Zustandsübersetzung

[Zum Anfang](#)

Angepasste CGI-Kopf- und
Fußzeilen

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Anangepasste CGI-Kopf- und Fußzeilen

[Zurück](#)

[Kapitel 7. Fortgeschrittene Themen](#)

[Weiter](#)

Anangepasste CGI-Kopf- und Fußzeilen

Einführung

Wenn Sie Icinga-Installationen für Kunden machen, dann möchten Sie vielleicht, dass in den [CGIs](#) angepasste Kopf- und Fußzeilen angezeigt werden. Dies ist besonders dann nützlich, wenn Sie Support-Kontaktinformationen u.ä. für den Endbenutzer anzeigen möchten.

Es ist wichtig anzumerken, dass angepasste Dateien mit Kopf- und Fußzeilen nicht in irgendeiner Form vorverarbeitet werden (solange es sich nicht um ausführbare Dateien handelt), bevor sie angezeigt werden. Der Inhalt der Kopf- und Fußzeilen wird ganz einfach gelesen und in der CGI-Ausgabe angezeigt. Das bedeutet, dass diese Dateien lediglich Informationen enthalten können, die ein Web-Browser versteht (HTML, JavaScript, usw.).

Wenn die angepassten Kopf- und Fußzeilendateien ausführbar sind, dann werden sie ausgeführt und die Ausgaben an den Benutzer zurückgeliefert, so dass die Dateien gültigen HTML-Code enthalten sollten. Auf diese Weise können Sie Ihre eigenen CGIs nutzen, um Daten in der Icinga-Anzeige auszugeben. Dies kann genutzt werden, um mit ddraw Grafiken aus rrdtool einzufügen und Befehlsmenüs im Icinga-Fenster anzuzeigen. Die ausführbaren angepassten Kopf- und Fußzeilendateien werden mit der gleichen CGI-Umgebung ausgeführt wie das Icinga-Haupt-CGI, so dass Ihre Dateien die Abfrageinformationen, Benutzerauthentifizierungsinformationen usw. analysieren können, um entsprechende Ausgaben zu erzeugen.

Wie funktioniert es?

Sie können angepasste Kopf- und Fußzeilen in die Ausgaben der CGIs einschließen, indem Sie entsprechend benannte HTML-Dateien im `ssi`-Unterverzeichnis des Icinga-HTML-Verzeichnisses (z.B. `/usr/local/icinga/share/ssi`) ablegen.

Anangepasste Kopfzeilen werden direkt hinter dem `<BODY>`-Tag in der CGI-Ausgabe eingefügt, während angepasste Fußzeilen direkt vor dem schließenden `</BODY>`-Tag eingefügt werden.

Es gibt zwei Arten von angepassten Kopf- und Fußzeilen:

- Globale Kopf-/Fußzeilen: diese Dateien sollten `common-header.ssi` und `common-footer.ssi` benannt werden. Wenn diese Dateien existieren, werden sie in die Ausgaben aller CGIs eingefügt.
- CGI-spezifische Kopf-/Fußzeilen: diese Dateinamen sollten im Format `CGINAME-header.ssi` und `CGINAME-footer.ssi` benannt werden, wobei `CGINAME` der (Datei-) Name des CGIs ohne die `.cgi`-Erweiterung ist. Die Kopf- und Fußzeilendateien des [alert summary CGI](#)

(summary.cgi) würden beispielsweise *summary-header.ssi* und *summary-footer.ssi* heißen.

Sie sind nicht gezwungen, irgendwelche angepassten Kopf- und Fußzeilen zu benutzen. Sie können nur eine globale Kopfzeile benutzen, wenn Sie möchten. Sie können nur CGI-spezifische Kopfzeilen und eine globale Fußzeile benutzen, wenn Sie möchten. Ganz wie Sie wollen. Wirklich.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Service- und
Host-Prüfungsplanung

[Zum Anfang](#)

Objektvererbung

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Objektvererbung

[Zurück](#)

Kapitel 7. Fortgeschrittene Themen

[Weiter](#)

Objektvererbung

Einführung

Dieses Dokument versucht Objektvererbung zu erklären und wie sie in Ihren [Objektdefinitionen](#) genutzt werden kann.

Wenn Sie nach dem Lesen verwirrt sind, wie Rekursion und Vererbung arbeiten, sollten Sie einen Blick in die Beispielobjektkonfigurationsdateien in der Icinga-Distribution werfen. Wenn das immer noch nicht hilft, dann senden Sie eine (englischsprachige) e-Mail mit einer *detaillierten* Beschreibung Ihres Problems an die *icinga-users*-Mailing-List.

Grundlagen

Es gibt drei Variablen in allen Objektdefinitionen, die Rekursion und Vererbung beeinflussen. Sie sind wie folgt "*dargestellt*":

```
define someobjecttype{
    object-specific variables ...
    name          template_name
    use           name_of_template_to_use
    register      [ 0/1 ]
}
```

Die erste Variable heißt *name*. Das ist lediglich ein "Vorlagen"-Name (template name), auf den in anderen Objektdefinitionen verwiesen wird, so dass diese die Objekteigenschaften/Variablen erben. Vorlagennamen müssen innerhalb der Objekte des gleichen Typs eindeutig sein, so dass Sie nicht zwei oder mehr Host-Definitionen mit "hosttemplate" als Namen haben können.

Die zweite Variable heißt *use*. Hier geben Sie den Namen der Vorlage an, deren Eigenschaften/Variablen Sie erben möchten. Der Name, den Sie für diese Variable angeben, muss als Vorlage definiert sein (mit Hilfe der *name*-Variable).

Die dritte Variable heißt *register*. Diese Variable wird benutzt, um anzuzeigen, ob die Objektdefinition "registriert" werden soll. Per Default werden alle Objektdefinitionen registriert. Wenn Sie eine partielle Objektdefinition als Vorlage nutzen, möchten Sie verhindern, dass sie registriert wird (ein Beispiel dazu folgt). Die Werte sind wie folgt: 0 = die Objektdefinition NICHT registrieren, 1 = die Objektdefinition registrieren (das ist der Default). Diese Variable wird NICHT vererbt, bei jeder als Vorlage genutzten (Teil-) Objektdefinition muss explizit die *register*-Direktive auf 0 gesetzt werden. Dies verhindert die Notwendigkeit, eine vererbte *register*-Direktive für jedes zu registrierende Objekt mit einem Wert von 1 zu übersteuern.

Lokale Variablen gegenüber vererbten Variablen

Bei der Vererbung ist es wichtig zu wissen, dass "lokale" Objektvariablen immer Vorrang vor Variablen aus der Vorlage haben. Werfen Sie einen Blick auf das folgende Beispiel mit zwei Host-Definitionen (nicht alle notwendigen Variablen sind dargestellt):

```
define host{
    host_name          bighost1
    check_command      check-host-alive
    notification_options d,u,r
    max_check_attempts 5
    name               hosttemplate1
}
define host{
    host_name          bighost2
    max_check_attempts 3
    use                hosttemplate1
}
```

Sie werden bemerken, dass die Definition für den Host *bighost1* mit Hilfe der Vorlage *hosttemplate1* definiert wurde. Die Definition für Host *bighost2* nutzt die Definition von *bighost1* als Vorlagenobjekt. Sobald Icinga diese Daten verarbeitet hat, wäre die resultierende Definition von *bighost2* äquivalent zu dieser Definition:

```
define host{
    host_name          bighost2
    check_command      check-host-alive
    notification_options d,u,r
    max_check_attempts 3
}
```

Sie sehen, dass die *check_command*- und *notification_options*-Variablen vom Vorlagenobjekt geerbt wurden (wo Host *bighost1* definiert wird). Trotzdem wurden die *host_name*- und *max_check_attempts*-Variablen nicht vom Vorlagenobjekt geerbt, weil sie lokal definiert wurden. Erinnern Sie sich, dass von einem Vorlagenobjekt geerbte Variablen von lokal definierten Variablen überschrieben werden. Das sollte ein ziemlich einfach zu verstehendes Konzept sein.



Hinweis: wenn Sie möchten, dass lokale Zeichenketten-Variablen an geerbte Zeichenkettenwerte angehängt werden, können Sie das tun. Lesen Sie [weiter unten](#) mehr darüber, wie das erreicht werden kann.

Vererbungsverkettung

Objekte können Eigenschaften/Variablen aus mehreren Ebenen von Vorlagenobjekten erben. Nehmen Sie das folgende Beispiel:

```
define host{
    host_name          bighost1
    check_command      check-host-alive
    notification_options d,u,r
    max_check_attempts 5
    name               hosttemplate1
}
define host{
    host_name          bighost2
    max_check_attempts 3
    use                hosttemplate1
    name               hosttemplate2
}
```

```
define host{
    host_name          bighost3
    use                hosttemplate2
}
```

Sie werden bemerken, dass die Definition von Host *bighost3* Variablen von der Definition von *bighost2* erbt, die wiederum Variablen von der Definition von Host *bighost1* erbt. Sobald Icinga diese Konfigurationsdaten verarbeitet, sind die resultierenden Host-Definition äquivalent zu den folgenden:

```
define host{
    host_name          bighost1
    check_command      check-host-alive
    notification_options d,u,r
    max_check_attempts 5
}
define host{
    host_name          bighost2
    check_command      check-host-alive
    notification_options d,u,r
    max_check_attempts 3
}
define host{
    host_name          bighost3
    check_command      check-host-alive
    notification_options d,u,r
    max_check_attempts 3
}
```

Es gibt keine eingebaute Beschränkung, wie "tief" Vererbung gehen kann, aber Sie sollten sich vielleicht selbst auf ein paar Ebenen beschränken, um die Übersicht zu behalten.

Unvollständige Objektdefinitionen als Vorlagen nutzen

Es ist möglich, unvollständige Objektdefinitionen als Vorlage für andere Objektdefinitionen zu nutzen. Mit "unvollständiger" Definition meinen wir, dass nicht alle benötigten Variablen in der Objektdefinition angegeben wurden. Es mag komisch klingen, unvollständige Definitionen als Vorlagen zu nutzen, aber es ist tatsächlich empfohlen, dies zu tun. Warum? Nun, sie können als ein Satz von Defaults für alle anderen Objektdefinitionen dienen. Nehmen Sie das folgende Beispiel:

```
define host{
    check_command      check-host-alive
    notification_options d,u,r
    max_check_attempts 5
    name               generichosttemplate
    register           0
}
define host{
    host_name          bighost1
    address            192.168.1.3
    use                generichosttemplate
}
define host{
    host_name          bighost2
    address            192.168.1.4
    use                generichosttemplate
}
```

Beachten Sie, dass die erste Host-Definition unvollständig ist, weil die erforderliche *host_name*-Variable fehlt. Wir müssen keinen Host-Namen angeben, weil wir diese Definition als Vorlage nutzen wollen. Um Icinga daran zu hindern, diese Definition als einen normalen Host anzusehen, setzen wir die *register*-Variable auf 0.

Die Definitionen von *bighost1* und *bighost2* erben ihre Werte von der generischen Host-Definition. Die einzige Variable, die überschrieben wird, ist die *address*-Variable. Das bedeutet, dass beide Hosts exakt die gleichen Eigenschaften haben, bis auf die *host_name*- und *address*-Variablen. Sobald Icinga die Konfigurationsdaten im Beispiel verarbeitet, wären die resultierenden Host-Definitionen äquivalent zu folgenden:

```
define host{
    host_name          bighost1
    address            192.168.1.3
    check_command      check-host-alive
    notification_options d,u,r
    max_check_attempts 5
}
define host{
    host_name          bighost2
    address            192.168.1.4
    check_command      check-host-alive
    notification_options d,u,r
    max_check_attempts 5
}
```

Die Nutzung einer Vorlagendefinition für Default-Werte erspart Ihnen mindestens eine Menge Tipparbeit. Es spart Ihnen auch eine Menge Kopfschmerzen, wenn Sie später die Default-Werte von Variablen für eine große Zahl von Hosts wollen.

eigene Objektvariablen (custom object variables)

Jede [eigene Objektvariable](#), die Sie in Ihren Host-, Service- oder Kontaktdefinitionen definieren, wird wie jede andere Standardvariable vererbt. Nehmen Sie das folgende Beispiel:

```
define host{
    _customvar1           somevalue ; <-- Custom host variable
    _snmp_community      public   ; <-- Custom host variable
    name                 generichosttemplate
    register             0
}
define host{
    host_name            bighost1
    address              192.168.1.3
    use                  generichosttemplate
}
```

Der Host *bighost1* wird die eigenen Host-Variablen *_customvar1* und *_snmp_community* von der *generichosttemplate*-Definition erben, zusammen mit den entsprechenden Werten. Die daraus resultierende Definition für *bighost1* sieht wie folgt aus:

```
define host{
    host_name            bighost1
    address              192.168.1.3
    _customvar1           somevalue
    _snmp_community      public
}
```

Vererbung für Zeichenketten-Werte aufheben

In einigen Fällen möchten Sie vielleicht nicht, dass Ihre Host-, Service- oder Kontakt-Definitionen Werte von Zeichenketten-Variablen aus Vorlagen erben. Wenn das der Fall ist, können Sie "null" (ohne Anführungszeichen) als den Wert der Variable, die Sie nicht erben möchten. Nehmen Sie das folgende Beispiel:

```

define host{
    event_handler      my-event-handler-command
    name               generichosttemplate
    register          0
}
define host{
    host_name         bighost1
    address           192.168.1.3
    event_handler     null
    use               generichosttemplate
}

```

In diesem Fall wird der Host *bighost1* nicht den Wert der *event_handler*-Variable erben, die in der *generichosttemplate*-Vorlage definiert ist. Die resultierende Definition von *bighost1* sieht wie folgt aus:

```

define host{
    host_name         bighost1
    address           192.168.1.3
}

```

additive Vererbung von Zeichenketten-Werten

Icinga gibt lokalen Variablen Vorrang vor Werten, die von Vorlagen vererbt werden. In den meisten Fällen überschreiben lokale Variablenwerte jene, die in Vorlagen definiert sind. In einigen Fällen ist es sinnvoll, dass Icinga die Werte von geerbten *und* lokalen Variablen gemeinsam nutzt.

Diese "additive Vererbung" kann durch Voranstellen eines Pluszeichens (+) vor den lokalen Variablenwert erreicht werden. Dieses Feature ist nur für Standard-Variablen verfügbar, die Zeichenketten-Werte enthalten. Nehmen Sie das folgende Beispiel:

```

define host{
    hostgroups        all-servers
    name              generichosttemplate
    register          0
}
define host{
    host_name         linuxserver1
    hostgroups        +linux-servers,web-servers
    use               generichosttemplate
}

```

In diesem Fall wird der *linuxserver1* den Wert der lokalen *hostgroups*-Variablen dem der *generichosttemplate*-Vorlage hinzufügen. Die resultierende Definition von *linuxserver1* sieht wie folgt aus:

```

define host{
    host_name         linuxserver1
    hostgroups        all-servers,linux-servers,web-servers
}

```

Implizite Vererbung

Normalerweise müssen Sie entweder explizit den Wert einer erforderlichen Variable in einer Objektdefinition angeben oder sie von einer Vorlage erben. Es gibt ein paar Ausnahmen zu dieser Regel, in denen Icinga annimmt, dass Sie einen Wert benutzen wollen, der statt dessen von einem verbundenen Objekt kommt. Die Werte einiger Service-Variablen werden zum Beispiel vom Host kopiert, mit dem der Service verbunden ist, wenn Sie diese nicht anderweitig angeben.

Die folgende Tabelle führt die Objektvariablen auf, die implizit von verbundenen Objekten vererbt werden, wenn Sie deren Werte nicht explizit angeben oder sie von einer Vorlage erben.

Objekttyp	Objektvariable	implizite Quelle
Services	<i>contact_groups</i>	<i>contact_groups</i> in der verbundenen Host-Definition
	<i>notification_interval</i>	<i>notification_interval</i> in der verbundenen Host-Definition
	<i>notification_period</i>	<i>notification_period</i> in der verbundenen Host-Definition
Host Escalations	<i>contact_groups</i>	<i>contact_groups</i> in der verbundenen Host-Definition
	<i>notification_interval</i>	<i>notification_interval</i> in der verbundenen Host-Definition
	<i>escalation_period</i>	<i>notification_period</i> in der verbundenen Host-Definition
Service Escalations	<i>contact_groups</i>	<i>contact_groups</i> in der verbundenen Service-Definition
	<i>notification_interval</i>	<i>notification_interval</i> in der verbundenen Service-Definition
	<i>escalation_period</i>	<i>notification_period</i> in der verbundenen Service-Definition

implizite/additive Vererbung bei Eskalationen

Service- und Host-Eskalationsdefinitionen können eine spezielle Regel benutzen, die die Möglichkeiten von impliziter und additiver Vererbung kombiniert. Wenn Eskalationen 1) nicht die Werte ihrer *contact_groups*- oder *contacts*-Direktiven von anderen Eskalationsvorlagen erben und 2) ihre *contact_groups*- oder *contacts*-Direktiven mit einem Plus-Zeichen (+) beginnen, dann werden die Werte der *contact_groups* oder *contacts*-Direktiven der entsprechenden Host- oder Service-Definitionen in der additiven Vererbungslogik benutzt.

Verwirrt? Hier ein Beispiel:

```
define host{
    name          linux-server
    contact_groups linux-admins
    ...
}
define hostescalation{
    host_name      linux-server
    contact_groups +management
    ...
}
```

Das ist ein viel einfacheres Äquivalent zu:

```
define hostescalation{
    host_name      linux-server
    contact_groups linux-admins,management
    ...
}
```

Wichtige Werte (important values)

Service-Vorlagen können eine spezielle Regel benutzen, die ihrem check_command-Wert Vorrang gibt. Wenn das check_command mit einem Ausrufungszeichen (!) beginnt, dann wird das check_command der Vorlage als wichtig markiert und wird statt des im Service definierten check_command (dies ist der CSS-Syntax nachempfunden, die ! als wichtiges Attribut benutzt).

Warum ist das nützlich? Es ist hauptsächlich dann sinnvoll, wenn ein unterschiedliches check_command für verteilte Systeme gesetzt wird. Sie wollen vielleicht einen Frische-Schwellwert und ein check_command setzen, der den Service in einen fehlerhaften Status versetzt, aber das funktioniert nicht mit dem normalen Vorlagensystem. Dieses "wichtig"-Kennzeichen erlaubt es, das angepasste check_command zu schreiben, aber eine allgemeine verteilte Vorlage zu benutzen, die das check_command überlagert, wenn es auf dem zentralen Icinga-Server eingesetzt wird.

Zum Beispiel:

```
# On master
define service {
    name                  service-distributed
    register              0
    active_checks_enabled 0
    check_freshness       1
    check_command         !set_to_stale
}
# On slave
define service {
    name                  service-distributed
    register              0
    active_checks_enabled 1
}
# Service definition, used by master and slave
define service {
    host_name            host1
    service_description   serviceA
    check_command         check_http...
    use                  service-distributed
    ...
}
```



Anmerkung

Bitte beachten Sie, dass nur eine Vererbungsebene bei diesen wichtigen Werten möglich ist. Das bedeutet, dass Sie nicht das check_command von einer Vorlage zu einer weiteren und von dort zum Service vererben können.

```
Template1 => Service1           <== funktioniert
Template1 => Template2 => Service1 <== funktioniert NICHT
```

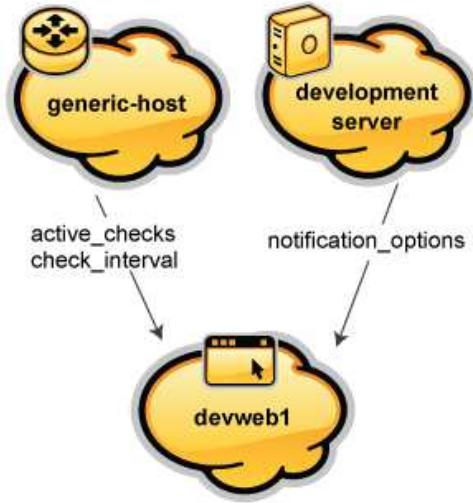
Mehrere Vererbungsquellen

Bisher haben alle Beispiele Objektdefinitionen gezeigt, die Variablen/Werte von einer einzelnen Quelle erben. Sie können für komplexere Konfigurationen auch Variablen/Werte von mehreren Quellen erben, wie unten gezeigt.

```

# Generic host template
define host{
    name          generic-host
    active_checks_enabled 1
    check_interval 10
    ...
    register      0
}
# Development web server template
define host{
    name          development-server
    check_interval 15
    notification_options d,u,r
    ...
    register      0
}
# Development web server
define host{
    use           generic-host,development-server
    host_name     devweb1
    ...
}

```



Im obigen Beispiel erbt *devweb1* Variablen/Werte von zwei Quellen: *generic-host* und *development-server*. Sie werden bemerken, dass in beiden Quellen eine *check_interval*-Variable definiert ist. Weil *generic-host* die erste in *devweb1* durch die *use*-Direktive angegebene Vorlage ist, wird der Wert für die *check_interval*-Variable durch den *devweb1*-Host vererbt. Nach der Vererbung sieht die Definition von *devweb1* wie folgt aus:

```

# Development web server
define host{
    host_name     devweb1
    active_checks_enabled 1
    check_interval 10
    notification_options d,u,r
    ...
}

```

Vorrang bei mehreren Vererbungsquellen

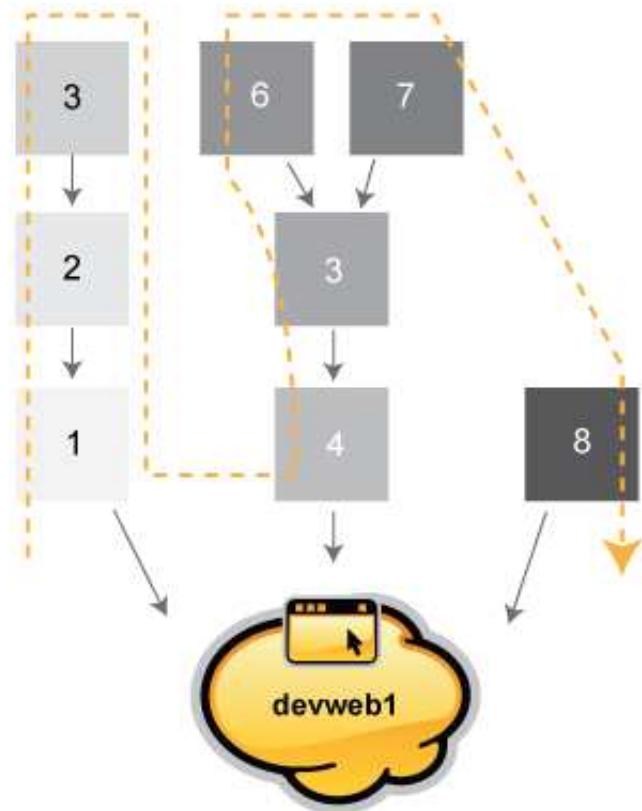
Wenn Sie mehrere Vererbungsquellen nutzen, ist es wichtig zu wissen, wie Icinga Variablen behandelt, die in mehreren Quellen definiert sind. In diesen Fällen wird Icinga die Variable/den Wert aus der ersten Quelle benutzen, die in der *use*-Direktive angegeben ist. Weil Vererbungsquellen ebenfalls Variablen/Werte aus ein oder mehreren Quellen erben können, kann es kompliziert werden herauszufinden, welche Variablen/Werte-Paare Vorrang haben.

Betrachten Sie die folgende Host-Definition, die drei Vorlagen referenziert:

```
# Development web server
define host{
    use 1, 4, 8
    host_name devweb1 ...
}
```

Wenn einige dieser referenzierten Vorlagen selbst Variablen/Werte von ein oder mehreren Vorlagen erben, werden die Vorrangregeln auf der rechten Seite gezeigt.

Test, Versuch und Irrtum werden Ihnen helfen, besser zu verstehen, wie die Dinge in komplexen Vererbungssituationen wie dieser funktionieren. :-)


[Zurück](#)
[Nach oben](#)
[Weiter](#)
[Angepasste CGI-Kopf- und Fußzeilen](#)
[Zum Anfang](#)
[Zeitsparende Tricks für Objektdefinitionen](#)



Zeitsparende Tricks für Objektdefinitionen

[Zurück](#)

[Kapitel 7. Fortgeschrittene Themen](#)

[Weiter](#)

Zeitsparende Tricks für Objektdefinitionen

oder...

Einführung

Dieses Dokument versucht zu erklären, wie Sie die (etwas) versteckten Möglichkeiten von [vorlagenbasierenden Objektdefinitionen](#) ausnutzen können, um Ihren Verstand zu bewahren. Sie fragen sich wie? Verschiedene Objekttypen erlauben es Ihnen, mehrere Host-Namen und/oder Hostgruppen-Namen in Definitionen anzugeben und die Objektdefinitionen in mehrere Hosts oder Services zu "kopieren". Wir werden jeden Objekttyp, der diese Möglichkeiten unterstützt, separat behandeln. Für den Anfang sind die Objekttypen, die diese zeitsparende Möglichkeit unterstützen, wie folgt:

- [Services](#)
- [Service-Eskalationen](#)
- [Service-Abhängigkeiten](#)
- [Host-Eskalationen](#)
- [Host-Abhängigkeiten](#)
- [Hostgruppen](#)

Objekttypen, die nicht oben aufgeführt sind (z.B. Zeitfenster, Befehle usw.), unterstützen nicht die Möglichkeiten, die wir beschreiben werden.

Übereinstimmung von regulären Ausdrücken (Regular Expression Matching)

Die Beispiele, die wir unten zeigen, benutzen "Standard"-Übereinstimmung (Matching) von Objektnamen und [*erfordern*](#), dass die Option [use_regex_matching](#)*deaktiviert* ist.

Wenn Sie wollen, können Sie die Übereinstimmung von regulären Ausdrücken mit Hilfe der [use_regex_matching](#)-Konfigurationsoption aktivieren. Reguläre Ausdrücke können in jedem der Felder benutzt werden, die in den Beispielen unten benutzt werden (Hostnamen, Hostgruppen-Namen, Service-Namen und Servicegruppen-Namen).

 Anmerkung: Seien Sie vorsichtig bei der Aktivierung der Übereinstimmung von regulären Ausdrücken - es kann sein, dass Sie Ihre Konfigurationsdatei ändern müssen, weil vielleicht einige der Direktiven als reguläre Ausdrücke interpretiert werden, bei denen Sie das nicht

möchten! Probleme sollten offensichtlich werden, sobald Sie Ihre Konfiguration überprüfen.

Service-Definitionen

Mehrere Hosts:

Wenn Sie identische **Services** erzeugen möchten, die mehreren Hosts zugeordnet sind, können Sie mehrere Hosts in der *host_name*-Direktive angeben. Die folgende Definition würde einen Service namens *SOMESERVICE* auf den Hosts *HOST1* bis *HOSTN* erzeugen. Jede Instanz des *SOMESERVICE*-Service wäre identisch (d.h. hätte den gleichen Prüfbefehl, Benachrichtigungsperiode, usw.).

```
define service{
    host_name           HOST1,HOST2,HOST3,...,HOSTN
    service_description SOMESERVICE
    weitere Service-Direktiven ...
}
```

Alle Hosts in mehreren Hostgruppen:

Wenn Sie identische Services erzeugen wollen, die allen Hosts in einer oder mehreren Hostgruppen zugeordnet sind, können Sie das mit einer einzigen Service-Definition erreichen. Wie? Die *hostgroup_name*-Direktive erlaubt es Ihnen, den Namen von einer oder mehreren Hostgruppen anzugeben, für den dieser Service erzeugt werden soll. Die folgende Definition würde einen Service namens *SOMESERVICE* auf allen Hosts anlegen, die Mitglied von Hostgruppe *HOSTGROUP1* bis *HOSTGROUPN* sind. Alle Instanzen des *SOMESERVICE*-Service wären identisch (d.h. hätten den gleichen Prüfbefehl, Benachrichtigungsperiode, usw.).

```
define service{
    hostgroup_name      HOSTGROUP1,HOSTGROUP2,...,HOSTGROUPN
    service_description SOMESERVICE
    weitere Service-Direktiven ...
}
```

Alle Hosts:

Wenn Sie identische Services erzeugen wollen, die allen Hosts in Ihren Konfigurationsdateien zugeordnet sind, können Sie einen Platzhalter (wildcard) in der *host_name*-Direktive benutzen. Die folgende Definition würde einen Service namens *SOMESERVICE* auf **allen Hosts** erzeugen, die in Ihren Konfigurationsdateien definiert sind. Alle Instanzen des *SOMESERVICE*-Service wären identisch (d.h. hätten den gleichen Prüfbefehl, Benachrichtigungsperiode, usw.).

```
define service{
    host_name          *
    service_description SOMESERVICE
    weitere Service-Direktiven ...
}
```

Hosts ausschließen:

Wenn Sie identische Services auf zahlreichen Hosts anlegen, aber einige Hosts von dieser Definition ausnehmen möchten, kann dies durch das Voranstellen eines Ausrufezeichens (!) vor dem Host oder der Hostgruppe geschehen.

```
define service{
    host_name          HOST1,HOST2,!HOST3,!HOST4,...,HOSTN
    hostgroup_name     HOSTGROUP1,HOSTGROUP2,!HOSTGROUP3,!HOSTGROUP4,...,HOSTGROUPN
    service_description SOMESERVICE
    weitere Service-Direktiven ...
}
```

Service-Eskalationsdefinitionen

Mehrere Hosts:

Wenn Sie identische **Service-Eskalationen** für Services mit dem gleichen Namen/der gleichen Beschreibung erzeugen möchten, die mehreren Hosts zugeordnet sind, können Sie mehrere Hosts in der *host_name*-Direktive angeben. Die folgende Definition würde eine Service-Eskalation für Services namens *SOMESERVICE* auf den Hosts *HOST1* bis *HOSTN* erzeugen. Alle Instanzen des *SOMESERVICE*-Service wären identisch (d.h. hätten den gleichen Prüfbefehl, Benachrichtigungsperiode, usw.).

```
define serviceescalation{
    host_name           HOST1,HOST2,HOST3,...,HOSTN
    service_description SOMESERVICE
    weitere Eskalations-Direktiven ...
}
```

Alle Hosts in mehreren Hostgruppen:

Wenn Sie identische Service-Eskalationen für Services mit dem gleichen Namen/der gleichen Beschreibung erzeugen wollen, die allen Hosts in einer oder mehreren Hostgruppen zugeordnet sind, können Sie das mit der *hostgroup_name*-Direktive tun. Die folgende Definition würde eine Service-Eskalation für Services namens *SOMESERVICE* auf allen Hosts anlegen, die Mitglied von Hostgruppe *HOSTGROUP1* bis *HOSTGROUPN* sind. Alle Instanzen des *SOMESERVICE*-Service wären identisch (d.h. hätten den gleichen Prüfbefehl, Benachrichtigungsperiode, usw.).

```
define serviceescalation{
    hostgroup_name      HOSTGROUP1,HOSTGROUP2,...,HOSTGROUPN
    service_description SOMESERVICE
    weitere Eskalations-Direktiven ...
}
```

Alle Hosts:

Wenn Sie identische Service-Eskalationen für Services mit dem gleichen Namen/der gleichen Beschreibung erzeugen wollen, die allen Hosts in Ihren Konfigurationsdateien zugeordnet sind, können Sie einen Platzhalter (wildcard) in der *host_name*-Direktive benutzen. Die folgende Definition würde eine Service-Eskalation für alle Service namens *SOMESERVICE* auf **allen Hosts** erzeugen, die in Ihren Konfigurationsdateien definiert sind. Alle Instanzen des *SOMESERVICE*-Service wären identisch (d.h. hätten den gleichen Prüfbefehl, Benachrichtigungsperiode, usw.).

```
define serviceescalation{
    host_name          *
    service_description SOMESERVICE
    weitere Eskalations-Direktiven ...
}
```

Hosts ausschließen:

Wenn Sie identische Service-Eskalationen für Services auf zahlreichen Hosts anlegen, aber einige Hosts von dieser Definition ausnehmen möchten, kann dies durch das Voranstellen eines Ausrufezeichens (!) vor dem Host oder der Hostgruppe geschehen.

```
define serviceescalation{
    host_name          HOST1,HOST2,!HOST3,!HOST4,...,HOSTN
    hostgroup_name     HOSTGROUP1,HOSTGROUP2,!HOSTGROUP3,!HOSTGROUP4,...,HOSTGROUPN
    service_description SOMESERVICE
    weitere Eskalations-Direktiven ...
}
```

Alle Services auf dem gleichen Host:

Wenn Sie **Service-Eskalationen** für alle Services eines bestimmten Hosts anlegen möchten, können Sie einen Platzhalter in der *service_description*-Direktive benutzen. Die folgende Definition würde eine Service-Eskalation für *alle* Services auf Host *HOST1* erzeugen. Alle Instanzen der Service-Eskalation wären identisch (d.h. hätten die gleichen Kontaktgruppe, Benachrichtigungsintervall, usw.).

Wenn Sie sich abenteuerlustig fühlen, dann können Sie einen Platzhalter sowohl bei der *host_name*- als auch bei der *service_description*-Direktive angeben. Dadurch würden Sie eine Service-Eskalation für **alle Services** anlegen, die Sie in Ihren Konfigurationsdateien definiert haben.

```
define serviceescalation{
    host_name           HOST1
    service_description *
    weitere Eskalations-Direktiven ...
}
```

Mehrere Services auf dem gleichen Host:

Wenn Sie **Service-Eskalationen** für mehrere Services eines bestimmten Hosts anlegen möchten, können Sie mehr als eine Service-Beschreibung in der *service_description*-Direktive benutzen. Die folgende Definition würde eine Service-Eskalation für die Services *SERVICE1* bis *SERVICEN* auf Host *HOST1* erzeugen. Alle Instanzen der Service-Eskalation wären identisch (d.h. hätten die gleichen Kontaktgruppe, Benachrichtigungsintervall, usw.).

```
define serviceescalation{
    host_name           HOST1
    service_description SERVICE1,SERVICE2,...,SERVICEN
    weitere Eskalations-Direktiven ...
}
```

Alle Services in mehreren Servicegruppen:

Wenn Sie **Service-Eskalationen** für alle Services anlegen möchten, die zu einer oder mehreren Servicegruppen gehören, können Sie die *servicegroup_name*-Direktive benutzen. Die folgende Definition würde Service-Eskalationen für alle Services anlegen, die Mitglied der Servicegruppen *SERVICEGROUP1* bis *SERVICEGROUPN* sind. Alle Instanzen der Service-Eskalation wären identisch (d.h. hätten die gleichen Kontaktgruppe, Benachrichtigungsintervall, usw.).

```
define serviceescalation{
    servicegroup_name   SERVICEGROUP1,SERVICEGROUP2,...,SERVICEGROUPN
    weitere Eskalations-Direktiven ...
}
```

Service-Abhängigkeitsdefinitionen

Mehrere Hosts:

Wenn Sie **Service-Abhängigkeiten** für Services mit dem gleichen Namen/der gleichen Beschreibung erstellen möchten, die mehreren Hosts zugeordnet sind, können Sie mehrere Hosts in den *host_name*- und/oder *dependent_host_name*-Direktiven benutzen. Im folgenden Beispiel wäre Service *SERVICE2* auf den Hosts *HOST3* und *HOST4* abhängig von *SERVICE1* auf den Hosts *HOST1* und *HOST2*. Alle Instanzen der Service-Abhängigkeiten wären identisch bis auf die Host-Namen (d.h. hätten die gleichen Fehlerbenachrichtigungs-Kriterien usw.).

```
define servicedependency{
    host_name                      HOST1,HOST2
    service_description             SERVICE1
    dependent_host_name            HOST3,HOST4
    dependent_service_description   SERVICE2
    weitere Abhängigkeits-Direktiven ...
}
```

Alle Hosts in mehreren Hostgruppen:

Wenn Sie Service-Abhängigkeiten für Services mit dem gleichen Namen/der gleichen Beschreibung erstellen möchten, die allen Hosts in einer oder mehreren Hostgruppen zugeordnet sind, können Sie die *hostgroup_name*- und/oder *dependent_hostgroup_name*-Direktiven benutzen. Im folgenden Beispiel wäre Service *SERVICE2* auf allen Hosts in den Hostgruppen *HOSTGROUP3* und *HOSTGROUP4* abhängig von *SERVICE1* auf allen Hosts in den Hostgruppen *HOSTGROUP1* und *HOSTGROUP2*. Angenommen, es gibt fünf Hosts in jeder der Hostgruppen, dann wäre diese Definition äquivalent zur Definition von 100 einzelnen Service-Abhängigkeitsdefinitionen! Alle Instanzen der Service-Abhängigkeiten wären identisch bis auf die Host-Namen (d.h. hätten die gleichen Fehlerbenachrichtigungs-Kriterien usw.).

```
define servicedependency{
    hostgroup_name                  HOSTGROUP1,HOSTGROUP2
    service_description              SERVICE1
    dependent_hostgroup_name        HOSTGROUP3,HOSTGROUP4
    dependent_service_description   SERVICE2
    weitere Abhängigkeits-Direktiven ...
}
```

Alle Services auf einem Host:

Wenn Sie Service-Abhängigkeiten für alle Services eines bestimmten Hosts erstellen möchten, können Sie einen Platzhalter in den *service_description*- und/oder *dependent_service_description*-Direktiven benutzen. Im folgenden Beispiel wären **alle Services** auf Host *HOST2* abhängig von **allen Services** auf Host *HOST1*. Alle Instanzen der Service-Abhängigkeiten wären identisch (d.h. hätten die gleichen Fehlerbenachrichtigungs-Kriterien usw.).

```
define servicedependency{
    host_name                      HOST1
    service_description             *
    dependent_host_name            HOST2
    dependent_service_description  *
    weitere Abhängigkeits-Direktiven ...
}
```

Mehrere Services auf einem Host:

Wenn Sie Service-Abhängigkeiten für mehrere Services eines bestimmten Hosts erstellen möchten, können Sie mehr als eine Service-Beschreibung in den *service_description*- und/oder *dependent_service_description*-Direktiven wie folgt angeben:

```
define servicedependency{
    host_name                      HOST1
    service_description             SERVICE1,SERVICE2,...,SERVICEN
    dependent_host_name            HOST2
    dependent_service_description  SERVICE1,SERVICE2,...,SERVICEN
    weitere Abhängigkeits-Direktiven ...
}
```

Alle Services in mehreren Servicegruppen:

Wenn Sie Service-Abhängigkeiten für alle Services erstellen möchten, die einer oder mehreren Servicegruppen zugeordnet sind, können Sie die *servicegroup_name*- und/oder *dependent_servicegroup_name*-Direktiven wie folgt benutzen:

```
define servicedependency{
    servicegroup_name           SERVICEGROUP1,SERVICEGROUP2,...,SERVICEGROUPN
    dependent_servicegroup_name SERVICEGROUP3,SERVICEGROUP4,...SERVICEGROUPN
    weitere Abhängigkeits-Direktiven ...
}
```

Abhängigkeiten des gleichen Hosts:

Wenn Sie Service-Abhängigkeiten für mehrere Services erstellen möchten, die von Services auf dem gleichen Host abhängig sind, lassen Sie die *dependent_host_name*- und *dependent_hostgroup_name*-Direktiven leer. Im folgenden Beispiel wird angenommen, dass den Hosts *HOST1* und *HOST2* mindestens die folgenden vier Services zugeordnet sind: *SERVICE1*, *SERVICE2*, *SERVICE3* und *SERVICE4*. In diesem Beispiel sind *SERVICE3* und *SERVICE4* auf *HOST1* abhängig von *SERVICE1* und *SERVICE2* auf *HOST1*. Ähnlich sind *SERVICE3* und *SERVICE4* auf *HOST2* abhängig von *SERVICE1* und *SERVICE2* auf *HOST2*.

```
define servicedependency{
    host_name                  HOST1,HOST2
    service_description         SERVICE1,SERVICE2
    dependent_service_description SERVICE3,SERVICE4
    weitere Abhängigkeits-Direktiven ...
}
```

Abhängigkeiten des gleichen Hosts mit Servicegruppen:

Wenn Sie Service-Abhängigkeiten für alle Services erstellen möchten, die zu einer oder mehreren Servicegruppen eines Service gehören, der auf dem gleichen Host wie der abhängige Service läuft, lassen Sie die *host_name*- und *hostgroup_name*-Direktiven leer. Im folgenden Beispiel wird angenommen, dass Hosts mit Services aus den Servicegruppen *SERVICEGROUP1* und *SERVICEGROUP2* auch folgender Service zugeordnet ist: *SERVICE1*. In diesem Beispiel sind alle Service aus den Servicegruppen *SERVICEGROUP1* und *SERVICEGROUP2* abhängig von *SERVICE1*, der auf dem gleichen Host läuft wie der abhängige Service.

```
define servicedependency{
    service_description          SERVICE1
    dependent_service_description SERVICEGROUP1,SERVICEGROUP2
    other dependency directives ...
}
```

Host-Eskalationsdefinitionen

Mehrere Hosts:

Wenn Sie **Host-Eskalationen** für mehrere Hosts erstellen möchten, können Sie mehrere Hosts in der *host_name*-Direktive angeben. Die folgende Definitione würde eine Host-Eskalation für die Hosts *HOST1* bis *HOSTN* anlegen. Alle Instanzen der Host-Eskalation wären identisch (d.h. hätten die gleichen Kontaktgruppen, Benachrichtigungsintervalle usw.).

```
define hostescalation{
    host_name                  HOST1,HOST2,HOST3,...,HOSTN
    weitere Eskalations-Direktiven ...
}
```

Alle Hosts in mehreren Hostgruppen:

Wenn Sie Host-Eskalationen für alle Hosts in einer oder mehreren Hostgruppen erstellen möchten, können Sie die *hostgroup_name*-Direktive benutzen. Die folgende Definition würde eine Host-Eskalation für alle Hosts anlegen, die Mitglieder der Hostgruppen *HOSTGROUP1* bis *HOSTGROUPN* sind. Alle Instanzen der Host-Eskalation wären identisch (d.h. hätten die gleichen Kontaktgruppen, Benachrichtigungsintervalle usw.).

```
define hostescalation{
    hostgroup_name           HOSTGROUP1,HOSTGROUP2,...,HOSTGROUPN
    weitere Eskalations-Direktiven ...
}
```

Alle Hosts:

Wenn Sie identische Host-Eskalationen für alle Hosts erstellen wollen, die in Ihren Konfigurationsdateien definiert sind, können Sie einen Platzhalter in der *host_name*-Direktive benutzen. Die folgende Definition würde eine Host-Eskalation für alle Hosts anlegen, die in Ihren Konfigurationsdateien definiert sind. Alle Instanzen der Host-Eskalation wären identisch (d.h. hätten die gleichen Kontaktgruppen, Benachrichtigungsintervalle usw.).

```
define hostescalation{
    host_name               *
    weitere Eskalations-Direktiven ...
}
```

Hosts ausschließen:

Wenn Sie identische Host-Eskalationen auf zahlreichen Hosts oder Hostgruppen erstellen, aber einige Hosts von der Definition ausschließen möchten, kann dies durch das Voranstellen eines Ausrufezeichens (!) vor dem Host oder der Hostgruppe geschehen.

```
define hostescalation{
    host_name               HOST1,HOST2,!HOST3,!HOST4,...,HOSTN
    hostgroup_name          HOSTGROUP1,HOSTGROUP2,!HOSTGROUP3,!HOSTGROUP4,...,HOSTGROUPN
    weitere Eskalations-Direktiven ...
}
```

Host-Abhängigkeitsdefinitionen

Mehrere Hosts:

Wenn Sie **Host-Abhängigkeiten** für mehrere Hosts erstellen möchten, können Sie mehrere Hosts in den *host_name*- und/oder *dependent_host_name*-Direktiven angeben. Die folgende Definition wäre äquivalent mit der Erstellung von sechs einzelnen Host-Abhängigkeiten. Im obigen Beispiel wären die Hosts *HOST3*, *HOST4* und *HOST5* abhängig von den Hosts *HOST1* und *HOST2*. Alle Instanzen der Host-Abhängigkeiten wären identisch bis auf die Host-Namen (d.h. sie hätten die gleichen Fehlerbenachrichtigungs-Kriterien, usw.).

```
define hostdependency{
    host_name               HOST1,HOST2
    dependent_host_name     HOST3,HOST4,HOST5
    weitere Abhängigkeits-Direktiven ...
}
```

Alle Hosts in mehreren Hostgruppen:

Wenn Sie Host-Abhängigkeiten für alle Hosts in einer oder mehreren Hostgruppen erstellen möchten, können Sie die *hostgroup_name*- und/oder *dependent_hostgroup_name*-Direktiven benutzen. Im folgenden Beispiel wären alle Hosts in den Hostgruppen *HOSTGROUP3* und *HOSTGROUP4* abhängig von allen Hosts in den Hostgruppen *HOSTGROUP1* und *HOSTGROUP2*.

HOSTGROUP2. Alle Instanzen der Host-Abhängigkeiten wären identisch bis auf die Host-Namen (d.h. sie hätten die gleichen Fehlerbenachrichtigungs-Kriterien, usw.).

```
define hostdependency{
    hostgroup_name           HOSTGROUP1,HOSTGROUP2
    dependent_hostgroup_name HOSTGROUP3,HOSTGROUP4
    weitere Abhängigkeits-Direktiven ...
}
```

Hostgruppen

Alle Hosts:

Wenn Sie eine Hostgruppe anlegen möchten, die alle Hosts aus Ihren Konfigurationsdateien als Mitglieder enthält, können Sie einen Platzhalter in der *members*-Direktive benutzen. Die folgende Definition würde eine Hostgruppe namens *HOSTGROUP1* erstellen, die **alle Hosts** aus Ihren Konfigurationsdateien als Mitglieder enthält.

```
define hostgroup{
    hostgroup_name           HOSTGROUP1
    members                  *
    weitere Hostgruppen-Direktiven ...
}
```

[Zurück](#)
[Nach oben](#)
[Weiter](#)
[Objektvererbung](#)
[Zum Anfang](#)
[Kapitel 8. Sicherheit und Leistungsoptimierung](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Kapitel 8. Sicherheit und Leistungsoptimierung

[Zurück](#)

[Weiter](#)

Kapitel 8. Sicherheit und Leistungsoptimierung

Inhaltsverzeichnis

[Sicherheitsüberlegungen](#)

[Verbesserte CGI-Sicherheit und Authentifizierung](#)

[Icinga für maximale Leistung optimieren](#)

[Schnellstart-Optionen](#)

[Large Installation Tweaks](#)

[Nutzung des Icingastats-Utilitys](#)

[grafische Darstellung von Performance-Informationen mit PNP4Nagios](#)

[Temporäre Daten](#)

[Zurück](#)

[Weiter](#)

[Zeitsparende Tricks für
Objektdefinitionen](#)

[Zum Anfang](#)

[Sicherheitsüberlegungen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Sicherheitsüberlegungen

[Zurück](#)

Kapitel 8. Sicherheit und Leistungsoptimierung

[Weiter](#)

Sicherheitsüberlegungen

Einführung



Dies ist als ein kurzer Überblick einiger Dinge gedacht, die Sie bei der Installation von Icinga im Hinterkopf behalten sollten, um es in einer sicheren Weise aufzusetzen.

Ihr Überwachungsrechner sollte als eine Hintertür in Ihre anderen System betrachtet werden. In vielen Fällen wird dem Icinga-Rechner der Zugriff auf Firewalls gewährt, um entfernte Server zu überwachen. In den meisten Fällen ist die Abfrage von verschiedenen Informationen der entfernten Server erlaubt. Überwachenden Servern wird ein gewisses Maß an Vertrauen entgegen gebracht, damit sie entfernte Systeme abfragen können. Das bietet einem potenziellen Angreifer eine attraktive Hintertür zu Ihren Systemen. Ein Angreifer könnte es einfacher haben, in Ihre Systeme einzudringen, wenn er zuerst den Überwachungsserver kompromittiert. Das trifft besonders dann zu, wenn Sie gemeinsame SSH-Schlüssel nutzen, um entfernte Systeme zu überwachen.

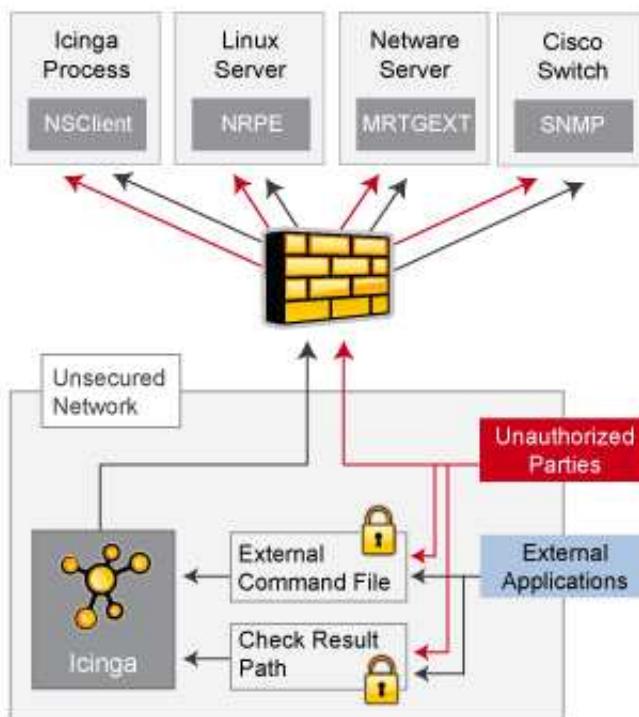
Wenn ein Eindringling in der Lage ist, Prüfergebnisse oder externe Befehle an den Icinga-Daemon zu erteilen, hat er die Möglichkeit, falsche Überwachungsdaten zu übertragen, Sie mit falschen Benachrichtigungen auf die Palme bringen oder Eventhandler-Scripte auszulösen. Wenn Sie Eventhandler-Scripte haben, die Services neu starten, Strom unterbrechen usw., dann kann das ziemlich problematisch sein.

Ein weiterer zu beachtender Bereich ist die Möglichkeit von Eindringlingen, Überwachungsdaten (Statusinformationen) zu belauschen, während sie über den Draht gehen. Wenn Übertragungskanäle nicht verschlüsselt sind, können Angreifer durch Beobachtung Ihrer Überwachungsdaten wertvolle Informationen gewinnen. Nehmen Sie als Beispiel die folgende Situation: ein Angreifer belauscht für eine gewisse Zeit die Überwachungsdaten und analysiert die typische CPU- und Plattenauslastung Ihrer Systeme zusammen mit der Zahl der Benutzer, die typischerweise angemeldet sind. Der Angreifer ist dann in der Lage, die beste Zeit für die Kompromittierung eines Systems und dessen Ressourcen (CPU usw.) zu ermitteln, ohne bemerkt zu werden.

Hier sind einige Hinweise, wie Sie Ihre Systeme sichern können, wenn Sie eine Icinga-basierte Überwachungslösung implementieren...

Optimale Verfahren

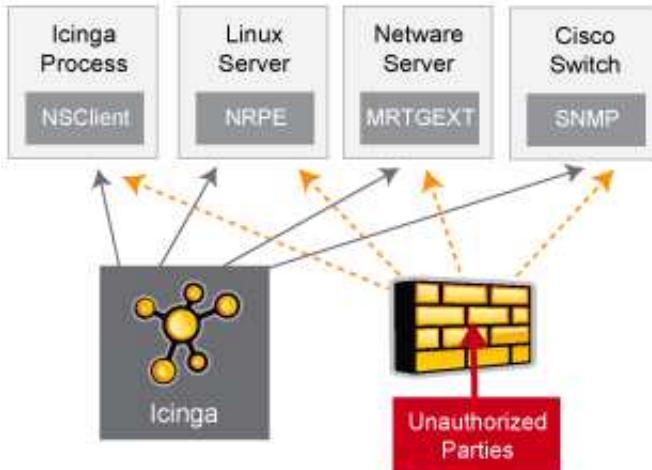
- Benutzen Sie eine eigene Überwachungs-Box.** Wir würden empfehlen, dass Sie einen Server benutzen, der nur für die Überwachung (und ggf. andere administrative Aufgaben) vorgesehen ist. Schützen Sie Ihren Überwachungsserver, als wäre es einer der wichtigsten Server Ihres Netzwerks. Halten Sie die laufenden Services auf einem Minimum und beschränken Sie den Zugang durch TCP-Wrapper, Firewalls usw. Weil der Icinga-Rechner berechtigt ist, mit Ihren Servern zu reden und vielleicht durch Ihre Firewalls zu gehen, kann es ein Sicherheitsrisiko sein, wenn Sie Benutzern Zugang zu Ihrem Überwachungsserver gewähren. Bedenken Sie, dass es einfacher ist, root-Zugang über eine Sicherheitslücke zu bekommen, wenn Sie ein lokales Benutzerkonto auf dem System haben.



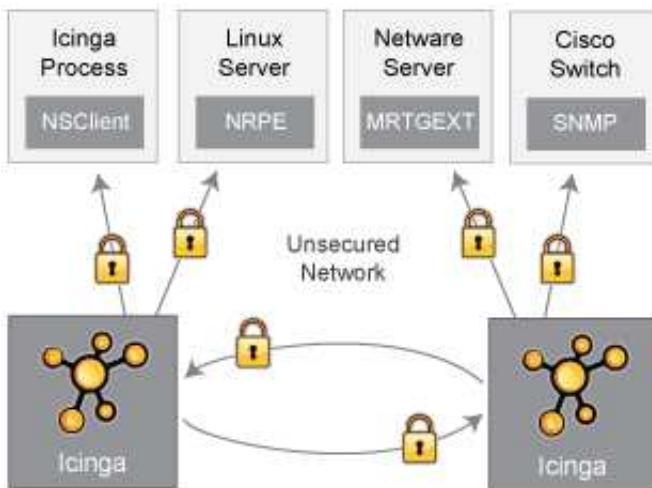
- Lassen Sie Icinga nicht als root laufen.** Icinga muss nicht als root laufen, also tun Sie es nicht. Sie können Icinga anweisen, die Berechtigungen nach dem Start zu "dropen" und mit Hilfe der `icinga_user-` und `icinga_group-`Direktiven in der Hauptkonfigurationsdatei unter anderen Benutzer- und/oder Gruppenberechtigungen zu laufen. Wenn Sie Eventhandler oder Plugins ausführen müssen, die Root-Berechtigungen benötigen, möchten Sie vielleicht `sudo` nutzen.
- Verriegeln Sie das Prüfergebnis-Verzeichnis.** Stellen Sie sicher, dass nur der `icinga`-Benutzer im `check result path` lesen und schreiben darf. Wenn andere Benutzer außer `icinga` (oder `root`) in diesem Verzeichnis schreiben dürfen, dann können sie falsche Host-/Service-Prüfergebnisse an den Icinga-Daemon senden. Dies kann zu Ärger (falschen Benachrichtigungen) oder Sicherheitsproblemen (ausgelösten Eventhandler) führen.
- Verriegeln Sie das External Command File.** Wenn Sie `externe Befehle` aktivieren, dann stellen Sie sicher, dass Sie passende Berechtigungen für das `/usr/local/icinga/var/rw`-Verzeichnis setzen. Nur der Icinga-Benutzer (normalerweise `icinga`) und der Web-Server-Benutzer (normalerweise `nobody`, `httpd`, `apache2` oder `www-data`) sollten Schreibberechtigung für das Command-File besitzen. Wenn Sie Icinga auf einer Maschine

installiert haben, die der Überwachung und administrativen Aufgaben dient, dann sollte das ausreichen. Wenn Sie es auf einer allgemeinen- oder Multi-User-Maschine installiert haben (nicht empfohlen) und dem Web-Server-Benutzer Schreibberechtigung auf das Command-File geben, kann das ein Sicherheitsproblem sein. Sie wollen schließlich nicht, dass jeder Benutzer auf Ihrem System Icinga über das External-Command-File kontrollieren kann. In diesem Fall würden wir raten, nur dem *nagios*-Benutzer Schreibberechtigung zu erlauben und etwas wie [CGIWrap](#) zu benutzen, um die CGIs als *icinga* statt als *nobody* laufen zu lassen.

5. **Fordern Sie Authentifizierung bei den CGIs.** Wir empfehlen dringend Authentifizierung für den Zugriff auf die CGIs. Sobald Sie das tun, lesen Sie die Dokumentation zu Standardberechtigungen von authentifizierten Kontakten und autorisieren Sie bestimmte Kontakte für zusätzliche Rechte nur, wenn es nötig ist. Eine Anleitung zur Einrichtung von Authentifizierung und Autorisierung finden Sie [hier](#). Wenn Sie mit der [use_authentication](#)-Direktive die Authentifizierung in der CGI-Konfigurationsdatei deaktivieren, wird das [command CGI](#) das Schreiben jeglicher Befehle in das [external command file](#) verweigern. Sie wollen schließlich nicht, dass alle Welt in der Lage ist, Icinga zu kontrollieren, oder?
6. **Benutzen Sie absolute Pfade in Befehlsdefinitionen.** Wenn Sie Befehle definieren, benutzen Sie den *absoluten Pfad* (keinen relativen) für Scripte oder Programm, die Sie ausführen.
7. **Verstecken Sie sensible Daten mit \$USERn\$-Makros.** Die CGIs lesen die [Hauptkonfigurationsdatei](#) und die [Objekt-Konfigurationsdatei\(en\)](#), so dass Sie dort keine sensiblen Informationen (Benutzernamen, Passwörter, usw.) ablegen sollten. Wenn Sie Benutzernamen und/oder Passwörter in einer Befehlsdefinition angeben müssen, dann nutzen Sie ein [\\$USERn\\$-Makro](#), um sie zu verstecken. \$USERn\$-Makros werden in einer oder mehreren [Ressourcen-Dateien](#) definiert. Die CGIs werden nicht versuchen, den Inhalt von Ressourcen-Dateien zu lesen, so dass Sie restriktivere Berechtigungen (600 oder 660) dafür benutzen können. Betrachten Sie die Beispiel-*resource.cfg*-Datei im Basisverzeichnis der Icinga-Distribution für ein Beispiel, wie \$USERn\$-Makros zu definieren sind.
8. **Entfernen Sie gefährliche Zeichen aus Makros.** Benutzen Sie die [illegal_macro_output_chars](#)-Direktive, um gefährliche Zeichen aus den \$HOSTOUTPUT\$-, \$SERVICEOUTPUT\$-, \$HOSTPERFDATA\$- und \$SERVICEPERFDATA\$-Makros zu entfernen, bevor sie in Benachrichtigungen usw. benutzt werden. Gefährliche Zeichen kann alles sein, was ggf. durch die Shell interpretiert wird und dadurch eine Sicherheitslücke öffnet. Ein Beispiel dafür sind Backtick-Zeichen (`) in den \$HOSTOUTPUT\$, \$SERVICEOUTPUT\$, \$HOSTPERFDATA\$ und /oder \$SERVICEPERFDATA\$-Makros, die es einem Angreifer erlauben, einen beliebigen Befehl als Icinga-Benutzer auszuführen (ein guter Grund, Icinga NICHT als root-Benutzer laufen zu lassen).
9. **Sicherer Zugang zu entfernten Agenten.** Verriegeln Sie den Zugang zu Agenten (NRPE, NSClient, SNMP, usw.) auf entfernten Systemen durch Firewalls, Zugangsliste usw. Sie wollen nicht, dass jeder Ihre Systeme nach Statusinformationen abfragt. Diese Informationen können durch einen Angreifer genutzt werden, um entfernte Eventhandler-Scripte auszuführen oder die beste Zeit zu ermitteln, um nicht beobachtet zu werden.



10. **Sichere Kommunikationskanäle.** Stellen Sie sicher, dass Sie die Kommunikationskanäle zwischen verschiedenen Icinga-Installationen und Ihren Überwachungskanälen verschlüsseln, wann immer möglich. Sie wollen nicht, dass jemand Statusinformationen belauscht, die über Ihr Netzwerk gehen. Diese Informationen können durch einen Angreifer genutzt werden, um die beste Zeit für einen unbeobachteten Zugang zu ermitteln.

[Zurück](#)[Nach oben](#)[Weiter](#)

Kapitel 8. Sicherheit und
Leistungsoptimierung

[Zum Anfang](#)

Verbesserte CGI-Sicherheit und
Authentifizierung



Verbesserte CGI-Sicherheit und Authentifizierung

[Zurück](#)

[Kapitel 8. Sicherheit und Leistungsoptimierung](#)

[Weiter](#)

Verbesserte CGI-Sicherheit und Authentifizierung

Einführung



Dies ist als eine Einführung für die Implementierung stärkerer Authentifizierung und Server-Sicherheit bezogen auf das CGI-Web-Interface gedacht.

Es gibt viele Wege, die Sicherheit Ihres Überwachungs-Servers und des Icinga-Umfeldes zu verbessern. Dies sollte nicht als das Ende aller Bemühungen angesehen werden. Nehmen Sie es statt dessen als eine Einführung für einige der Techniken, die Sie nutzen können, um die Sicherheit Ihres Systems zu verstärken. Wie immer sollten Sie forschen und die besten Techniken nutzen, die verfügbar sind. Behandeln Sie Ihren Überwachungs-Server, als wäre es der wichtigste Server in Ihrem Netzwerk und Sie werden belohnt werden.

Zusätzliche Techniken

- **Stärkere Authentifizierung durch Digest Authentication.** Wenn Sie den [Schnellstartanleitungen](#) gefolgt sind, werden Sie wahrscheinlich Apaches [Basic Authentication](#) nutzen. "Basic Authentication" wird Benutzer und Passwort bei jedem HTTP-Request im Klartext übertragen. Ziehen Sie eine sicherere Authentifizierungsmethode wie z.B. [Digest Authentication](#) in Betracht, die aus Ihrem Benutzernamen und Passwort einen MD5-Hash erzeugt, der bei jeder Anfrage gesendet wird.
- **Erzwingen von TLS/SSL für jede Web-Kommunikation.** Apache bietet [TLS/SSL](#) durch das [mod_ssl](#)-Modul. TLS/SSL bietet einen sicheren Tunnel zwischen Client und Server, der Abhören und Verfälschung durch starke publickey/privatekey-Kryptographie verhindert.
- **Beschränken Sie Apache mit Hilfe von Zugangskontrollen.** Überlegen Sie, ob Sie den Zugang zur Icinga-Box auf Ihre IP-Adresse, IP-Adressbereich oder IP-Subnetz beschränken. Wenn Sie Zugang von außen auf Ihr Netzwerk benötigen, können Sie VPN und SSH-Tunnel nutzen. Es ist einfach, den Zugang zu Ihrem System auf HTTP/HTTPS zu begrenzen.

Implementieren der Digest Authentication

Die Implementierung der Digest Authentication ist einfach. Dazu müssen Sie den neuen Typ der Passwort-Datei mit dem '[htdigest](#)'-Tool anlegen, dann die Apache-Konfiguration für Icinga anpassen (typischerweise /etc/httpd/conf.d/icinga.conf).

Legen Sie eine neue Passwort-Datei mit dem '[htdigest](#)'-Tool an. Den Unterschied, den Sie feststellen werden, wenn Sie mit dem '[htpasswd](#)'-Tool vertraut sind, ist die Anforderung, ein 'Realm'-Parameter anzugeben. In diesem Fall bezieht sich 'realm' auf den Wert der 'AuthName'-Direktive in der Apache-Konfiguration.

```
htdigest -c /usr/local/icinga/etc/.digest_pw "Icinga Access" icingaadmin
```

Als nächstes editieren Sie die Apache-Konfigurationsdatei für Icinga (typischerweise /etc/httpd/conf.d/icinga.conf) mit Hilfe des folgenden Beispiels:

```
## BEGIN APACHE CONFIG SNIPPET - Icinga.CONF
ScriptAlias /icinga/cgi-bin "/usr/local/icinga/sbin"
<Directory "/usr/local/icinga/sbin">
    Options ExecCGI
    AllowOverride None
    Order allow,deny
    Allow from all
    AuthType Digest
    AuthName "Icinga Access"
    AuthDigestFile /usr/local/icinga/etc/.digest_pw
    Require valid-user
</Directory>
Alias /icinga "/usr/local/icinga/share"
<Directory "/usr/local/icinga/share">
    Options None
    AllowOverride None
    Order allow,deny
    Allow from all
    AuthType Digest
    AuthName "Icinga Access"
    AuthDigestFile /usr/local/icinga/etc/.digest_pw
    Require valid-user
</Directory>
## END APACHE CONFIG SNIPPETS
```

Danach starten Sie den Apache-Service, damit die neuen Einstellungen aktiv werden können.

```
/etc/init.d/httpd restart
```

Implementieren erzwungener TLS/SSL-Kommunikation

Stellen Sie sicher, dass Sie Apache und OpenSSL installiert haben. Normalerweise sollten Sie [mod_ssl](#)-Unterstützung haben. Falls Sie trotzdem Schwierigkeiten haben, finden Sie ggf. Hilfe durch das Lesen von Apaches [TLS/SSL Encryption Documentation](#).

Als nächstes prüfen Sie durch den Aufruf des Icinga-Web-Interfaces über HTTPS (<https://your.domain/Icinga>), dass die TLS/SSL-Unterstützung funktioniert. Wenn es funktioniert, können Sie mit den nächsten Schritten fortfahren, die die Nutzung von HTTPS erzwingen und alle HTTP-Anfragen an das Icinga-Web-Interface blockiert. Wenn Sie Schwierigkeiten haben, lesen Sie bitte Apaches [TLS/SSL Encryption Documentation](#) und nutzen Sie [Google](#) für die Suche nach Lösungen zu Ihrer Apache-Installation.

Danach editieren Sie die Apache-Konfigurationsdatei für Icinga (typischerweise /etc/httpd/conf.d/icinga.conf) und fügen Sie den 'sbin'- und 'share'-Verzeichnissen die 'SSLRequireSSL'-Direktive hinzu.

```
## BEGIN APACHE CONFIG SNIPPET - Icinga.CONF
ScriptAlias /icinga/cgi-bin "/usr/local/icinga/sbin"
<Directory "/usr/local/icinga/sbin">
  ...
  SSLRequireSSL
  ...
</Directory>
Alias /icinga "/usr/local/icinga/share"
<Directory "/usr/local/icinga/share">
  ...
  SSLRequireSSL
  ...
</Directory>
## END APACHE CONFIG SNIPPETS
```

Danach starten Sie den Apache-Service, damit die neuen Einstellungen aktiv werden können.

```
/etc/init.d/httpd restart
```

Implementieren von IP-Subnetz-Beschränkung

Das folgende Beispiel zeigt, wie Sie den Zugang auf die Icinga-CGIs auf eine bestimmte IP-Adresse, einen IP-Adressbereich oder ein IP-Subnetz mit Hilfe von Apaches [Access Controls](#) beschränken.

Danach editieren Sie die Apache-Konfigurationsdatei für Icinga (typischerweise `/etc/httpd/conf.d/icinga.conf`) und fügen Sie die 'Allow'-, 'Deny'- und 'Order'-Direktiven hinzu. Dazu folgendes Beispiel:

```
## BEGIN APACHE CONFIG SNIPPET - Icinga.CONF
ScriptAlias /icinga/cgi-bin "/usr/local/icinga/sbin"
<Directory "/usr/local/icinga/sbin">
  ...
  AllowOverride None
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1 10.0.0.25          # Allow single IP addresses
  Allow from 10.0.0.0/255.255.255.0       # Allow network/netmask pair
  Allow from 10.0.0.0/24                   # Allow network/nnn CIDR spec
  ...
</Directory>
Alias /icinga "/usr/local/icinga/share"
<Directory "/usr/local/icinga/share">
  ...
  AllowOverride None
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1 10.0.0.25          # Allow single IP addresses
  Allow from 10.0.0.0/255.255.255.0       # Allow network/netmask pair
  Allow from 10.0.0.0/24                   # Allow network/nnn CIDR spec
  ...
</Directory>
## END APACHE CONFIG SNIPPET
```

Wichtige Anmerkungen

- **Digest Authentication sendet Daten im Klartext, aber nicht Ihren Benutzernamen und Passwort .**
- **Digest Authentication ist nicht ganz so gut unterstützt wie Basic Authentication .**

- **TLS/SSL hat das Potential für einen "Man-in-the-middle-Angriff".** MITM-Angriffe machen verletzbar, wenn ein Angreifer in der Lage ist, sich zwischen Server und Client zu schieben wie bei einem Phishing-Angriff, ISP-Monitoring oder Resignierung von Unternehmens-LAN Firewall-Zertifikaten. Bitte machen Sie sich kundig zu Zertifikats-Verifikation!
 - **Apache Access Controls schützen nur die HTTP/HTTPS-Protokolle .** Sehen Sie sich [IPtables](#) für eine starke systemweite Firewall-Kontrolle an.
 - **Am wichtigsten: Sicherheit ist ein bewegliches Ziel, also bleiben Sie informiert und forschen Sie !** Vielleicht durch das Anhören eines Podcasts wie z.B. "[Security Now!](#)".
-

[Zurück](#)[Nach oben](#)[Weiter](#)[Sicherheitsüberlegungen](#)[Zum Anfang](#)[Icinga für maximale Leistung optimieren](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Icinga für maximale Leistung optimieren

[Zurück](#)
[Kapitel 8. Sicherheit und Leistungsoptimierung](#)
[Weiter](#)

Icinga für maximale Leistung optimieren

Einführung



Jetzt haben Sie Icinga endlich eingerichtet und lauffähig und nun wollen Sie wissen, wie man ein wenig daran drehen kann. Die Leistung von Icinga zu optimieren kann notwendig sein, wenn Sie eine große Zahl (> 1.000) von Hosts und Services haben. Hier ein paar Dinge, nach denen Sie schauen können, um Icinga zu optimieren...

Optimierungshinweise:

1. **Stellen Sie Performance-Statistiken mit PNP4Nagios dar.** Um zu verfolgen, wie die Last Ihrer Icinga-Installation aussieht und welche Auswirkungen Ihre Konfigurationsänderungen darauf haben, sollten Sie verschiedene wichtige Statistiken mit PNP4Nagios darstellen. Das ist wirklich sehr, sehr sinnvoll, wenn es um die Leistungsoptimierung einer Icinga-Installation geht. Informationen, wie das zu tun ist, finden Sie [hier](#).
2. **Benutzen Sie "Verbesserungen für große Installationen"** (large installation tweaks). Das Aktivieren der `use_large_installation_tweaks`-Option kann Ihnen bessere Leistung bringen. Lesen Sie [hier](#) mehr darüber, was diese Option tut.
3. **Deaktivieren Sie Umgebungs-Makros.** Makros werden Prüfungen, Benachrichtigungen, Eventhandler usw. normalerweise über Umgebungsvariablen zur Verfügung gestellt. Das kann in einer großen Icinga-Installation zu einem Problem werden, weil es zusätzlichen Speicher (und wichtiger) mehr CPU verbraucht. Wenn Ihre Scripte nicht über Umgebungsvariablen auf Makros zugreifen (d.h., wenn Sie alle benötigen Makros in der Kommandozeile übergeben), dann brauchen Sie dieses Feature nicht. Sie können über die `enable_environment_macros`-Option einstellen, ob Makros als Umgebungsvariablen verfügbar sind.

4. **Prüfergebnis-Ernterhythmus** (Check Result Reaper Frequency). Die [check_result_reaper_frequency](#)-Variable legt fest, wie oft Icinga prüfen soll, ob Host- und Service-Ergebnisse verarbeitet werden müssen. Die maximale Zeit, die es zur Verarbeitung solcher Ergebnisse benötigen darf, ist durch die maximale Erntezeit (max reaper time) festgelegt (siehe unten). Wenn Ihr Ernterhythmus zu hoch (zu selten) ist, könnten Sie hohe Latenzzeiten für Host- und Service-Prüfungen sehen.
5. **maximale Erntezeit** (Max Reaper Time). Die [max_check_result_reaper_time](#)-Variable legt die maximale Zeit fest, die der Icinga-Daemon für die Verarbeitung der Ergebnisse von Host- und Service-Prüfungen verbringen darf, bevor er sich anderen Dingen zuwendet - wie z.B. dem Ausführen von neuen Host- und Service-Prüfungen. Ein zu hoher Wert kann zu hohen Latenzzeiten bei Ihren Host- und Service-Prüfungen führen. Ein zu niedriger Wert kann den gleichen Effekt haben. Wenn Sie zu hohe Latenzzeiten haben, dann passen Sie diesen Wert an und sehen Sie, welchen Effekt das hat. [Graphisch dargestellte Statistiken](#) helfen Ihnen bei der Auswertung der Auswirkungen.
6. **Anpassen der Pufferwerte.** Gegebenenfalls müssen Sie den Wert der [external_command_buffer_slots](#)-Option anpassen. Die graphische Analyse mit [PNP4Nagios](#) (siehe oben) zeigt Ihnen, welche Werte Sie für diese Option nutzen sollten.
7. **Prüfen Sie Service-Latzenzeiten, um den besten Wert für die maximale Anzahl von gleichzeitigen Prüfungen zu ermitteln.** Icinga kann die Anzahl von gleichzeitig ausgeführten Prüfungen durch die [max_concurrent_checks](#)-Option begrenzen. Das ist gut, weil es Ihnen etwas Kontrolle darüber gibt, wieviel Last Icinga auf Ihrem Überwachungsrechner erzeugt, aber es kann auch die Dinge verlangsamen. Wenn Sie für die Mehrzahl Ihrer Service-Prüfungen hohe Latenzzeiten sehen (> 10 oder 15 Sekunden), dann enthalten Sie Icinga Prüfungen vor, die es braucht. Das ist nicht der Fehler von Icinga - es ist Ihrer. Unter idealen Bedingungen hätten alle Service-Prüfungen eine Latenzzeit von 0, was bedeutet, dass alle Prüfungen zu der Zeit stattfinden, für die sie geplant sind. Allerdings ist es normal, dass einige Prüfungen kleine Latenzzeiten haben. Wir würden empfehlen, die niedrigste Zahl der meisten gleichzeitigen Prüfungen zu nehmen, wenn Sie Icinga mit der [-s](#)-Option starten und diesen Wert zu verdoppeln. Erhöhen Sie diesen Wert dann soweit, bis die durchschnittlichen Latenzzeiten für Service-Prüfungen ziemlich niedrig ist. Mehr Informationen zur Planung von Service-Prüfungen finden Sie [hier](#).
8. **Nutzen Sie passive Prüfungen, wenn möglich.** Der nötige Overhead, um die Ergebnisse von [passiven Service-Prüfungen](#) zu verarbeiten, ist viel niedriger als bei "normalen" aktiven Prüfungen, also machen Sie Gebrauch von dieser Information, wenn Sie eine Menge von Services überwachen. Es sollte angemerkt werden, dass passive Prüfungen nur dann wirklich sinnvoll sind, wenn Sie irgendeine externe Applikation haben, die überwachen oder berichten kann; wenn also Icinga all die Arbeit machen muss, ist das nicht hilfreich.
9. **Vermeiden Sie interpretierte Plugins.** Etwas, was spürbar die Last Ihres Überwachungs-Hosts senkt, ist die Nutzung von kompilierten (C/C++, usw.) Plugins statt interpretierter Scripts (Perl, usw.). Während Perl und ähnliches einfach zu schreiben ist und gut läuft, kann die Tatsache, dass es bei jeder Ausführung kompiliert/interpretiert werden muss, zu einer spürbaren Steigerung der Last Ihres Überwachungs-Hosts führen, wenn Sie eine Menge von Service-Prüfungen haben. Wenn Sie Perl-Plugins nutzen wollen, dann überlegen Sie, ob Sie diese nicht mit perlcc(1) (einem Utility, das Teil der Standard-Perl-Distribution ist) zu einem richtigen Programm umwandeln oder Icinga mit eingebettetem Perl-Interpreter kompilieren (siehe unten).
10. **Nutzen Sie den eingebetteten Perl-Interpreter.** Wenn Sie eine Menge von Perl-Skripten für Prüfungen benutzen, dann werden Sie vielleicht feststellen, dass das Kompilieren des [eingebetteten Perl-Interpreters](#) (embedded Perl interpreter) in das Icinga-Binary die Dinge beschleunigt.

11. **Optimieren Sie Host-Prüfbefehle.** Wenn Sie Host-Zustände mit dem check_ping-Plugin prüfen, dann werden Sie feststellen, dass die Host-Prüfungen viel schneller durchgeführt werden, wenn Sie diese abbrechen. Statt einen *max_attempts*-Wert von 1 anzugeben und mit dem check_ping-Plugins 10 ICMP-Pakete an den Host zu schicken, wäre es viel schneller, den *max_attempts*-Wert auf 10 zu setzen und jedes Mal nur ein ICMP-Paket zu senden. Das liegt daran, dass Icinga den Zustand eines Hosts oft nach der Ausführung eines Plugins feststellen kann, so dass Sie die erste Prüfung so schnell wie möglich machen sollten. Diese Methode hat in einigen Situationen ihre Fallstricke (z.B. Hosts, die langsam reagieren, könnten als "down" angesehen werden), aber wir denken, dass Sie schnellere Host-Prüfungen sehen werden, wenn Sie sie benutzen. Eine weitere Möglichkeit wäre, statt check_ping ein schnelleres Plugin (z.B. check_fping) als *host_check_command* zu benutzen.
12. **Planen Sie regelmäßige Host-Prüfungen.** Regelmäßige Host-Prüfungen zu planen kann tatsächlich die Leistung von Icinga steigern. Das liegt an der Art, wie die [Zwischenspeicher-Prüflogik](#) (cached check logic) arbeitet (siehe unten). Um regelmäßige Prüfungen eines Hosts zu planen, setzen Sie die *check_interval*-Direktive in der [Host-Definition](#) auf einen Wert größer als Null.
13. **Aktivieren Sie zwischengespeicherte Host-Prüfungsergebnisse** (cached host checks). Host-Prüfungen nach Bedarf können von der Zwischenspeicherung (caching) profitieren. Host-Prüfungen nach Bedarf werden ausgeführt, wenn Icinga einen Service-Zustandswechsel feststellt. Diese Prüfungen nach Bedarf werden ausgeführt, wenn Icinga wissen will, ob der mit dem Service verbundene Host den Zustand gewechselt hat. Durch die Aktivierung von zwischengespeicherten Host-Prüfungsergebnissen können Sie die Leistung optimieren. In einigen Fällen könnte Icinga in der Lage sein, den alten/zwischengespeicherten Zustand des Hosts zu benutzen, statt eine Host-Prüfung auszuführen. Das kann die Dinge beschleunigen und die Last des Überwachungsservers reduzieren. Damit zwischengespeicherte Prüfungen effektiv sind, müssen Sie regelmäßige Prüfungen für Ihre Hosts planen (siehe oben). Mehr Informationen zu zwischengespeicherten Prüfungen finden Sie [hier](#).
14. **Nutzen Sie keine aggressiven Host-Prüfungen.** Solange Sie keine Probleme damit haben, dass Icinga Host-Erholungen nicht korrekt erkennt, würden wir empfehlen, die [use_aggressive_host_checking](#)-Option nicht zu aktivieren. Wenn diese Option abgeschaltet ist, werden Host-Prüfungen viel schneller ausgeführt, was zu schnellerer Ausführung von Service-Prüfungen führt. Allerdings können Host-Erholungen unter bestimmten Umständen übersehen werden, wenn sie ausgeschaltet ist. Wenn sich z.B. der Host erholt, aber alle mit ihm verbundenen Services in einem nicht-OK-Zustand bleiben (und nicht zwischen verschiedenen nicht-OK-Zuständen "kippeln"), dann könnte Icinga übersehen, dass sich der Host erholt hat. Einige wenige Leute könnten diese Option aktivieren, aber die Mehrheit nicht und wir würden empfehlen, sie nicht zu aktivieren, solange Sie nicht glauben, dass Sie sie benötigen...
15. **Optimierung externer Befehle.** Wenn Sie eine Menge externer Befehle verarbeiten (d.h. passive Prüfungen in einer [verteilten Umgebung](#)), dann wollen Sie vielleicht die *command_check_interval*-Variable auf -1 setzen. Das bewirkt, dass Icinga so oft wie möglich auf externe Befehle prüft. Sie sollten außerdem überlegen, die Anzahl verfügbarer [externer Befehlpuffer](#) zu erhöhen. Puffer werden benutzt, um externe Befehle zu speichern, die (durch einen separaten Thread) aus dem [external command file](#) gelesen werden, bevor sie vom Icinga-Daemon verarbeitet werden. Wenn Ihr Icinga-Daemon eine Menge von passiven Prüfungen oder externen Befehlen empfängt, dann könnten Sie in eine Situation kommen, in der immer alle Puffer voll sind. Das führt zu blockierenden Kind-Prozessen (externe Scripte, NSCA-Daemon usw.), wenn sie versuchen, in das "external command file" zu schreiben. Wir würden sehr empfehlen, dass Sie die Nutzung von externen Befehlpuffern graphisch mit Hilfe von PNP4Nagios und dem icingastats-Utility darstellen, wie es [hier](#) beschrieben ist, so dass Sie die typische externe Befehlpuffernutzung Ihrer Icinga-Installation sehen.

16. **Optimieren Sie die Hardware für maximale Leistung.** Hinweis: Hardware-Leistung sollte kein Thema sein, solange Sie nicht 1) Tausende von Services überwachen, 2) eine Menge von Nachverarbeitung von Performance-Daten usw. machen. Ihre Systemkonfiguration und Ihre Hardware-Ausstattung werden direkt beeinflussen, was Ihr Betriebssystem leistet, so dass sie beeinflussen, was Icinga leistet. Die häufigste Hardware-Optimierung betrifft die Festplatte(n). CPU und Speichergeschwindigkeit sind offensichtliche Faktoren, die die Leistung beeinflussen, aber der Plattenzugriff wird Ihr größter Flaschenhals sein. Speichern Sie Plugins, das Status-Log usw. nicht auf langsamen Platten (d.h. alte IDE-Platten oder NFS-Mounts). Wenn Sie sie haben, dann nutzen Sie UltraSCSI- oder schnelle IDE-Platten. Ein wichtiger Hinweis für IDE/Linux-Benutzer ist, dass viele Linux-Installationen nicht versuchen, den Plattenzugriff zu optimieren. Wenn Sie die Plattenzugriffsparameter nicht ändern (z.B. mit einem Utility wie **hdparam**), werden Sie eine **Menge** der schnellen Features der neuen IDE-Platten verlieren.
17. **Benutzen Sie eine RAM-Disk für temporäre Daten .** Verschiedene Dateien werden sehr oft angelegt und verarbeitet. Das betrifft u.a. den aktuellen Zustand, der im **status file** gespeichert wird und die laufende Konfiguration, die im **object cache file** abgelegt ist. Um physikalischen I/O zu reduzieren, ist es ratsam, diese Daten auf einer RAM-Disk abzulegen. Datenverlust durch einen Stromausfall oder etwas ähnliches ist nicht kritisch, weil diese beiden Dateien bei jedem (Re-)Start von Icinga neu erzeugt werden. Das Anlegen einer RAM-Disk und die Änderungen an der Hauptkonfigurationsdatei werden **hier** beschrieben.

[Zurück](#)[Nach oben](#)[Weiter](#)[Verbesserte CGI-Sicherheit und Authentifizierung](#)[Zum Anfang](#)[Schnellstart-Optionen](#)



Schnellstart-Optionen

[Zurück](#)

Kapitel 8. Sicherheit und Leistungsoptimierung

[Weiter](#)

Schnellstart-Optionen

Einführung

Es gibt einige Dinge, die Sie tun können, um die Zeit zu verringern, die Icinga zum (Neu-)Start benötigt. Diese Beschleunigung umfasst u.a. Änderungen bei der Verarbeitung Ihrer Konfigurationsdateien.

Diese Techniken zu benutzen ist besonders dann sinnvoll, wenn bei Ihnen einer oder mehrere der folgenden Punkte zutreffen:

- große Konfigurationen
- komplexe Konfigurationen (massiver Einsatz von Template-Features)
- Installationen, bei denen häufige Neustarts notwendig sind

Hintergrund

Bei jedem (erneuten) Start von Icinga müssen die Konfigurationsdateien verarbeitet werden, bevor die Überwachung beginnen kann. Dieser Konfigurationsanlaufprozess umfasst eine Reihe von Schritten:

- Lesen der Konfigurationsdateien
- Auflösen von Template-Definitionen
- "Recombobulating" Ihrer Objekte (ein [ausgedachter] Begriff für die verschiedenen Arten von Arbeiten, die auftreten)
- duplizieren von Objektdefinitionen
- vererben von Objekteigenschaften
- sortieren Ihrer Objektdefinitionen
- überprüfen der Objektbeziehungsintegrität
- prüfen von zirkulären Pfaden
- und mehr...

Einige dieser Schritte können ziemlich zeitintensiv sein, wenn Sie große oder komplexe Konfigurationen haben. Gibt es einen Weg, einen dieser Schritte zu beschleunigen? Ja!

Bewertung von Anlaufzeiten

Bevor wir weitermachen, die Dinge zu beschleunigen, müssen wir sehen was möglich ist und ob wir uns mit der ganzen Sache beschäftigen sollten oder nicht. Das ist einfach - starten Sie Icinga mit der **-s** oder **--test-scheduling**-Option, um Zeiten und Planungsinformationen zu bekommen.

Beginnend mit Icinga 1.0.2 gibt es eine zusätzliche Option **-S** oder **--show-scheduling**. Damit erhalten Sie weitere Informationen zur Scheduling Queue.

Ein Beispiel für die Ausgabe (gekürzt, um nur relevante Teile zu zeigen) sehen Sie nachfolgend. In diesem Beispiel nutzen wir eine Icinga-Konfigurations mit 25 Host und etwas mehr als 10.000 Services.

```
#> /usr/local/icinga/bin/icinga -s /usr/local/icinga/etc/icinga.cfg
Icinga 1.6
Copyright (c) 1999-2007 Ethan Galstad (http://www.nagios.org/)
Last Modified: 01-27-2007
License: GPL
Timing information on object configuration processing is listed
below. You can use this information to see if precaching your
object configuration would be useful.
Object Config Source: Config files (uncached)
OBJECT CONFIG PROCESSING TIMES      (* = Potential for precache savings with -u option)
-----
Read:          0.486780 sec
Resolve:       0.004106 sec *
Recomb Contactgroups: 0.000077 sec *
Recomb Hostgroups:   0.000172 sec *
Dup Services:    0.028801 sec *
Recomb Servicegroups: 0.010358 sec *
Duplicate:      5.666932 sec *
Inherit:        0.003770 sec *
Recomb Contacts: 0.030085 sec *
Sort:           2.648863 sec *
Register:       2.654628 sec
Free:            0.021347 sec
=====
TOTAL:         11.555925 sec * = 8.393170 sec (72.63%) estimated savings
Timing information on configuration verification is listed below.
CONFIG VERIFICATION TIMES      (* = Potential for speedup with -x option)
-----
Object Relationships: 1.400807 sec
Circular Paths:      54.676622 sec *
Misc:                0.006924 sec
=====
TOTAL:         56.084353 sec * = 54.676622 sec (97.5%) estimated savings
```

Okay, lassen Sie uns ansehen was passiert ist. Wenn wir die Summen ansehen, dauerte es ungefähr **11,6** Sekunden, die Konfigurationsdateien zu verarbeiten und weitere **56** Sekunden, die Konfigurations zu verifizieren. Das bedeutet, dass es fast **68 Sekunden** dauert, bis die erste Überwachung beginnen kann! Das ist nicht akzeptierbar, wenn wir Icinga ziemlich regelmäßig neu starten müssem.

Was kann man daran ändern? Werfen Sie einen erneuten Blick auf die Ausgabe und Sie sehen, dass Icinga schätzt, dass wir etwa **8,4** Sekunden bei der Verarbeitung der Konfiguration und weitere **54,7** bei der Verifizierung einsparen können. Icinga denkt, dass wir **63 Sekunden** der normalen Anlaufzeit sparen können, wenn einige Optimierungen vorgenommen werden.

Wow! Von **68 Sekunden** auf gerade mal **5 Sekunden**? Yep, lesen Sie weiter, um zu sehen, wie das geht.

Pre-Caching der Objektkonfiguration

Icinga kann einige Zeit beim analysieren Ihrer Konfigurationsdateien verbringen, besonders dann, wenn Sie Template-Features wie z.B. Vererbung usw. nutzen. Um die Zeit der Analyse Ihrer Konfiguration zu verringern, können Sie Icinga veranlassen, Ihre Konfigurationsdateien für die Zukunft vorzuverarbeiten (pre-process) und vor-zwischenzuspeichern (pre-cache).

Wenn Sie Icinga mit der **-p**-Kommandozeilenoption starten, wird Icinga Ihre Konfigurationsdateien einlesen, verarbeiten und sie in einer vor-zwischengespeicherten (pre-cached) (durch die [precached_object_file](#)-Direktive angegebene) Konfigurationsdatei sichern. Diese Konfigurationsdatei enthält vorverarbeitete Konfigurationseinträge, die Icinga in Zukunft einfacher/schneller verarbeiten kann.

Sie müssen die **-p**-Kommandozeilenoption zusammen mit der **-v** oder **-s**-Kommandozeilenoption benutzen, wie nachfolgend gezeigt. Dies stellt sicher, dass Ihre Konfiguration überprüft wird, bevor die precached-Datei erstellt wird.

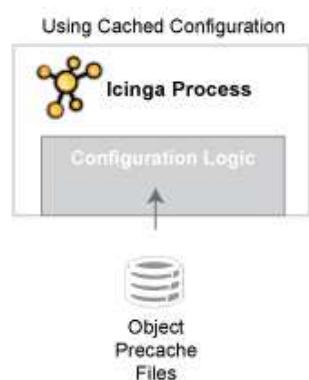
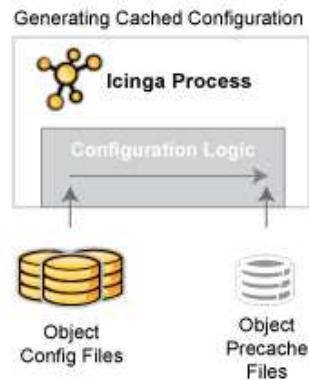
```
#> /usr/local/icinga/bin/icinga -pv /usr/local/icinga/etc/icinga.cfg
```

Die precached-Konfigurationsdatei wird wahrscheinlich um einiges größer sein als die Summe Ihrer Objektkonfigurationsdateien. Das ist normal und beabsichtigt.

Sobald die precached-Objektkonfigurationdatei erstellt wurde, können Sie Icinga starten und mit der **-u**-Kommandozeilenoption angeben, dass diese Datei statt Ihrer Konfigurationsdatei(en) benutzt werden soll.

```
#> /usr/local/icinga/bin/icinga -ud /usr/local/icinga/etc/icinga.cfg
```

 Wenn Sie Ihre Konfigurationsdateien ändern, müssen Sie diese erneut überprüfen und die precached-Konfigurationsdatei neu erstellen, bevor Sie Icinga erneut starten. Wenn Sie die precached-Konfigurationsdatei nicht neu generieren, wird Icinga Ihre alte Konfiguration benutzen, weil die precached-Konfigurationsdatei gelesen wird und nicht Ihre geänderten Konfigurationsdateien.



Überspringen der Test von zirkulären Pfaden

Der zweite (und zeitintensivste) Teil der Konfigurationsanlaufphase ist die Prüfung auf zirkuläre Pfade. Im obigen Beispiel dauerte es fast eine Minute, um diesen Schritt der Konfigurationsprüfung auszuführen.

Was ist diese zirkuläre-Pfad-Prüfung und warum dauert sie so lange? Die zirkuläre-Pfad-Prüfung soll verhindern, dass Sie zirkuläre Pfade in Ihren Host-, Host-Abhängigkeits- oder Service-Abhängigkeitsdefinitionen haben. Wenn ein zirkulärer Pfad in Ihren Konfigurationsdateien existiert, könnte Icinga in einer Deadlock-Situation enden. Der wahrscheinlichste Grund dafür, dass die Prüfung so lange dauert, dürfte darin liegen, dass wir keinen effizienten Algorithmus benutzen. Ein effizienterer Algorithmus wäre daher willkommen. Wink: das bedeutet, dass alle Absolventen der Computerwissenschaften, die ihre Thesen zu Icinga gemalt haben, ein wenig Code liefern könnten. :-)

Wenn Sie die Prüfung auf zirkuläre Pfade überspringen möchten, wenn Sie Icinga starten, dann fügen Sie die **-x**-Option wie folgt hinzu:

```
#> /usr/local/icinga/bin/icinga -xd /usr/local/icinga/etc/icinga.cfg
```



Es ist von äußerster Wichtigkeit, dass Sie Ihre Konfiguration überprüfen, bevor Sie Icinga (erneut) starten, wenn Sie auf die Prüfung auf zirkuläre Pfade verzichten. Wenn Sie es nicht tun, kann dies zu Deadlocks führen. Sie sind gewarnt worden.

Alles zusammenfassen

Folgen Sie diesen Schritten, wenn Sie mögliche Beschleunigungen durch pre-Caching Ihrer Konfiguration und überspringen der Prüfungen auf zirkuläre Pfade nutzen wollen.

1. Überprüfen Sie Ihre Konfiguration und legen Sie die precache-Datei mit den folgenden Befehlen an:

```
#> /usr/local/icinga/bin/icinga -vp /usr/local/icinga/etc/icinga.cfg
```

2. Stoppen Sie Icinga, wenn es momentan läuft.

3. Starten Sie Icinga wie folgt, um die precached-Konfigurationsdatei zu nutzen und auf Prüfung auf zirkuläre Pfade zu überspringen:

```
#> /usr/local/icinga/bin/icinga -uxd /usr/local/icinga/etc/icinga.cfg
```

4. Wenn Sie in Zukunft Ihre Konfigurationsdateien verändern und Icinga erneut starten müssen, damit diese Änderungen aktiv werden, dann wiederholen Sie Schritt 1, um Ihre Konfiguration erneut zu überprüfen und die precached-Konfigurationsdatei zu erstellen. Sobald das getan ist, können Sie Icinga über das Web-Interface oder durch das Senden eines SIGHUP-Signals neustarten. Wenn Sie die precached-Objektdatei nicht neu erstellen, wird Icinga wieder Ihre alte Konfiguration benutzen, weil es die precached-Datei liest statt Ihrer Konfigurationsdateien.

5. Das war's! Erfreuen Sie sich am Geschwindigkeitsgewinn beim Start.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Icinga für maximale Leistung
optimieren

[Zum Anfang](#)

Large Installation Tweaks

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Large Installation Tweaks

[Zurück](#)

Kapitel 8. Sicherheit und Leistungsoptimierung

[Weiter](#)

Large Installation Tweaks

Einführung

Benutzer mit großen Icinga-Installation können von der [use_large_installation_tweaks](#)-Konfigurationsoption profitieren. Das Aktivieren dieser Option erlaubt es dem Icinga-Daemon, bestimmte Abkürzungen zu nehmen, die in geringerer Systembelastung und besserer Leistung resultieren.

Effekte

Wenn Sie die [use_large_installation_tweaks](#)-Option in Ihrer Icinga-Hauptkonfigurationsdatei aktivieren, werden mehrere Anpassungen gemacht, wie der Icinga-Daemon arbeitet:

1. **Keine Zusammenfassungsmakros in Umgebungsvariablen** - Die [Zusammenfassungsmakros](#) werden Ihnen nicht als Umgebungsvariablen zur Verfügung stehen. Die Berechnung der Werte dieser Makros kann in großen Konfigurationen ziemlich zeitintensiv sein, so dass sie nicht als Umgebungsvariablen zur Verfügung stehen, wenn Sie diese Option benutzen. Zusammenfassungsmakros sind weiterhin als reguläre Makros verfügbar, wenn Sie diese Ihren Scripts als Parameter übergeben.
2. **Unterschiedliche Speicherbereinigung** - Normalerweise wird Icinga den allokierten Speicher in Kind-Prozessen freigeben, bevor sie enden. Dies ist wahrscheinlich die beste Vorgehensweise, aber vielleicht in großen Installationen unnötig, weil die meisten Betriebssysteme selbst darauf achten, allokierten Speicher freizugeben, wenn Prozesse enden. Das Betriebssystem neigt dazu, belegten Speicher schneller freizugeben, als Icinga das kann, so dass Icinga nicht versucht, Speicher in Kind-Prozessen freizugeben, wenn Sie diese Option aktivieren.
3. **Weniger fork()** - Normalerweise wird Icinga zweimal fork() aufrufen, wenn es Host- und Service-Prüfungen ausführt. Das wird getan, um (1) ein hohes Maß an Resistenz sicherzustellen gegen Plugins, die fehlschlagen und einen SegFault erzeugen und (2) dafür sorgen, dass das Betriebssystem sich um die Bereinigung der Enkel-Prozesse kümmert, sobald sie enden. Der zusätzliche fork() ist nicht wirklich nötig, so dass er übersprungen wird, wenn Sie diese Option aktivieren. Als Ergebnis werden Kind-Prozesse von Icinga selbst bereinigt (anstatt diese Aufgabe dem Betriebssystem zu überlassen). Dieses Feature sollte für spürbare Lasteinsparungen in Ihrer Icinga-Installation sorgen.

[Zurück](#)

Schnellstart-Optionen

[Nach oben](#)

[Zum Anfang](#)

[Weiter](#)

Nutzung des Icingastats-Utilitys

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Nutzung des Icingastats-Utilitys

[Zurück](#)

Kapitel 8. Sicherheit und Leistungsoptimierung

[Weiter](#)

Nutzung des Icingastats-Utilitys

Einführung

Ein Utility namens **icingastats** ist in der Icinga-Distribution enthalten. Es wird zusammen mit dem Icinga-Daemon kompiliert und installiert. Das icingastats-Utility liefert Ihnen verschiedene Informationen zu einem laufenden Icinga-Prozess, die sehr hilfreich bei der [Leistungsoptimierung](#) sein können. Sie können Informationen in einem menschlich-lesbaren oder im Performance-Daten-kompatiblen Format erhalten.

Gebrauchshinweise

Sie können das *icingastats*-Utility mit der **--help**-Option starten, um Gebrauchshinweise zu bekommen.

menschlich-lesbare Ausgabe

Um menschlich-lesbare Informationen zur Leistung eines laufenden Icinga-Prozesses zu erhalten, starten Sie das *icingastats*-Utility mit dem **-c**-Kommandozeilenargument, um die Position Ihrer Hauptkonfigurationsdatei wie folgt anzugeben:

```
[icinga@monitoring ~]# /usr/local/icinga/bin/icingastats -c /usr/local/icinga/etc/nagios.cfg
Icinga Stats 1.6
Copyright (c) 2009 Nagios Core Development Team and Community Contributors
Copyright (c) 1999-2009 Ethan Galstad
Last Modified: 02-16-2011
License: GPL
CURRENT STATUS DATA
-----
Status File:                      /usr/local/icinga/var/status.dat
Status File Age:                  0d 0h 0m 27s
Status File Version:               1.3.0

Program Running Time:             0d 14h 28m 16s
Icinga PID:                      21182
Used/High/Total Command Buffers:  0 / 3 / 4096

Total Services:                  1001
Services Checked:                945
Services Scheduled:              950
Services Actively Checked:       1000
Services Passively Checked:      1
Total Service State Change:       0.000 / 100.000 / 1.881 %
Active Service Latency:          0.000 / 285.165 / 25.045 sec
Active Service Execution Time:   0.000 / 304.925 / 0.834 sec
Active Service State Change:     0.000 / 100.000 / 1.883 %
Active Services Last 1/5/15/60 min: 20 / 191 / 471 / 926
Passive Service Latency:          0.862 / 0.862 / 0.862 sec
```

```

Passive Service State Change:          0.000 / 0.000 / 0.000 %
Passive Services Last 1/5/15/60 min: 1 / 1 / 1 / 1
Services Ok/Warn/Unk/Crit:           816 / 56 / 51 / 78
Services Flapping:                  39
Services In Downtime:               0

Total Hosts:                         111
Hosts Checked:                      104
Hosts Scheduled:                     104
Hosts Actively Checked:             111
Host Passively Checked:              0
Total Host State Change:            0.000 / 100.000 / 10.574 %
Active Host Latency:                0.000 / 279.257 / 21.700 sec
Active Host Execution Time:         0.000 / 6.405 / 0.432 sec
Active Host State Change:           0.000 / 100.000 / 10.574 %
Active Hosts Last 1/5/15/60 min:   17 / 50 / 74 / 104
Passive Host Latency:               0.000 / 0.000 / 0.000 sec
Passive Host State Change:          0.000 / 0.000 / 0.000 %
Passive Hosts Last 1/5/15/60 min:  0 / 0 / 0 / 0
Hosts Up/Down/Unreach:              89 / 7 / 15
Hosts Flapping:                     22
Hosts In Downtime:                 0

Active Host Checks Last 1/5/15 min: 73 / 97 / 246
  Scheduled:                        13 / 21 / 50
  On-demand:                        60 / 76 / 196
  Parallel:                          45 / 63 / 171
  Serial:                            0 / 0 / 0
  Cached:                            28 / 34 / 75
Passive Host Checks Last 1/5/15 min: 0 / 0 / 0
Active Service Checks Last 1/5/15 min: 142 / 192 / 501
  Scheduled:                        142 / 192 / 500
  On-demand:                         0 / 0 / 1
  Cached:                            0 / 0 / 0
Passive Service Checks Last 1/5/15 min: 6 / 6 / 15

External Commands Last 1/5/15 min:    6 / 6 / 15

```

[icinga@monitoring ~]#

Wie Sie sehen können, zeigt das Utility ein Reihe von verschiedenen Metriken zum Icinga-Prozess an. Metriken mit mehreren Werten sind (wenn nicht anders angegeben) Minimum-, Maximum- und Durchschnittswerte für die betreffende Metrik.

PNP4Nagios-Integration

Sie können das *icingastats*-Utility benutzen, um verschiedene Icinga-Metriken mit PNP4Nagios (oder anderen kompatiblen Programmen) anzuzeigen. Um das zu tun, starten Sie das *icingastats*-Utility mit den **--mrtg**- und **--data**-Optionen. Die **--data**-Option wird benutzt, um anzugeben, welche Statistiken dargestellt werden sollen. Mögliche Werte für die **--data**-Option finden Sie durch Start des *icingastats*-Utilities mit der **--help**-Option.

 Anmerkung: Informationen zum Gebrauch des *icingastats*-Utilitys zu Generierung von PNP4Nagios-Grafiken zu Darstellung von Icinga-Leistungsstatistiken finden Sie [hier](#).

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Large Installation Tweaks

[Zum Anfang](#)

grafische Darstellung von
Performance-Informationen mit
PNP4Nagios

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



grafische Darstellung von Performance-Informationen mit PNP4Nagios

[Zurück](#)

[Kapitel 8. Sicherheit und Leistungsoptimierung](#)

[Weiter](#)

grafische Darstellung von Performance-Informationen mit PNP4Nagios

Einführung

Das [icingastats](#)-Utility erlaubt Ihnen zusammen mit [PNP4Nagios](#), verschiedene Icinga-Performance-Statistiken über eine bestimmten Zeitraum grafisch darzustellen. Das ist wichtig, weil es Ihnen helfen kann

- dass Icinga effizient arbeitet
- um Problembereiche im Überwachungsprozess zu lokalisieren
- um die Einflüsse von Änderungen in Ihrer Icinga-Konfiguration zu beobachten

Voraussetzungen

PNP4Nagios ist eines der populärsten Addons wegen der einfachen Installation und geringem Wartungsaufwand während des Betriebs. Die Dokumentation zusammen mit weiteren Links zum Download der Software finden Sie unter <http://docs.pnp4nagios.org/de/pnp-0.6/start>.

[check_nagiosstats](#) wurde von [Jochen Bern](#) erstellt. Es wird über die crontab aufgerufen und liefert die Daten als passive Prüfergebnisse. Trotz des Namens funktioniert das Plugin auch mit Icinga.

- Nach dem Herunterladen des Plugins und Ablegen im Plugin-Verzeichnis (z.B. /usr/local/icinga/libexec, falls Sie die Schnellstartanleitung benutzt haben) müssen Sie die Werte im Konfigurationsabschnitt des Scripts anpassen.
 - Am **wichtigsten** ist "EXEC=/path/to/icingastats" (z.B. /usr/local/icinga/bin/icingastats), das auf das icingastats-Binary zeigen muss.
 - Abhängig von Ihren Bedürfnissen möchten Sie ggf. den Wert für CUMULATE von "AVG" auf "MIN" oder "MAX" ändern: Die Einstellung von TIMEFRAME beeinflusst die Zeitperiode, die für die Ausgabe von kumulierten Werten benutzt wird
 - Das Ändern der Werte von PASSIVE_EMERGENCY_HOST und PASSIVE_EMERGENCY_SERVICE sollte nicht notwendig sein, weil diese Werte als Parameter an das Script übergeben werden.

- Stellen Sie sicher, dass Ihre Objektkonfigurationsdateien eine passende Service-Definition enthalten, wie z.B.

```
define service{
    host_name           <the Icinga server>
    service_description icingastats # (or something appropriate)
    active_checks_enabled 0
    check_command       check_dummy!0
    ...
}
```

Vergessen Sie nicht den Neustart von Icinga nach dieser Änderung.

- Legen Sie einen logischen Link im templates-Verzeichnis von PNP4Nagios an

```
$> ln -s ../templates.dist/nagiostats.php icingastats.php
```

Stellen Sie sicher, dass *icingastats* (ohne die Endung .php) zu dem Wert passt, den Sie in der Service-Definition angegeben haben. Leerzeichen in der Service-Beschreibung müssen durch Unterstriche (_) im Dateinamen ersetzt werden (z.B. "Icinga Stats" --> "Icinga_Stats.php")

- Fügen Sie eine Zeile zur crontab des Icinga-Benutzers hinzu, die das *icingastats*-Binary aufruft und die Ergebnisse an die Command-Pipe weiterleitet

```
* * * * * /usr/local/icinga/libexec/check_nagiostats --passive <host> <service> >> /usr/local/icinga/var/icinga.cmd
```

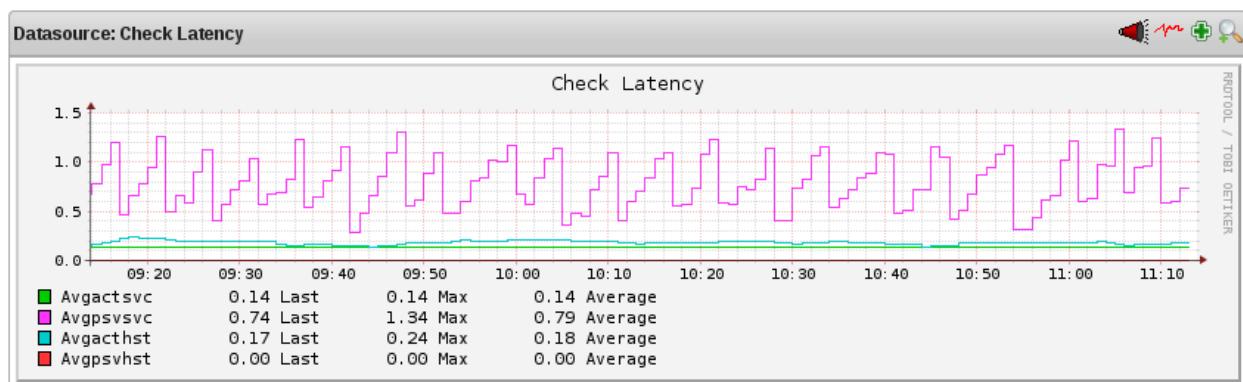
Auf diese Weise werden die Werte in regelmäßigen Intervallen aktualisiert.

Beispiel-Graphen

Wir werden beschreiben, was die durch `check_nagiostats` erzeugten Graphen bedeuten und wofür sie benutzt werden können...

Durchschnittliche Host-/Service-Prüfungslatenz

Abbildung 8.1. Durchschnittliche Host-/Service-Prüfungslatenz



Dieser Graph zeigt die durchschnittlichen Latenzzeiten von Hosts und Services über die Zeit gesehen, getrennt nach aktiven und passiven Prüfungen. Das ist nützlich zum Verständnis von:

- Host-Prüfungen
- Service-Prüfungen

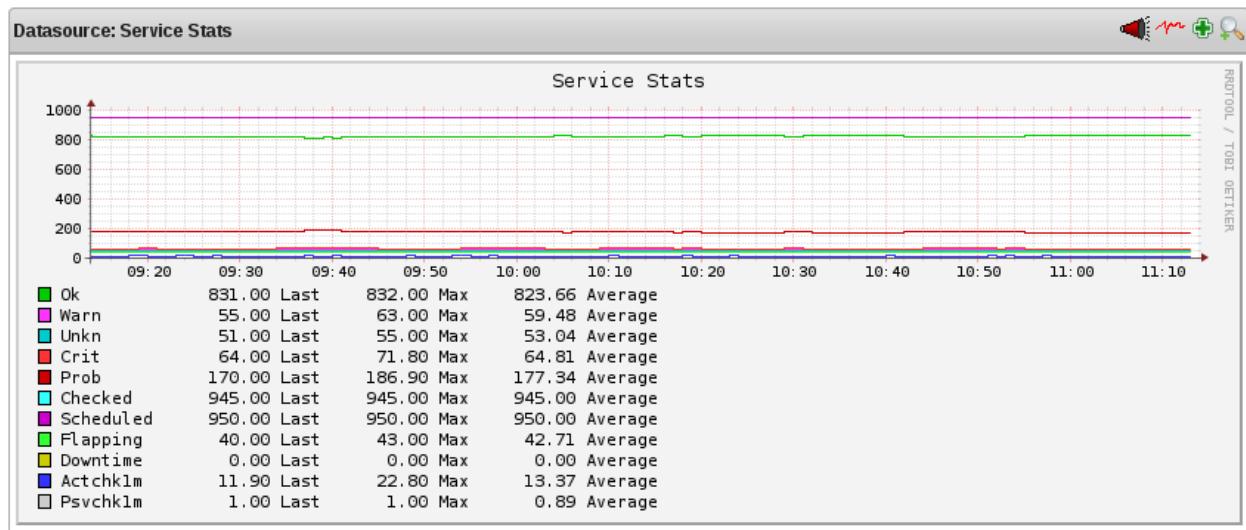
- Aktiven Prüfungen
- Passiven Prüfungen
- Performance-Tuning

Durchgehend hohe Latenzen können ein Hinweis darauf sein, dass eine oder mehrere der folgenden Variablen angepasst werden sollten:

- `max_concurrent_checks`
- `check_result_reaper_frequency`
- `max_check_result_reaper_time`

Service-Statistiken

Abbildung 8.2. Service-Statistiken

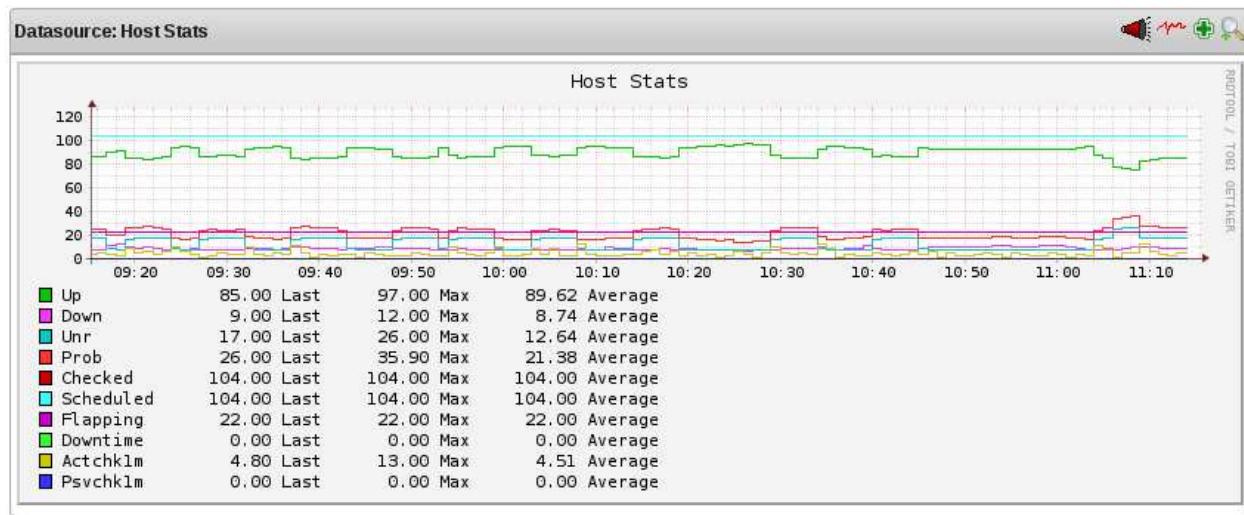


Dieser Graph zeigt die Werte für die einzelnen Service-Zustände zusammen mit der durchschnittlichen Zahl von geprüften Services an, die aktiv bzw. passiv in der von Ihnen angegebenen Zeitperiode geprüft wurden. Das ist nützlich zum Verständnis von:

- Service-Prüfungen
- Vorausschauenden Service-Abhängigkeitsprüfungen (predictive service dependency checks)
- Zwischengespeicherten Prüfungen (cached checks)
- Flattererkennung (flap detection)

Host-Statistiken

Abbildung 8.3. Host-Statistiken

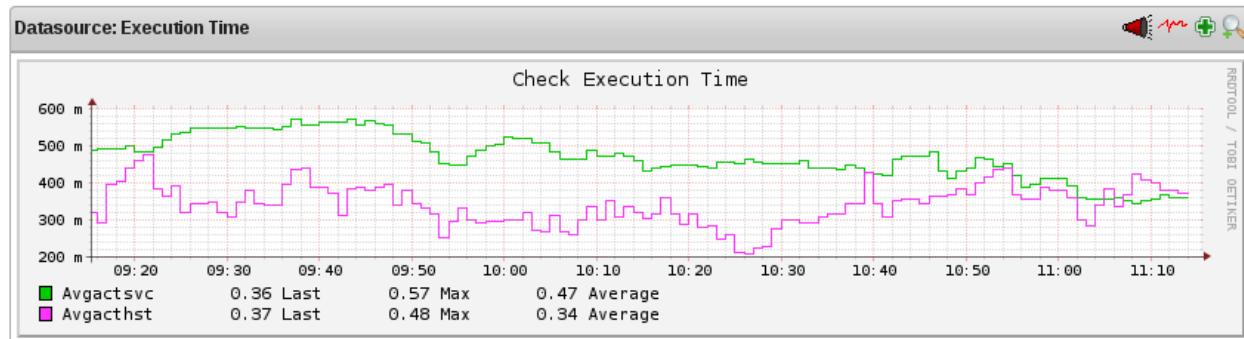


Dieser Graph zeigt die Werte für die einzelnen Host-Zustände zusammen mit der durchschnittlichen Zahl von geprüften Hosts an, die aktiv bzw. passiv in der von Ihnen angegebenen Zeitperiode geprüft wurden. Das ist nützlich zum Verständnis von:

- Host-Prüfungen
- Vorausschauenden Host-Abhängigkeitsprüfungen (predictive host dependency checks)
- Zwischengespeicherten Prüfungen (cached checks)
- Flattererkennung (flap detection)

Durchschnittliche Ausführungszeiten

Abbildung 8.4. Durchschnittliche Ausführungszeiten



Dieser Graph zeigt die durchschnittlichen Ausführungszeit von Host- und Service-Prüfungen über die Zeit gesehen. Das ist nützlich zum Verständnis von:

- Host-Prüfungen
- Service-Prüfungen
- Performance-Tuning



Anmerkung

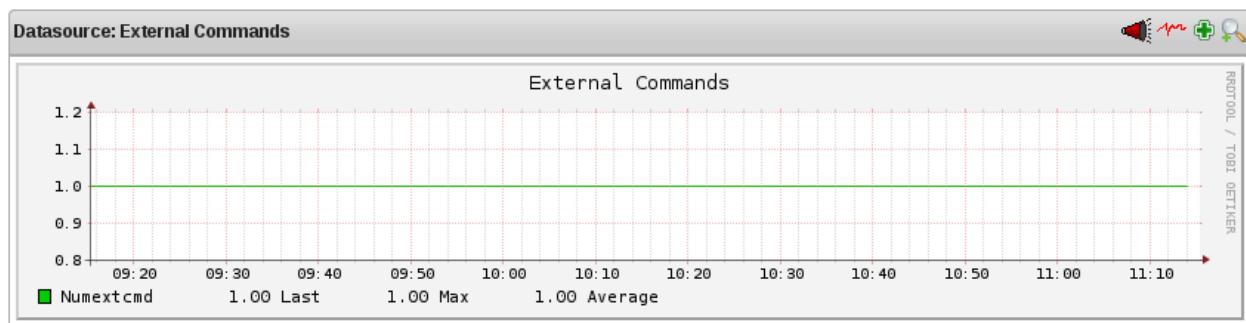
Um ehrlich zu sein: Wir haben die Graphen ein wenig verändert, bezogen auf die Farben. Gelb ist teilweise schwierig vom Hintergrund zu unterscheiden so dass wir einige Zeilen in der PNP4Nagios-Template-Datei `template.dist/nagiostats.php` von `$i=0;` in `$i=1;` geändert haben.

Zusätzliche Graphen

Nun ja, wir haben das Template noch ein bisschen mehr verändert, weil das Plugin zwar die Daten liefert, aber keine dazugehörigen Graphen. (Beim Blick in das Template ist es aber sehr anzupassen, falls Sie die folgenden Graphen wirklich benötigen.)

Externe Befehle (external commands)

Abbildung 8.5. Externe Befehle

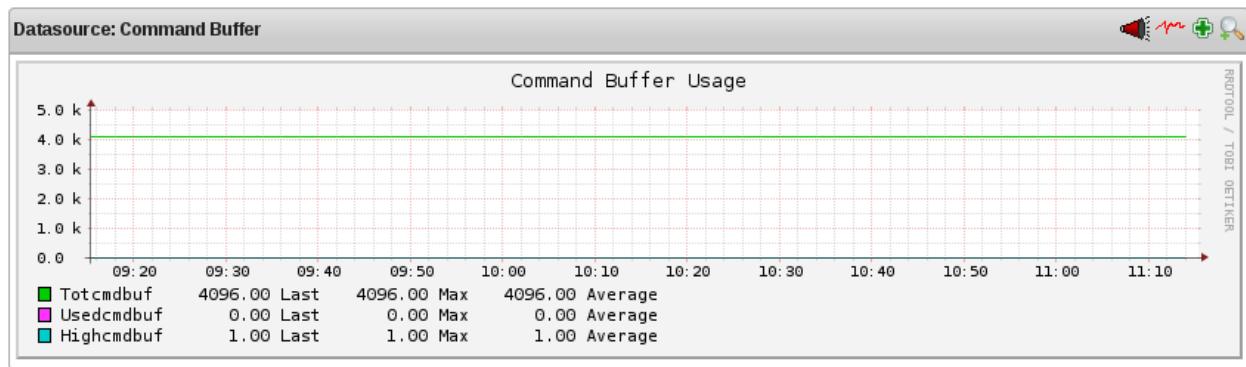


Dieser Graph zeigt, wie viele externe Befehle vom Icinga-Daemon über die Zeit gesehen verarbeitet wurden. Solange Sie keine große Anzahl von externen Befehlen verarbeiten (wie z.B. im Falle einer verteilten Überwachungsumgebung), dann kann dieser Graph fast leer sein. Die Überwachung von externen Befehlen kann nützlich sein für das Verständnis der Auswirkung von:

- [Passiven Prüfungen](#)
- [Verteilter Überwachung](#)
- [Redundante/Failover-Überwachung](#)

Puffer für externe Befehle (external command buffers)

Abbildung 8.6. Puffer für externe Befehle

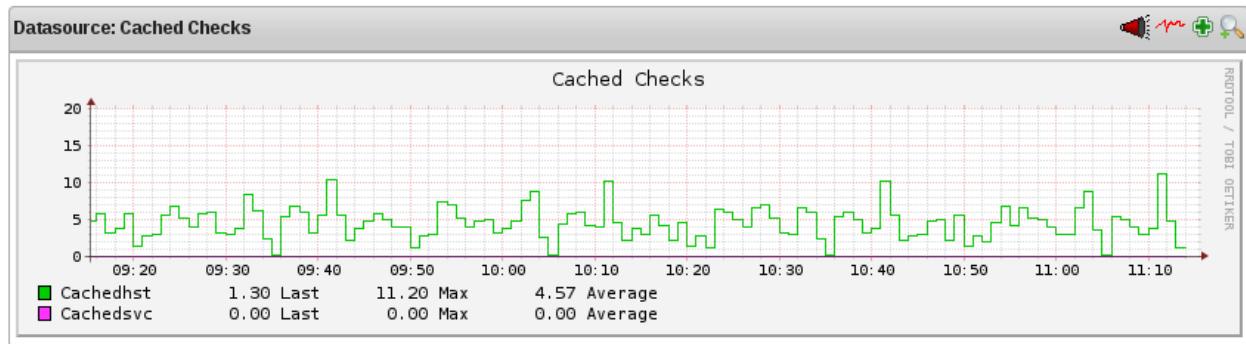


Der Graph zeigt, wie viele Puffer für externe Befehle über die Zeit gesehen benutzt wurden. Wenn die Zahl von benutzten Puffern regelmäßig fast die Zahl von verfügbaren Puffern erreicht, dann ist es wahrscheinlich, dass Sie die Anzahl von verfügbaren Puffern mit Hilfe der Direktive [external command buffer slots](#) erhöhen sollten. Jeder Puffer kann genau einen externen Befehl aufnehmen. Puffer werden für die vorübergehende Aufbewahrung von externen Befehlen genutzt, und zwar vom Lesen aus dem [external command file](#) bis zur Verarbeitung durch den IcingaDaemon.

Wie Sie sehen wird nur ein Puffer genutzt und das ist genau der für die Ergebnisse des check_nagiosstats-Plugins.

Zwischengespeicherte Host- und Service-Prüfungen (cached host and service checks)

Abbildung 8.7. Zwischengespeicherte Host- und Service-Prüfungen

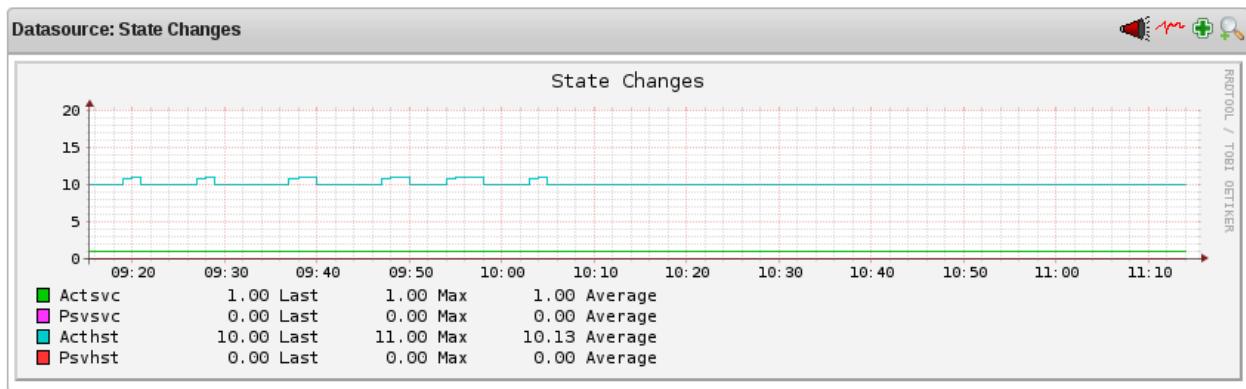


Dieser Graph zeigt, wie viele zwischengespeicherte Host- und Service-Prüfungen über die Zeit aufgetreten sind. Das ist nützlich zum Verständnis von:

- [Zwischengespeicherten Prüfungen \(cached checks\)](#)
- [Vorausschauenden Host- und Service-Abhängigkeitsprüfungen \(predictive host and service dependency checks\)](#)

Durchschnittliche Zustandswechsel

Abbildung 8.8. Durchschnittliche Zustandswechsel



Dieser Graph zeigt den durchschnittlichen prozentualen Zustandswechsel (ein Maß für die Sprunghäufigkeit) über die Zeit gesehen, unterschieden nach Hosts und Service, die zuletzt aktiv oder passiv geprüft wurden. Das ist nützlich zum Verständnis von:

- Flattererkennung (flap detection)
-

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Nutzung des Icingastats-Utilitys

[Zum Anfang](#)

Temporäre Daten

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Temporäre Daten

[Zurück](#)
[Kapitel 8. Sicherheit und Leistungsoptimierung](#)
[Weiter](#)

Temporäre Daten

Verschiedene Dateien werden beim Start von Icinga angelegt und während der Laufzeit sehr oft verarbeitet. Abhängig von der Größe Ihrer Konfiguration kann dies zu hoher I/O-Last und damit zu einer eingeschränkten Bedienbarkeit führen. Um physikalische I/O-Operationen zu reduzieren kann es sinnvoll sein, temporäre Daten auf einer RAM-Disk abzulegen. Die folgenden Zeilen zeigen die Schritte zum Anlegen einer RAM-Disk und die Änderungen an der Hauptkonfigurationsdatei.



Anmerkung

Bitte denken Sie daran, dass die Dateien verloren sind, wenn Sie das System neu starten. Beachten Sie auch, dass es teilweise schwierig ist, die Größe der Dateien zu ermitteln, was ggf. zu einer vollen RAM-Disk führen kann.

1. Werfen Sie einen Blick auf den aktuellen Standort des `status file` (z.B. `/usr/local/icinga/var/status.dat`) und des `object cache file` (z.B. `/usr/local/icnng/var/objects.cache`) und ermitteln Sie die Größe der beiden Dateien.

```
#> ls -la /usr/local/icinga/var/
-rw-rw-r-- 1 icinga icinga 8.2M Jun 10 11:57 status.dat
-rw-r--r-- 1 icinga icinga 5.9M Jun 10 11:58 objects.cache
```

2. Erhöhen Sie die Zahl um einen nennenswerten Betrag, um auch zukünftiges Wachstum zu ermöglichen (100 MB sollten in diesem Fall ausreichen) und legen Sie die RAM-Disk an.



Achtung

Wenn der Wert zu groß gewählt ist, wird dies Ihr System drosseln, weil es dann anfängt zu "swappen", was wieder zu physikalischem I/O führt.

```
#> mkdir /var/icinga/ramdisk
#> mount -t tmpfs tmpfs /var/icinga/ramdisk -o size=100m
#> chown icinga:icinga /var/icinga/ramdisk
```

Passen Sie die Angaben von Benutzer und Gruppe auf die Werte an, die in Ihrer Konfiguration benutzt werden (falls notwendig).

3. Fügen Sie einen Eintrag zur Datei `/etc/fstab` hinzu, um die Änderungen permanent zu machen, damit die RAM-Disk beim nächsten Systemneustart automatisch angelegt wird.

```
tmpfs          /var/icinga/ramdisk    tmpfs    size=100m      0 0
```

4. Editieren Sie die Icinga Hauptkonfigurationsdatei und ändern Sie die Einstellungen der betreffenden Direktiven

```
#object_cache_file=/usr/local/icinga/var/objects.cache
object_cache_file=/var/icinga/ramdisk/objects.cache
```

```
#status_file=/usr/local/icinga/var/status.dat
status_file=/var/icinga/ramdisk/status.dat
```

5. Starten Sie Icinga neu, damit die Änderungen aktiv werden

```
#> /etc/init.d/icinga restart
```

Möglicherweise möchten Sie die RAM-Disk auch für anderen Dateien wie z.B. die check results benutzen. Bitte erhöhen Sie die Größe der RAM-Disk - falls notwendig - und ändern Sie die Direktive in der Hauptkonfigurationsdatei

```
check_result_path=/var/icinga/ramdisk/checkresults
```

Stellen Sie sicher, dass auch anderen Addons wie z.B. check_mk diese Änderungen bekannt sind.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

grafische Darstellung von
Performance-Informationen mit
PNP4Nagios

[Zum Anfang](#)

Kapitel 9. Integration mit anderer
Software

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Kapitel 9. Integration mit anderer Software

[Zurück](#)

[Weiter](#)

Kapitel 9. Integration mit anderer Software

Inhaltsverzeichnis

- [Integrationsüberblick](#)
 - [SNMP-Trap-Integration](#)
 - [TCP-Wrapper-Integration](#)
 - [MKLiveStatus-Integration](#)
 - [Installation von Icinga-Reporting mit JasperServer](#)
-

[Zurück](#)

[Weiter](#)

[Temporäre Daten](#)

[Zum Anfang](#)

[Integrationsüberblick](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Integrationsüberblick

[Zurück](#)

Kapitel 9. Integration mit anderer Software

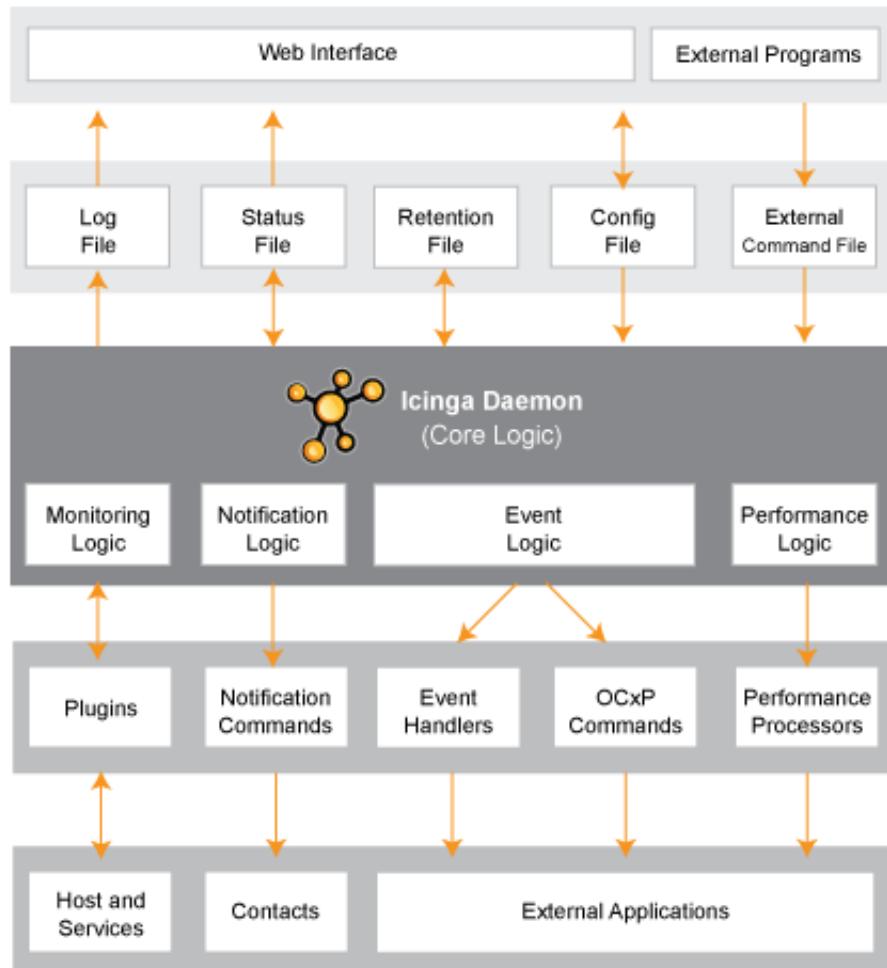
[Weiter](#)

Integrationsüberblick

Einführung

Einer der Gründe, warum Icinga solch eine populäre Überwachungsapplikation ist, liegt in der Tatsache, dass es einfach in Ihre vorhandene Infrastruktur integriert werden kann. Es gibt mehrere Methoden, um Icinga mit der Management-Software zu integrieren, die Sie bereits nutzen und Sie können fast jede Art von neuer oder angepasster Hardware, Service oder Applikation überwachen, die Sie haben.

Integrationsstellen



Um neue Hardware, Services oder Applikationen zu überwachen, prüfen Sie die Dokumentationen zu:

- [Plugins](#)
- [Plugin API](#)
- [Passive Prüfungen](#)
- [Eventhandler](#)

Um Daten aus externen Applikationen in Icinga zu bekommen, prüfen Sie die Dokumentationen zu:

- [Passive Prüfungen](#)
- [Externe Befehle](#)

Um Zustands-, Leistungs- oder Benachrichtigungsinformationen von Icinga an externe Applikationen zu senden, prüfen Sie die Dokumentationen zu:

- [Eventhandlers](#)
- [OCSP - und OCHP-Befehlen](#)
- [Performance-Daten](#)
- [Benachrichtigungen](#)

Integrationsbeispiele

Wir haben ein paar Beispiele dokumentiert, wie Icinga mit externen Applikationen integriert wird:

- [TCP-Wrappers](#) (Sicherheitsalarme)
- [SNMP-Traps](#) (Arcserve Backup-Job-Status)
- [mklivestatus](#) (Interface von Icinga zu verschiedenen Addons wie [NagVis](#) und [Thruk](#))

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Kapitel 9. Integration mit anderer
Software

[Zum Anfang](#)

SNMP-Trap-Integration



SNMP-Trap-Integration

[Zurück](#)**Kapitel 9. Integration mit anderer Software**[Weiter](#)

SNMP-Trap-Integration

Einführung

 Hinweis: Icinga ist nicht als Ersatz für eine ausgewachsene SNMP-Management-Applikation wie HP-OpenView oder [OpenNMS](#) gedacht. Allerdings können Sie die Dinge so einrichten, dass SNMP-Traps, die von einem Host in Ihrem Netzwerk empfangen werden, Alarne in Icinga zu generieren.

Als wenn es dazu gemacht wäre, die Götter der Scheinheiligkeit vor Lachen sterben zu lassen, ist SNMP alles andere als einfach. SNMP-Traps zu übersetzen und sie in Icinga zu bekommen (als passive Prüfresultate) kann ein wenig mühselig sein. Um diese Aufgabe zu erleichtern, empfehlen wir, dass Sie sich Alex Burger's SNMP Trap Translator Projekt unter <http://www.snmptt.org> ansehen. Wenn es mit Net-SNMP kombiniert wird, liefert SNMPTT ein fortgeschrittenes Trap-Behandlungssystem, das mit Icinga integriert werden kann.

Yep, das ist alles.

[Zurück](#)[Nach oben](#)[Weiter](#)[Integrationsüberblick](#)[Zum Anfang](#)[TCP-Wrapper-Integration](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>

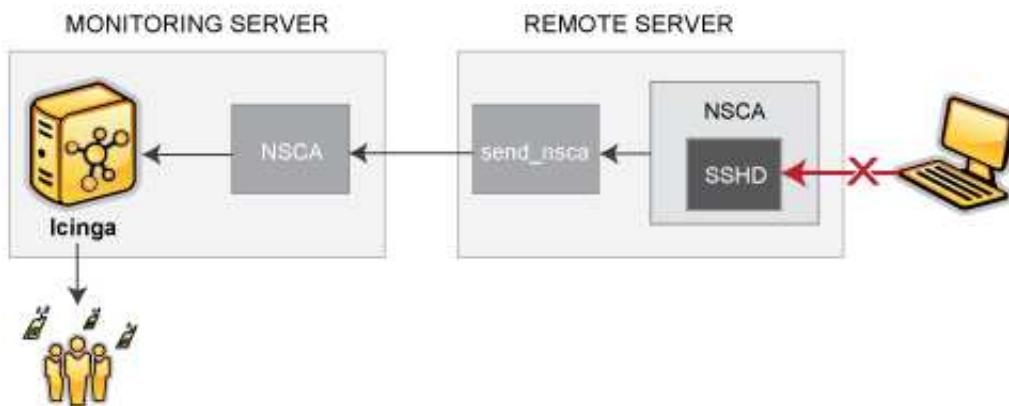


TCP-Wrapper-Integration

[Zurück](#)
[Kapitel 9. Integration mit anderer Software](#)
[Weiter](#)

TCP-Wrapper-Integration

Einführung



Dieses Dokument erklärt, wie einfach in Icinga Alarne für Verbindungsversuche generiert werden können, die von TCP-Wrappern zurückgewiesen werden. Wenn zum Beispiel ein unautorisierte Host versucht, sich mit Ihrem SSH-Server zu verbinden, können Sie in Icinga einen Alarm empfangen, der den Namen des Hosts enthält, der zurückgewiesen wurde. Wenn Sie das auf Ihren Linux/Unix-Boxen installieren, dann werden Sie erstaunt sein, wie viele Port-Scans Sie in Ihrem Netzwerk entdecken.

Diese Anweisungen gehen davon aus, dass

1. Sie bereits mit [passiven Prüfungen](#) vertraut sind und wissen, wie sie arbeiten.
2. Sie bereits mit [sprunghaften Services](#) vertraut sind und wissen, wie sie arbeiten.
3. der Host, für den Sie Alarne generieren (d.h. der Host, auf dem Sie TCP-Wrapper benutzen), ein entfernter Host ist (in diesem Beispiel *firestorm* genannt). Wenn Sie Alarne auf dem gleichen Host generieren möchten, müssen Sie ein paar Anpassungen an den Beispielen machen, die wir bereitstellen.
4. Sie den [NSCA-Daemon](#) auf Ihrem Überwachungs-Server und den NSCA client (*send_nsca*) auf der entfernten Maschine installiert haben, für die Sie TCP-Wrapper-Alarne generieren möchten.

Einen Service definieren

Wenn Sie es nicht bereits getan haben, erstellen Sie eine [Host-Definition](#) für den entfernten Host (*firestorm*).

Als nächstes definieren Sie einen Service in einer Ihrer [Objektkonfigurationsdateien](#) für die TCP-Wrapper-Alarme auf dem Host *firestorm*. Die Service-Definition könnte wie folgt aussehen:

```
define service{
    host_name          firestorm
    service_description TCP Wrappers
    is_volatile        1
    active_checks_enabled 0
    passive_checks_enabled 1
    max_check_attempts 1
    check_command      check_none
    ...
}
```

Es gibt einige wichtige Dinge zu der obigen Service-Definition anzumerken:

1. Die *volatile*-Option ist aktiviert. Wir wollen, dass diese Option aktiviert ist, weil wir eine Benachrichtigung für jeden Alarm haben wollen, der herein kommt.
2. Aktive Prüfungen für den Service sind deaktiviert, während passive Prüfungen aktiviert sind. Das bedeutet, dass der Service niemals aktiv von Icinga geprüft wird - alle Alarminformationen müssen passiv von einer externen Quelle empfangen werden.
3. Der *max_check_attempts*-Wert wird auf 1 gesetzt. Das gewährleistet, dass Sie eine Benachrichtigung erhalten, sobald der erste Alarm generiert wird.

TCP-Wrapper konfigurieren

Nun müssen Sie die */etc/hosts.deny*-Datei auf *firestorm* editieren. Damit die TCP-Wrapper einen Alarm an den Überwachungs-Host senden, sobald ein Verbindungsversuch verweigert wird, müssen Sie eine Zeile hinzufügen, die der folgenden ähnlich ist.

```
ALL: ALL: RFC931: twist (/usr/local/icinga/libexec/eventhandlers/handle_tcp_wrapper %h %d) &
```

Diese Zeile nimmt an, dass es ein Script namens *handle_tcp_wrapper* im */usr/local/icinga/libexec/eventhandlers/-Verzeichnis* auf *firestorm* gibt. Wir werden dieses Script als nächstes schreiben.

Das Script schreiben

Als letztes müssen Sie das *handle_tcp_wrapper*-Script auf *firestorm* schreiben, das den Alarm zurück an den Icinga-Server schickt. Es könnte ungefähr so aussehen:

```
#!/bin/sh
/usr/local/icinga/libexec/eventhandlers/submit_check_result firestorm "TCP Wrappers" 2 "Denied $2-$1" > /dev/null 2> /dev/null
```

Beachten Sie, dass das *handle_tcp_wrapper*-Script das *submit_check_result*-Script aufruft, um den Alarm zurück an den Überwachungs-Host zu schicken. Angenommen, Ihr Icinga-Server heißt *monitor*, dann könnte das *submit check result*-Script wie folgt aussehen:

```
#!/bin/sh
# Arguments
#   $1 = name of host in service definition
#   $2 = name/description of service in service definition
#   $3 = return code
#   $4 = output
/bin/echo -e "$1\t$2\t$3\t$4\n" | /usr/local/icinga/bin/send_nsca monitor -c /usr/local/icinga/etc/send_nsca.cfg
```

Aufräumen

Sie haben nun alles konfiguriert, was Sie brauchen, so dass Sie nur noch den *inetd*-Prozess auf *firestorm* und Icinga auf Ihrem Überwachungs-Server neu starten müssen. Das war's! Wenn die TCP-Wrapper auf *firestorm* einen Verbindungsversuch verweigern, dann sollten Sie Alarme in Icinga erhalten. Die Plugin-Ausgabe für den Alarm könnte wie folgt aussehen:

```
Denied sshd2-sdn-ar-002mmminnP321.dialsprint.net
```

[Zurück](#)[Nach oben](#)[Weiter](#)[SNMP-Trap-Integration](#)[Zum Anfang](#)[MKLiveStatus-Integration](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



MKLiveStatus-Integration

[Zurück](#)
[Kapitel 9. Integration mit anderer Software](#)
[Weiter](#)

MKLiveStatus-Integration

Einführung

MKLiveStatus ist ein Modul von Mathias Kettner zur Anbindung von verschiedenen Addons wie z.B. NagVis oder Thruk an Icinga (oder Nagios). Solange Sie keine Datenbank zur Speicherung von historischen Werten benötigen, könnte dies eine gute Wahl sein, weil es relativ klein und einfach zu installieren ist. Auf der [offiziellen Website](#) finden Sie die komplette Dokumentation, denn hier beschreiben wir nur in sehr kurzer Form die Installation und Konfiguration von MKLiveStatus für Icinga. Wir nehmen an, dass Sie Icinga in `/usr/local/icinga` installiert haben.

1. Laden Sie die Software und kompilieren Sie das Modul (bitte schauen Sie auf der Website nach der aktuellsten Version)

```
wget http://mathias-kettner.de/download/mk-livestatus-1.1.12p3.tar.gz
tar xzvf mk-livestatus-1.1.12p3.tar.gz
cd mk-livestatus-1.1.12p3
./configure --prefix=/usr/local/icinga --exec-prefix=/usr/local/icinga
make
cp src/livestatus.o /usr/local/icinga/bin
```

2. Editieren Sie `icinga.cfg`, um das Modul zu integrieren. Stellen Sie sicher, dass das Verzeichnis `/usr/local/icinga/var/rw` existiert und dass der Icinga-Benutzer dort Schreibrechte hat. Es sollte das gleiche Verzeichnis sein, das auch für das Command File (meistens `icinga.cmd`) benutzt wird. "live" ist ein Socket, der nur während der Laufzeit des Moduls vorhanden ist.

```
broker_module=/usr/local/icinga/bin/livestatus.o
/usr/local/icinga/var/rw/live
```



Anmerkung

Ab Icinga 1.4 können Sie statt des `broker_module`-Eintrags die neue `module`-Definition in einer Ihrer Objektkonfigurationsdateien benutzen:

```
define module{
    module_name      mklivestatus
    path            /usr/local/icinga/bin/livestatus.o
    module_type     neb
    args           /usr/local/icinga/var/rw/live
}
```

3. Restarten Sie Icinga

```
service icinga restart
```

oder

```
/etc/init.d/icinga restart
```

4. Prüfen Sie, ob das Modul läuft

```
ps -ef | grep livestatus  
ls -la /usr/local/icinga/var/rw/live
```

Falls es keinen Prozess und/oder keinen Socket gibt, dann prüfen Sie das Icinga-Log und bereinigen Sie vorhandene Fehler.

[Zurück](#)[Nach oben](#)[Weiter](#)[TCP-Wrapper-Integration](#)[Zum Anfang](#)[Installation von Icinga-Reporting mit JasperServer](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Installation von Icinga-Reporting mit JasperServer

[Zurück](#)

[Kapitel 9. Integration mit anderer Software](#)

[Weiter](#)

Installation von Icinga-Reporting mit JasperServer

Icinga-Reporting basiert auf dem IDOUtils-Backend und unterstützt, genau wie der Icinga-Core, alle großen Datenbankplattformen.

Hier geben wir Ihnen einige Hinweise, wie Sie das Icinga-Reporting mit JasperServer installieren.

Offizielle Dokumentation

Mehr Information finden Sie unter

<http://sourceforge.net/projects/jasperserver/files/JasperServer/JasperServer%204.2.1/JasperReports-Server-CP-Install-Guide.pdf>
oder

http://jasperforge.org/espdocs/espbrowse.php?id=86&type=folder&group_id=112&parent=86

Voraussetzungen

- Apache Tomcat

Nur falls Sie nicht den mit JasperServer gebündelten Tomcat benutzen möchten.

```
#> apt-get install tomcat6
```

- PostgreSQL

Nur falls Sie nicht das mit JasperServer gebündelte PostgreSQL benutzen möchten.
PostgreSQL ist für JasperServers interne Speicherung erforderlich.



Anmerkung

JasperServer haben ihr Backend in Version 4.2.1 von MySQL nach PostgreSQL geändert! [Source](#)

- JasperServer

Abhängig von Ihrer Architektur (`uname -a`) holen Sie sich den entsprechenden Installer - hier im Beispiel x64

```
$> wget http://sourceforge.net/projects/jasperserver/files/JasperServer/JasperServer%204.2.1/jasperreports-server-cp-4.2.1-linux-x64-installer.run
$> chmod +x jasperreports-server-cp-4.2.1-linux-x64-installer.run
$> sudo ./jasperreports-server-cp-4.2.1-linux-x64-installer.run
```

- stimmen Sie der Lizenz zu
- wählen Sie "bundled tomcat"
- wählen Sie "bundled postgresql"
 - wählen Sie eine anderen PostgreSQL-Port, wie z.B. 5443, um nicht mit existierenden Einstellungen zu kollidieren
- installieren Sie die Beispiel-Reports und "ireports"

Der Installer wird automatisch die benötigte Datenbank und das Schema anlegen und die Beispiele installieren.

Start

```
#> cd /opt/jasperreports-server-cp-4.2.1/
#> ./ctlscript.sh start
```

Für einzelne Komponenten

```
#> ./ctlscript.sh postgresql start|stop
#> ./ctlscript.sh tomcat start|stop
```

Icinga Reporting

Beginnend mit Icinga 1.6 liefert das Paket configure- und make-Skripte.

- Sourcen

Laden Sie das Paket von sourceforge herunter und entpacken es. Falls Sie ein Problem haben, das ggf. in der Entwickerversion gelöst ist, dann holen Sie sich den aktuellen Entwickler-Snapshot aus git

```
$> wget http://sourceforge.net/projects/icinga/files/icinga-reporting/1.6.0/icinga-reports-1.6.0.tar.gz; \
tar xzf icinga-reports-1.6.0.tar.gz; \
cd icinga-reports-1.6.0
```

oder

```
$> git clone git://git.icinga.org/icinga-reports.git ; \
cd icinga-reports
```

Falls ein Entwickler Sie gebeten hat, ihren/seinen Branch zu benutzen, dann benutzen Sie

```
$> git branch localbranch origin/devhead/devbranch
$> git checkout localbranch
$> git log
```

- Konfigurieren

Falls Sie den JasperServer vorher nicht mit dem Default-Präfix installiert haben, dann müssen Sie configue den Standort mitteilen.

```
$> ./configure --with-jasper-server=/opt/jasperreports-server-cp-4.2.1
```

- Make

Rufen Sie 'make' ohne Parameter auf, um eine komplette Liste der verfügbaren Optionen zu erhalten.

- MySQL Connector

JasperServer wird mit dem PostgreSQL-Connector ausgeliefert, aber wenn Sie MySQL benötigen, dann bietet das Icinga-Reporting-Paket eins zur Installation an.

```
#> make install-mysql-connector
```

Starten Sie den Tomcat-Server neu

```
#> cd /opt/jasperreports-server-cp-4.2.1
#> ./ctlscript.sh stop tomcat
#> ./ctlscript.sh start tomcat
```

- Installieren

```
#> make install
```

Starten Sie den Tomcat-Server neu

```
#> cd /opt/jasperreports-server-cp-4.2.1
#> ./ctlscript.sh stop tomcat
#> ./ctlscript.sh start tomcat
```

Konfiguration

Öffnen Sie das Web-Interface, Standardwerte sind jasperadmin:jasperadmin

<http://127.0.0.1:8080/jasperserver>

- IDO

Gehen Sie nach /root/Icinga/datasource (stellen Sie sicher, dass Refine den Wert "changed by anyone" enthält).

- editieren Sie die vorhandene "datasource" und stellen Sie Ihre Werte ein.
- testen Sie die Konfiguration und speichern Sie die Verbindung.
- alle Reports in unserem Paket zeigen auf diese "datasource" und sollten lauffähig sein.

Reports

Überprüfen Sie, dass die Reports verfügbar und lauffähig sind.

Abbildung 9.1. Icinga reporting

Fehler

- Anzeigefehler

```
org.springframework.web.util.NestedServletException: Handler processing failed;
nested exception is java.lang.InternalError:
Can't connect to X11 window server using ':0' as the value of the DISPLAY variable.
```

Der Tomcat läuft als root und versucht, auf das X11-Display des Benutzers zuzugreifen. Erlauben Sie es für jeden lokalen Benutzer.

```
$> xhost +local:all
```

- Generierungsfehler

```
org.springframework.web.util.NestedServletException: Handler processing failed;
nested exception is java.lang.NoClassDefFoundError:
Could not initialize class net.sf.jasperreports.engine.util.JRStyledTextParser
```

Es könnte Sun AWT sein, das versucht, auf X11-Klassen zuzugreifen, aber tatsächlich ist es die PDF-Generierung für einen Report.

```
$> sudo apt-get install msttcorefonts
$> cd /opt/jasperreports-server-cp-4.2.1
$> sudo sh ctlscript.sh stop
$> sudo sh ctlscript.sh start
```

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[MKLiveStatus-Integration](#)

[Zum Anfang](#)

[Kapitel 10. weitere Software](#)



Kapitel 10. weitere Software

[Zurück](#)

[Weiter](#)

Kapitel 10. weitere Software

Inhaltsverzeichnis

[Icinga Addons](#)

[NRPE](#)

[NSCA](#)

[Zurück](#)

[Weiter](#)

Installation von Icinga-Reporting
mit JasperServer

[Zum Anfang](#)

[Icinga Addons](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Icinga Addons

[Zurück](#)[Kapitel 10. weitere Software](#)[Weiter](#)

Icinga Addons

Einführung

Es gibt eine Menge von "Addon"-Software-Paketen, die für Icinga verfügbar sind. Addons können genutzt werden, um die Funktionalität von Icinga zu erweitern oder Icinga mit anderen Applikationen zu integrieren.

Addons gibt es für:

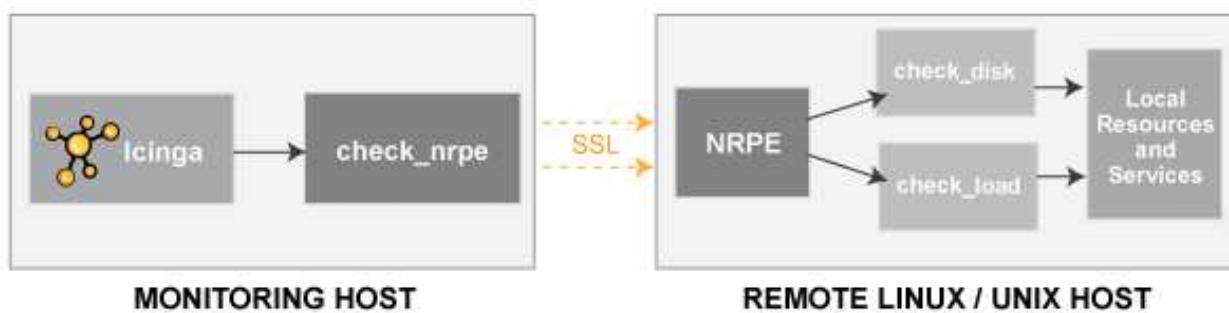
- die Verwaltung der Konfigurationsdateien über ein Web-Interface
 - [NConf](#), [NagiosQL](#), [LConf](#), [Lilac](#), ...
- die Überwachung von entfernten Hosts (*NIX, Windows, etc.)
 - [NSCA](#), [NRPE](#), [check_mk](#), ...
 - [NSClient++](#), ...
- die Erteilung von passiven Prüfungen von entfernten Hosts
 - [NSClient++](#), [check_mk](#), ...
- die Vereinfachung/Erweiterung der Überwachungslogik
 - [Business Process Addon](#) ...
- Visualisierung der Informationen
 - [PNP4Nagios](#)
 - [NagVis](#)
- alternative Web-Interfaces
 - [Thruk](#), [MultiSite](#)
- ... und vieles mehr

Sie finden viele Addons für Icinga unter:

- <http://www.icinga.org/>
- SourceForge.net
- <http://www.monitoringexchange.org>

Wir werden eine kurze Einführung für ein paar Addons geben, die Ethan Galstad für Nagios entwickelt hat...

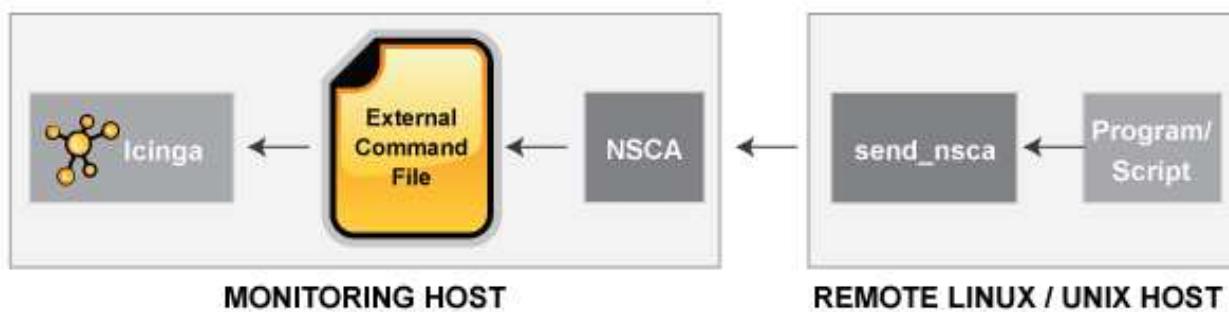
NRPE



NRPE ist ein Addon, das es Ihnen erlaubt, [Plugins](#) auf entfernten Linux-/Unix-Hosts auszuführen. Dies ist nützlich, wenn Sie lokale Ressourcen/Attribute wie Plattenbelegung, CPU-Last, Speicherbelegung usw. auf entfernten Hosts überwachen wollen. Ähnliche Funktionalitäten können durch das *check_by_ssh*-Plugin erreicht werden, obwohl es auf dem Überwachungsrechner für eine höhere CPU-Belastung sorgen kann - besonders dann, wenn Sie hunderte oder tausende von Hosts überwachen.

Das NRPE-Addon finden Sie unter <https://git.icinga.org/?p=icinga-nrpe.git>. Die [Dokumentation](#) finden Sie in nächsten Abschnitt.

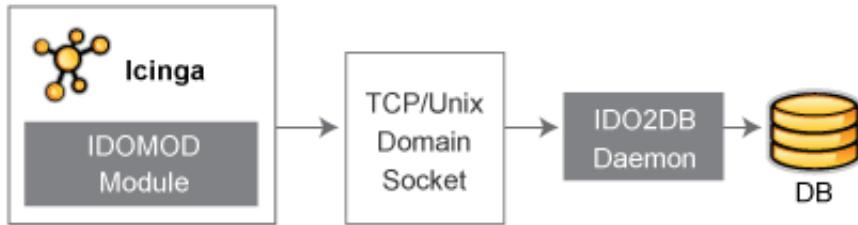
NSCA



NSCA ist ein Addon, das es Ihnen erlaubt, [passive Prüf](#)-Resultate von entfernten Linux-/Unix-Hosts an den Icinga-Daemon zu senden, der auf dem Überwachungs-Server läuft. Das ist sehr hilfreich in [verteilten](#) und [redundanten/Failover](#)-Überwachungs-Umgebungen.

Das NSCA-Addon finden Sie unter <https://git.icinga.org/?p=icinga-nsca.git>. Die [Dokumentation](#) finden Sie in nächsten Abschnitt.

IDOUtils



IDOUtils ist ein Addon, das es Ihnen erlaubt, alle Icinga-Statusinformationen in einer Datenbank zu speichern. Mehrere Instanzen von Icinga können all ihre Informationen in einer gemeinsamen Datenbank für ein zentrales Berichtswesen speichern. Dies wird wahrscheinlich in der Zukunft als Basis für ein neues PHP-basiertes Web-Interface für Icinga dienen. Neben MySQL werden auch Oracle und PostgreSQL unterstützt.

Das IDOUtils-Addon und die Dokumentation finden Sie unter <http://docs.icinga.org/>.

[Zurück](#)[Nach oben](#)[Weiter](#)[Kapitel 10. weitere Software](#)[Zum Anfang](#)[NRPE](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



NRPE

[Zurück](#)

Kapitel 10. weitere Software

[Weiter](#)

NRPE

Einführung

Nagios Remote Plugin Executor (oder kurz NRPE) ist ein Addon, das benutzt wird, um Plugins auszuführen, die "lokale" Ressourcen auf entfernten (Linux/Unix) Systemen überwachen. Einige Ressourcen können (oder sollen) nicht per SNMP oder andere Agenten über das Netzwerk überwacht werden, so dass Sie diese Prüfungen mit Programmen durchführen müssen, die lokal auf den zu überwachenden Maschinen installiert sind und die Ergebnisse an den Icinga-Server zurückliefern. Im Gegensatz zu NSCA geschieht dies aktiv, d.h. durch den Icinga-Server initiiert.

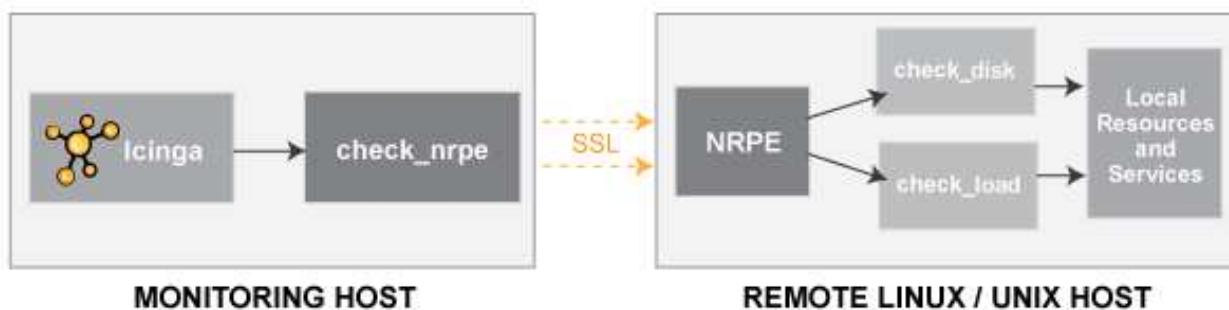


Anmerkung

Mit Hilfe von [NSClient++](#) anstatt NRPE auf dem entfernten Host können Sie auch Prüfungen auf Windows-Maschinen ausführen.

Sie können *check_by_ssh* benutzen, um Plugins auf entfernten Maschinen auszuführen, aber es gibt einen Nachteil bei diesen Ansatz. Der Aufbau einer SSH-Session erfordert CPU-Ressourcen sowohl auf dem Überwachungsrechner als auch auf dem entfernten Host, was zu einer Performance-Beeinträchtigung führen kann, wenn Sie auf diese Weise eine Vielzahl von Hosts und/oder Services überwachen. Die Benutzung von NRPE ist ein wenig unsicherer als SSH, aber in vielen Fällen mag der Performance-Gewinn die Abstriche bei der Sicherheit überwiegen. SSL kann übrigens aktiviert werden, wenn Sie eine sicherere Verbindung benötigen.

Abbildung 10.1. NRPE



check_nrpe ist ein Plugin, das auf dem lokalen Icinga-Server genau wie jedes andere Plugin ausgeführt wird. Es verbindet sich mit dem NRPE-Prozess, der als Daemon auf der entfernten Maschine läuft. Der Daemon selbst führt das Plugin auf der gleichen Maschine aus und

überträgt die gesammelten Informationen an das check_nrpe-Plugin, das sie wiederum an Icinga weitergibt.

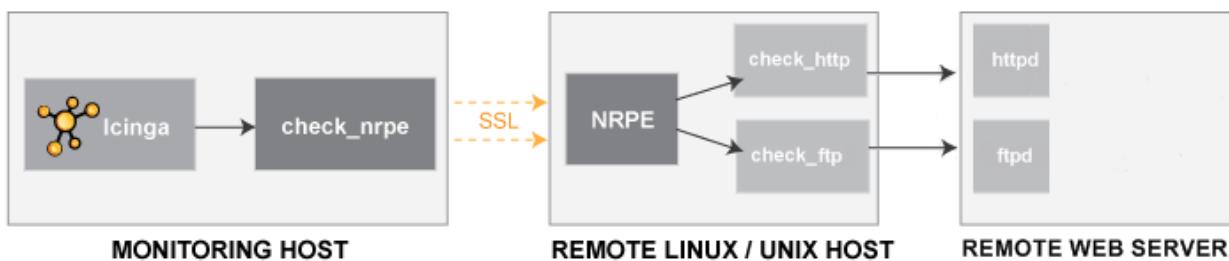


Anmerkung

Abhängig von der CPU / dem OS auf der entfernten Maschine müssen Sie ggf. NRPE und die Plugins auf verschiedenen Plattformen kompilieren.

Mit Hilfe von NRPE werden Sie vorwiegend Ressourcen überwachen, die lokal auf der gleichen Maschine sind, wie z.B. die CPU-Auslastung, Speicherbrauch, Plattenplatz, Prozesse, etc., aber es kann auch genutzt werden, um Ressourcen zu kontrollieren, die nicht direkt vom Überwachungsrechner erreichbar sind. Die Maschine mit dem NRPE-Daemon dient in diesem Fall als eine Art Relais.

Abbildung 10.2. NRPE remote



Die folgenden Anweisungen basieren teilweise auf der Dokumentation, die im ursprünglichen NRPE-Paket von Ethan Galstad zu finden ist.

Voraussetzungen

- Icinga sollte auf dem Überwachungsserver installiert und lauffähig sein
- ein C-Compiler (wie z.B. gcc) ist auf dem lokalen Host installiert. Wenn nicht:

```
#> yum install gcc          # RHEL / Fedora / CentOS
#> apt-get install gcc      # Debian / Ubuntu
#> zypper install gcc       # SLES / openSuSE (oder benutzen Sie YaST)
```

- openssl ist (optional) auf dem lokalen Host installiert. Wenn nicht:

```
#> yum install openssl openssl-devel  # RHEL / Fedora / CentOS
#> apt-get install openssl openssl-devel # Debian / Ubuntu
#> zypper install openssl openssl-devel # SLES / openSuSE (oder benutzen Sie YaST)
```

Download



Anmerkung

Anstatt NRPE von Grund auf zu installieren möchten Sie vielleicht ein Paket benutzen, das es möglicherweise für Ihr Betriebssystem gibt. Bitte werfen Sie einen Blick auf die [Tabelle der Pakete](#).

Falls Sie aus den Sourcen installieren möchten, dann benutzen Sie bitte die offiziellen Release-Tarballs.



Wichtig

Bitte benutzen Sie keine GIT-Snapshots, solange Sie kein Problem haben, das in der aktuellen Entwicklerversion ggf. gelöst ist.

Optionale Anpassungen

Die maximale Länge von Daten, die übertragen werden können, ist auf 2.048 Bytes begrenzt, die maximale Länge von Plugin-Ausgaben auf 512 Bytes. Wenn das für Ihre Zwecke nicht ausreicht, dann müssen Sie die entsprechenden Werte in der Datei `icinga-nrpe/include/common.h` anpassen (und Icinga neu kompilieren!).

```
#define MAX_INPUT_BUFFER      2048     /* max size of most buffers we use */
#define MAX_PLUGINOUTPUT_LENGTH 512
```

Bitte denken Sie daran, dass Sie die Programme erneut kompilieren müssen, wenn Sie diese Werte zu einem späteren Zeitpunkt ändern.

Bedingt durch die Einstellung des folgenden `define` in `include/common.h` (im Icinga-Core) kann der maximale Wert 8.192 Bytes nicht überschreiten.

```
#define MAX_EXTERNAL_COMMAND_LENGTH    8192    /* max length of an external command */
```

Kompilieren auf dem Icinga-Server

Wechseln Sie in das neu angelegte Verzeichnis und führen Sie "configure" und "make" aus

```
#> cd icinga-nrpe
#> ./configure
#> make all
#> make install-plugin
```



Anmerkung

Wenn Sie später SSL benutzen wollen, dann müssen Sie stattdessen ". /configure --enable-ssl" angeben. Außerdem gibt es weitere Optionen, um den Standort von SSL-Dateien anzugeben, falls sie nicht automatisch gefunden werden.

Wenn Benutzer oder Gruppe für den Daemon-Prozess von "icinga" abweichen oder der zu benutzende Port nicht der Default 5666 ist, dann können Sie verschiedene Optionen benutzen, um abweichende Werte anzugeben (--with-nrpe-user=<user>, --with-nrpe-group=<group>, --with-nrpe-port=<port>). Benutzen Sie ". /configure -h", um eine vollständige Liste der Optionen zu erhalten. "make install-plugin" kopiert `check_nrpe` in das Plugin-Verzeichnis.



Anmerkung

Mit Hilfe von "ldd src/check_nrpe" und "ldd src/nrpe" sollten Sie feststellen können, ob SSL-Bibliotheken in den erzeugten Programmen enthalten sind.

Erster Test

Starten Sie den Daemon-Prozess und rufen Sie das Plugin auf

```
#> /usr/src/icinga-nrpe/src/nrpe -n \
  -c /usr/src/icinga-nrpe/sample-config/nrpe.cfg -d
#> /usr/local/icinga/libexec/check_nrpe -H 127.0.0.1 -n
```

Dies sollte die Versionsnummer von NRPE zurückliefern. Wenn Sie die Meldung "CHECK_NRPE: Error receiving data from daemon" erhalten, wurde der angegebene Host nicht in der Datei nrpe.cfg gefunden (Direktive allowed_hosts). Mehrere IP-Adressen werden durch Komma getrennt.

Stoppen Sie den Daemon-Prozess

```
#> kill `ps -ef | grep "sample-config/nrpe.cfg" | grep -v grep | awk '{print $2}'`
```

Entfernte System/e

Die Konfiguration und Installation auf dem Icinga-Server ist vorerst abgeschlossen. Der zweite Teil erfolgt auf dem entfernten System, auf dem der NRPE-Daemon auf ankommende Anfragen lauscht, sie ausführt und die Ergebnisse an den Icinga-Server zurückliefert.

Voraussetzungen auf dem entfernten System

- Stellen Sie sicher, dass die benötigten Plugins auf dem entfernten System verfügbar sind. Lesen Sie ggf. in der [Schnellstartanleitung](#), wie die Plugins zu installieren sind.
- Sie können das Verzeichnis icinga-nrpe samt Unterverzeichnisses vom Icinga-Server kopieren. Eine Möglichkeit ist

```
#> cd /usr/src/
#> scp -pr <Icinga-server>:/$PWD/icinga-nrpe .
```



Anmerkung

Wenn die Architektur auf Ihrem entfernten System von Ihrem Icinga-Server abweicht, dann müssen Sie die Sourcen erneut kompilieren. Dies trifft zu, wenn Sie verschiedene CPUs (i386/Itanium/PA-RISC/...) und/oder unterschiedliche OS-Versionen (32-Bit/64-Bit) einsetzen. Wenn dies der Fall ist, dann müssen Sie (wie oben beschrieben) einen C-Compiler und OpenSSL (wenn Sie SSL benutzen wollen) installieren, bevor Sie fortfahren können.

```
#> cd icinga-nrpe
#> make distclean
#> ./configure      # bitte benutzen Sie die gleichen Optionen wie auf dem Icinga-Server
#> make all
```

Editieren Sie die Konfigurationsdatei sample-config/nrpe.cfg und ändern Sie die Einstellung von "allowed_hosts=<IP address>" auf die IP-Adresse Ihres Icinga-Servers. Mehrere IP-Adressen werden durch Komma getrennt.

Zweiter Test

Starten Sie den Daemon-Prozess auf dem entfernten Host

```
#> /usr/src/icinga-nrpe/src/nrpe -n \
-c /usr/src/icinga-nrpe/sample-config/nrpe.cfg -d
```

und führen Sie das Plugin auf dem Icinga-Server erneut aus, dieses Mal mit der IP-Adresse des entfernten Hosts

```
#> /usr/local/icinga/libexec/check_nrpe -H <IP remote host> -n
```

Dies sollte die Versionsnummer von NRPE zurückliefern. Wenn Sie die Meldung "CHECK_NRPE: Error receiving data from daemon" erhalten, wurde der Icinga-Server nicht in der Datei nrpe.cfg (Direktive allowed_hosts) auf dem entfernten Host gefunden.

Stoppen Sie den Daemon-Prozess auf dem entfernten Host

```
#> kill `ps -ef | grep "sample-config/nrpe.cfg" | grep -v grep | awk '{print $2}'`
```

Installation auf dem entfernten Host

Unabhängig vom verwendeten Modus, in dem der NRPE-Prozess auf dem entfernten Host läuft, benötigen Sie eine Konfigurationsdatei, die die auszuführenden Befehle enthält. Mit dem folgenden Befehl wird sie installiert

```
#> make install-daemon-config
```

Es gibt zwei Arten, den NRPE-Prozess zu starten, entweder als eigenständigen Daemon-Prozess oder per xinetd (was empfohlen wird).

- **nrpe-Daemon**

Installieren Sie zuerst den Daemon

```
#> make install-daemon
```

Wenn Sie xinetd benutzen, wird der Daemon automatisch gestartet. Den eigenständigen Prozess müssen Sie manuell starten

```
#> /usr/local/icinga/bin/nrpe -c /usr/local/icinga/etc/nrpe.cfg
```

- **inetd/xinetd**

Wenn der Daemon durch (x)inetd gestartet werden soll, dann müssen Sie /etc/services erweitern, eine weitere Datei ändern/kopieren und (x)inetd erneut starten. Wenn das Paket nicht installiert ist, dann tun Sie bitte folgendes

```
#> yum install xinetd      # RHEL / Fedora / CentOS
#> apt-get install xinetd  # Debian / Ubuntu
#> zypper install xinetd   # SLES / openSuSE (oder benutzen Sie YaST)
```



Anmerkung

Die Einstellung von "server_port" in der Datei nrpe.cfg wird ignoriert, wenn Sie inetd/xinetd benutzen.

```
#> echo "nrpe 5666/tcp # nrpe" >> /etc/services
```

Abhängig vom installierten Superserver auf dem entfernten System gibt es drei Alternativen

- **inetd MIT tcpwrappers**

Fügen Sie Einträge in /etc/hosts.allow und /etc/hosts.deny ein, um TCP-wrapper-protection für den nrpe-Service zu aktivieren. Dies ist optional, wird aber wärmstens empfohlen. Fügen Sie "nrpe stream tcp nowait <user> /usr/sbin/tcpd <nrpe-binary> -c <nrpe-cfg> --inetd" zur Datei /etc/inetd.conf hinzu, z.B.

```
#> echo "nrpe stream tcp nowait icinga /usr/sbin/tcpd /usr/local/icinga/bin/nrpe \
-c /usr/local/icinga/etc/nrpe.cfg --inetd" >> /etc/inetd.conf
#> /etc/init.d/inetd restart
```

- inetd OHNE tcpwrappers

Fügen Sie "nrpe stream tcp nowait <user> <nrpe-binary> -c <nrpe-cfg> --inetd" zur Datei /etc/inetd.conf hinzu, z.B.

```
#> echo "nrpe stream tcp nowait icinga /usr/local/icinga/bin/nrpe \
-c /usr/local/icinga/etc/nrpe.cfg --inetd" >> /etc/inetd.conf
#> /etc/init.d/inetd restart
```

- xinetd (empfohlen)

Editieren Sie die Konfigurationsdatei nrpe.xinetd im Verzeichnis sample-config und ersetzen Sie die Adresse hinter <only_from> durch die IP-Adresse des Icinga-Servers (wo check_nrpe laufen wird). Mehrere IP-Adressen werden durch Leerzeichen voneinander getrennt.

Fügen Sie Einträge in /etc/hosts.allow und /etc/hosts.deny ein, um TCP-wrapper-protection für den nrpe-Service zu aktivieren. Dies ist optional, wird aber wärmstens empfohlen. Kopieren Sie die Datei in das xinetd-Verzeichnis und starten Sie den xinetd-Prozess neu

```
#> make install-xinetd
#> /etc/init.d/xinetd restart
```

Dritter Test

Wechseln Sie auf dem Icinga-Server zum Icinga-Benutzer und führen Sie einen weiteren Test aus

```
#> su - icinga
$> /usr/local/icinga/libexec/check_nrpe -H <IP remote server>
```

Dies sollte ein weiteres Mal die NRPE-Versionsnummer ausgeben. Wenn dieser Test fehlschlägt, dann ist es nicht sinnvoll fortzufahren. Prüfen Sie stattdessen die Einstellungen in nrpe.cfg/nrpe.xinet auf dem entfernten Server. Prüfen Sie außerdem, ob es Meldungen im Syslog (z.B. /var/log/messages) auf dem entfernten Host gibt.

Fehlersuche

Prüfen Sie auf dem entfernten Host, ob der nrpe-Prozess läuft

- wenn als eigenständiger Prozess installiert

```
#> ps -ef | grep -v grep | grep nrpe
```

Falls der Prozess nicht läuft, dann

- starten Sie ihn wie oben angegeben
- prüfen Sie, ob die Konfigurationsdatei /usr/local/icinga/etc/nrpe.cfg vorhanden ist
- prüfen Sie, ob die Direktive *allowed_hosts* in der Datei /usr/local/icinga/etc/nrpe.cfg einen Eintrag für die IP-Adresse des Icinga-Servers enthält. Mehrere IP-Adressen werden durch Komma getrennt

- wenn per xinetd gestartet

```
#> netstat -at | grep -v grep | grep nrpe
```

Die Ausgabe sollte etwa dem Folgenden entsprechen

```
tcp 0 0 *:nrpe *:* LISTEN
```

Wenn das nicht der Fall ist, dann prüfen Sie, ob

- die Datei /etc/services einen Eintrag für nrpe enthält
- die Datei /etc/xinetd.d/nrpe vorhanden ist
- die Direktive *only_from* directive in der Datei /etc/xinetd.d/nrpe einen Eintrag für die IP-Adresse des Icinga-Servers enthält. Mehrere IP-Adressen werden durch Leerzeichen voneinander getrennt
- xinetd installiert und gestartet ist
- die System-Logs Fehler von xinetd und/oder nrpe enthalten. Beheben Sie die gemeldeten Probleme

Aktivieren Sie "debug=1" in nrpe.cfg, starten Sie den Daemon (falls zutreffend) und schauen Sie nach Meldungen in Syslog / nrpe.log.

Sicherheit

Konsultieren Sie die Datei SECURITY, um mehr Informationen zu den Sicherheitsrisiken zu kommen, die beim Betrieb von NRPE auftreten können, zusammen mit einer Erklärung, welche Art von Schutz die Verschlüsselung bietet.

Definition von lokalen Prüfungen

Einige Dinge wurden bereits in etc/nrpe.cfg auf dem entfernten Host vorkonfiguriert

```
# command[<command_name>]=<command_line>
command[check_users]=/usr/local/icinga/libexec/check_users -w 5 -c 10
command[check_load]=/usr/local/icinga/libexec/check_load -w 1.5,1.1,0.9 -c 3.0,2.2,1.9
command[check_hdal]=/usr/local/icinga/libexec/check_disk -w 20% -c 10% -p /dev/hdal
command[check_zombie_procs]=/usr/local/icinga/libexec/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/local/icinga/libexec/check_procs -w 150 -c 200
```

Die erste Zeile zeigt das generelle Format

Zeichenkette	Beschreibung
command	Kennzeichnung, dass das Folgende eine command-Definition ist
<command_name>	Verbindung zwischen der command-Definition auf dem Icinga-Server und dem Befehl auf dem entfernten Host
<command_line>	Aufruf des Plugins inklusive aller notwendigen Argumente

Definitionen auf dem Icinga-Server

Nun wechseln wir auf den Icinga-Server, um einige Objekt-Definitionen anzulegen. Zuerst fügen Sie eine command-Definition zu Ihrer Konfiguration hinzu (falls Sie noch keine passende haben). Wie immer ist der Name der Konfigurationsdatei Ihnen überlassen, aber bei den meisten Leuten heißt sie commands.cfg.

```
define command{
    command_name      check_nrpe
    command_line      $USER1$/check_nrpe -H $HOSTADDRESS$ -c $ARG1$
}
```

Wir nehmen an, dass Sie bereits eine Host-Definition haben, die der folgenden ähnlich ist

```
define host{
    use          generic-host      ; Default-Werte von einer Vorlage erben
    host_name    remotehost        ; Der Name, den wir diesem Host geben
    alias        Linux Host         ; Ein laengerer Name fuer diesen Host
    address      192.168.0.1       ; IP-Adresse des Servers
}
```

Diese Beispieldefinitionen benutzen die oben gezeigten Befehle.

Der folgende Service wird die Anzahl der momentan angemeldeten Benutzer auf dem entfernten Host überwachen

```
define service{
    use          generic-service
    host_name    remotehost
    service_description Current Users
    check_command check_nrpe!check_users
}
```

"check_nrpe" ist die Verbindung zwischen der Service-Direktive "check_command" und der "command_name"-Direktive in der command-Definition auf dem Icinga-Server. Die "command_line" in der command-Definition zeigt, dass "check_nrpe" aufgerufen wird. "check_users" wird als erstes Argument übergeben. Der nrpe-Prozess auf dem entfernten Host nimmt dieses Argument und sucht nach einer passenden Definition in nrpe.cfg. Der Befehl wird ausgeführt und das Ergebnis an das check_nrpe-Plugin auf dem Icinga-Server zurückgeliefert.

Der folgende Service wird die CPU-Auslastung auf dem entfernten Host überwachen

```
define service{
    use          generic-service
    host_name    remotehost
    service_description CPU Load
    check_command check_nrpe!check_load
}
```

Der folgende Service wird den Plattenplatz auf /dev/hda1 auf dem entfernten Host überwachen

```
define service{
    use          generic-service
    host_name    remotehost
    service_description /dev/hda1 Free Space
    check_command check_nrpe!check_hda1
}
```

Der folgende Service wird die Anzahl der Prozesse auf dem entfernten Host überwachen

```
define service{
    use          generic-service
    host_name    remotehost
    service_description Total Processes
    check_command check_nrpe!check_total_procs
}
```

Der folgende Service wird die Anzahl der Zombie-Prozesse auf dem entfernten Host überwachen

```
define service{
    use generic-service
    host_name remotehost
    service_description Zombie Processes
    check_command check_nrpe!check_zombie_procs
}
```

Starten Sie Icinga neu, damit die Änderungen in Ihre laufende Konfiguration übernommen werden

```
#> /etc/init.d/icinga restart
```

Nach einer Weile sollten Ihre Plugins ausgeführt worden sein.

Weitere Fehlersuche

Einige Fehler während der Installation wurden bereits angesprochen. Unglücklicherweise gibt es weitere Fehlermöglichkeiten. Nachfolgend finden Sie Hinweise für einige der häufigsten Fehler mit dem NRPE-Addon.

- "NRPE: Command timed out after x seconds"

Der Befehl, der vom NRPE-Daemon ausgeführt wurde, endete nicht innerhalb der angegebenen Zeit. Sie können den Timeout-Wert für Befehle durch Editieren der NRPE-Konfigurationsdatei erhöhen und die Wert der command_timeout-Variable anpassen. Verwenden Sie die -t -Kommandozeilenoption, um einen höheren Timeout-Wert für das check_nrpe-Plugin anzugeben. Das folgende Beispiel erhöht den Timeout auf 30 Sekunden:

```
/usr/local/icinga/libexec/check_nrpe -H localhost -c somecommand -t 30
```

Wenn Sie den NRPE-Daemon im Standalone-Modus ausführen (und nicht unter inetd oder xinetd), dann müssen Sie ihn neustarten, damit der neue Timeout-Wert erkannt wird.

- "Connection refused or timed out"

Dieser Fehler kann mehrere Ursachen haben:

- Es gibt eine Firewall, die die Kommunikation zwischen dem Überwachungs-Server (auf dem das check_nrpe-Plugin läuft) und dem entfernten Host (auf dem der NRPE-Daemon läuft) blockiert. Stellen Sie sicher, dass die Firewall-Regeln (z.B. iptables) auf dem entfernten Host die Kommunikation erlauben und prüfen Sie, dass sich keine physikalische Firewall zwischen den Überwachungs-Server und dem entfernten Host befindet.
- Wenn Sie den Daemon-Modus benutzen: Die IP-Adresse in nrpe.cfg (allowed_hosts=...) stimmt nicht mit der IP-Adresse des Überwachungs-Servers überein. Falls die Adressen übereinstimmen, dann haben Sie ggf. vergessen, den Daemon nach der letzten Änderung neu zu starten.
- Wenn Sie die (x)inetd-Version benutzen: Die IP-Adresse in /etc/xinetd/nrpe (only_from=...) stimmt nicht mit der IP-Adresse des Überwachungs-Servers überein. Falls die Adressen übereinstimmen, dann haben Sie ggf. vergessen, den xindetd-Prozess nach der letzten Änderung neu zu starten.

- Der NRPE-Daemon ist nicht installiert oder läuft nicht auf dem entfernten Host. Prüfen Sie mit einem der folgenden Befehle, dass der NRPE-Daemon als eigenständiger Prozess läuft bzw. unter inetd/xinetd:

```
ps axuw | grep nrpe      # falls standalone-Daemon
netstat -at | grep nrpe  # falls via xinetd
```

- "CHECK_NRPE: Received 0 bytes from daemon. Check the remote server logs for an error message."

Als erstes sollten Sie die Protokolle des entfernten Hosts auf Fehlermeldungen prüfen. Ehrlich :-). Dieser Fehler kann u.a. folgende Ursachen haben:

- Das check_nrpe-Plugin konnte keinen SSL-Handshake mit dem NRPE-Daemon durchführen. Eine Fehlermeldung in den Log-Dateien sollte zeigen, ob dies der Fall war oder nicht. Prüfen Sie die Versionsstände von OpenSSL auf dem Überwachungs-Server und dem entfernten Host. Wenn Sie eine kommerzielle SSL-Version auf dem entfernten Host betreiben, dann kann es ggf. zu Kompatibilitätsproblemen kommen.
 - "NRPE: Unable to read output"
- Dieser Fehler zeigt an, dass der vom NRPE-Daemon ausgeführte Befehl keinerlei Zeichenausgaben zurückliefert. Dies kann auf folgende Probleme hinweisen:
- Der Pfad des auszuführenden Plugins auf dem entfernten Host ist inkorrekt. Falls Sie die Definition in nrpe.cfg ändern, dann denken Sie daran, den Daemon neu zu starten.
 - Das hinter command_line angegebene Plugin arbeitet fehlerhaft. Führen Sie den Befehl manuell auf der Kommandozeile aus, um sicherzustellen, dass das Plugin Textausgaben liefert. Starten Sie den Befehl NICHT als root!
- "NRPE: Command 'x' not defined"

Der Befehl 'x' ist nicht in der NRPE-Konfigurationsdatei auf dem entfernten Host definiert. Bitte fügen Sie die Befehlsdefinition für 'x' hinzu. Sehen Sie sich die vorhandenen Befehlsdefinitionen in der NRPE-Konfigurationsdatei an, um einen Eindruck davon zu bekommen, wie es aussehen sollte. Wenn Sie den NRPE-Daemon im Standalone-Modus ausführen (und nicht unter inetd oder xinetd), dann müssen Sie ihn neustarten, damit der neue Timeout-Wert erkannt wird.

Falls Sie immer noch Probleme haben, dann setzen Sie "debug=1" in nrpe.cfg auf dem entfernten Host. Denken Sie daran, den NRPE-Prozess neu zu starten, wenn dieser im Standalone-Modus läuft. Führen Sie die Prüfung auf dem Überwachungs-Server aus. Anschließend sollten Sie Debugging-Informationen im Syslog (z.B. /var/log/messages) finden, die bei der Fehlerbehebung weiterhelfen sollten.

Sie können auch an eine der Mailing-Listen bzw. Foren wenden (<http://www.icinga.org/community/get-help/>).

Aktualisierung

- Aktualisierung des Icinga-Servers

Laden Sie die Software herunter

```
#> cd /usr/src
#> wget "https://git.icinga.org/?p=icinga-nrpe.git;a=snapshot;h=HEAD;sf=tgz" -O nrpe.tgz
#> tar xzf nrpe.tgz
```

oder benutzen Sie die aktuelle Version aus dem git-Repository

```
#> git clone git://git.icinga.org/icinga-nrpe.git
```

Dann kompilieren Sie die Software und installieren das Plugin

```
#> cd icinga-nrpe
#> make distclean
#> ./configure      # benutzen Sie die gleichen Optionen wie beim ersten Mal
#> make all
#> make install-plugin
```

- Aktualisierung des entfernten Hosts

Laden Sie die Software herunter

```
#> cd /usr/src
#> wget "https://git.icinga.org/?p=icinga-nrpe.git;a=snapshot;h=HEAD;sf=tgz" -O nrpe.tgz
#> tar xzf nrpe.tgz
```

oder benutzen Sie die aktuelle Version aus dem git-Repository

```
#> git clone git://git.icinga.org/icinga-nrpe.git
```

Dann kompilieren Sie die Software und installieren den Daemon-Prozess

```
#> cd icinga-nrpe
#> make distclean
#> ./configure      # benutzen Sie die gleichen Optionen wie beim ersten Mal
#> make all
### beenden Sie den Standalone-Daemon (falls zutreffend)
#> kill `ps -ef | grep "sample-config/nrpe.cfg" | grep -v grep | awk '{print $2}'``
#> make install-daemon
### starten Sie den Standalone-Daemon (falls zutreffend)
#> /usr/src/icinga-nrpe/src/nrpe -n \
    -c /usr/src/icinga-nrpe/sample-config/nrpe.cfg -d
```

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Icinga Addons](#)

[Zum Anfang](#)

[NSCA](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



NSCA

[Zurück](#)

Kapitel 10. weitere Software

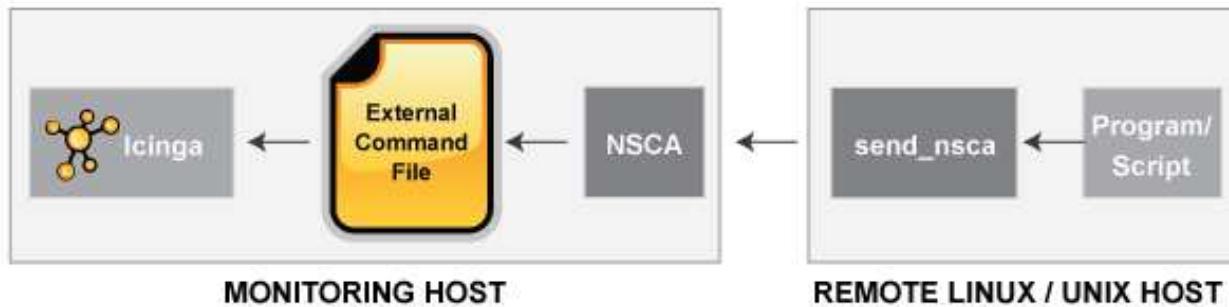
[Weiter](#)

NSCA

Einführung

Nagios Service Check Acceptor (oder kurz NSCA) ist ein Addon, um Prüfergebnisse von einem System zu einem anderen zu übertragen. Es besteht aus zwei Modulen: Dem Sender (`send_nsca`) und dem Empfänger (`nsca`). Die Daten werden verschlüsselt übertragen. Trotz des Namens werden auch Host-Prüfergebnisse übertragen.

Abbildung 10.3. NSCA



NSCA läuft als Daemon auf dem Icinga-Server. Er lauscht auf Informationen, die von entfernten Maschinen mit Hilfe des `send_nsca`-Programms (auf Unix/Linux-Maschinen) oder NSClient++ (oft auf Windows-Maschinen benutzt) gesendet werden. Die Daten werden mit der in `send_nsca.cfg` (oder `nsc.ini` im Falle von NSClient++) festgelegten Methode verschlüsselt. Der Daemon wird die Daten in einer *sehr* einfachen Weise validieren, indem die Informationen mit dem in `nsca.cfg` eingetragenen Passwort entschlüsselt werden. Wenn die Daten so aussehen, als seien sie mit dem gleichen Passwort verschlüsselt worden, dann wird der Daemon versuchen, diese Daten als externen Befehl in die lokale Icinga-Command-Pipe auszugeben.

Die folgenden Anweisungen basieren hauptsächlich auf dem README aus dem NSCA-Paket.

Voraussetzungen

- Icinga sollte laufen
- `check_external_commands = 1` sollte in `icinga.cfg` gesetzt sein

- *command_check_interval = <n>[s]* sollte in *icinga.cfg* gesetzt sein
- *log_passive_checks = 1* sollte während der Testphase in *icinga.cfg* gesetzt sein, anderenfalls gibt es keine Meldungen über eintreffende passive Prüfungen
- die libmcrypt- und libmcrypt-devel-Pakete sind installiert (die je nach Distribution ggf. anders heißen können), anderenfalls nutzen Sie einen der folgenden Befehle, um die Pakete zu installieren:

```
#> apt-get install libmcrypt libmcrypt-devel # Debian / Ubuntu
#> yum install libmcrypt libmcrypt-devel      # RHEL / Fedora / CentOS
#> zypper install libmcrypt libmcrypt-devel    # SLES / OpenSuSE (oder benutzen Sie Yast)
```

Download und Compile



Anmerkung

Anstatt NSCA von Grund auf zu installieren möchten Sie vielleicht ein Paket benutzen, das es möglicherweise für Ihr Betriebssystem gibt. Bitte werfen Sie einen Blick auf die [Tabelle der Pakete](#).

Falls Sie aus den Sourcen installieren möchten, dann benutzen Sie bitte die offiziellen Release-Tarballs.



Wichtig

Bitte benutzen Sie keine GIT-Snapshots, solange Sie kein Problem haben, das in der aktuellen Entwicklerversion ggf. gelöst ist.

Die maximale Länge der zu übertragenen Daten ist auf 2.048 Bytes begrenzt, die maximale Länge von Plugin-Ausgaben auf 512 Bytes. Falls das nicht ausreicht, dann müssen Sie den entsprechenden Wert in *icinga-nsca/include/common.h* anpassen.

```
#define MAX_INPUT_BUFFER          2048     /* max size of most buffers we use */
#define MAX_PLUGINOUTPUT_LENGTH   512
```

Bitte denken Sie daran, dass Sie die Programme erneut kompilieren müssen, wenn Sie sich zu einem späteren Zeitpunkt für eine Änderung entscheiden.

Bedingt durch den folgenden Wert in *include/common.h* (im Icinga-Core) kann der maximale Wert 8.192 Bytes nicht überschreiten.

```
#define MAX_EXTERNAL_COMMAND_LENGTH 8192     /* max length of an external command */
```

Nach Änderung des Besitzers wechseln Sie in das neu erstellte Verzeichnis und rufen Sie *configure* und *make* auf

```
#> chown -R icinga icinga-nsca
#> cd icinga-nsca
#> ./configure
#> make all
```

Anschließend gibt es zwei Programme (*send* und *send_nsca*) im *src*-Verzeichnis.



Achtung

Wenn die libmcrypt-Dateien nicht gefunden werden, dann wird "./configure" sich beschweren, aber NICHT mit einem Return-Code ungleich Null enden, so dass Sie *config.log* mit dem folgenden Befehl prüfen sollten

```
#> grep mcrypt.h: config.log
```

Dieser Befehl sollte keine Zeilen ausgeben.

Falls die libmcrypt-Module nicht gefunden werden, Benutzer oder Gruppe vom Wert "icinga" abweichen oder der zu benutzende Port nicht dem Default 5667 entspricht, dann können Sie dies über verschiedene Optionen beeinflussen. Rufen Sie "./configure -h" auf, um mehr über die verfügbaren Optionen zu erfahren.

Nach dem Wechsel in "nsca_tests" können Sie versuchen, "./runtests" auszuführen. Bitte beachten Sie, dass diese Tests verschiedene Perl-Module benötigen, die in der Datei README beschrieben sind.

Anpassen

Das sample-config-Verzeichnis enthält *nsca.cfg* und *send_nsca.cfg*. Mindestens die Einstellungen der Direktiven "password" und "encryption_method" / "decryption_method" sollten Sie ansehen/verändern, bevor die Dateien kopiert werden. Bitte denken Sie daran, das gleiche Passwort in allen Kopien dieser Konfigurationsdateien zu setzen. Wenn Sie verschiedene Passwörte auf verschiedenen entfernten Servern einsetzen möchten, dann müssen Sie mehrere nsca-Daemons auf dem Icinga-Server starten, die auf unterschiedlichen Ports lauschen. Das funktioniert nicht, wenn Sie den Daemon über inetd/xinetd starten.

Erster Test

Wechseln Sie zum Icinga-Benutzer und starten Sie einen ersten Test

```
#> su - icinga
$> cd /usr/src/icinga-nsca/src
$> ./nsca -c ../sample-config/nsca.cfg
$> echo -e "A\tB\tC\tD\n" | ./send_nsca -H localhost -c ../sample-config/send_nsca.cfg
$> exit
```

Dies sollte die Meldung "1 data packet(s) sent to host successfully." zurückliefern. Eigentlich heißt das nur, dass sich *send_nsca* und *nsca* unter Verwendung der Konfigurationsdateien auf dem lokalen Host miteinander unterhalten können, denn dieser Test funktioniert auch ohne eine lauffähige Icinga-Instanz. Allerdings ist er trotzdem wichtig: Wenn dieser Test fehlschlägt, dann ist es nicht sinnvoll fortzufahren. Prüfen Sie stattdessen die Einstellungen *nsca.cfg* und *send_nsca.cfg*. Schauen Sie auch im Syslog (z.B. */var/log/messages*) nach Meldungen.

Wenn die Voraussetzungen erfüllt sind, dann sollten Sie einige Warnungen in *icinga.log* sehen, dass Host "A" und Service "B" nicht in der Icinga-Konfiguration gefunden werden konnten. Dies bedeutet, dass *nsca* ausreichende Berechtigungen hat, um in das Icinga-Command-File zu schreiben. Prüfen Sie, ob der *nsca*-Daemon und Icinga mit unterschiedlichen Benutzern laufen, wenn es keine Meldungen in *icinga.log* gibt. Prüfen Sie außerdem die Einstellung von [log_passive_checks](#) in *icinga.cfg*.

Installation

"make install" macht (im Moment) nichts, so dass Sie selbst einige Dateien kopieren müssen. Die folgenden Befehle kopieren das nsca-Modul in das Verzeichnis, in dem das Icinga-Binary zu finden ist und die Konfigurationsdatei in den Icinga-Konfigurationsordner. Wir nehmen an, dass Sie Icinga nach einer der Schnellstartanleitungen installiert haben.

```
#> cp -p nsca /usr/local/icinga/bin/
#> cp ./sample-config/nsca.cfg /usr/local/icinga/etc/
```

- **nsca daemon**

Wenn Sie sich für xinetd entscheiden, dann wird der Daemon automatisch gestartet. Andernfalls müssen Sie den Daemon manuell starten, nachdem Sie zum Icinga-Benutzer gewechselt haben (was Sie ggf. bereits in "Erster Test" gemacht haben).

```
#> su - icinga
$> /usr/local/icinga/bin/nsca -c /usr/local/icinga/etc/nsca.cfg
```

- **inetd/xinetd**

Wenn Sie möchten, dass der Daemon von (x)inetd gestartet wird, dann müssen Sie /etc/services erweitern, eine weitere Datei ändern/kopieren und (x)inetd neustarten. Vergessen Sie nicht, den nsca-Daemon zu stoppen, der in "Erster Test" gestartet wurde.



Anmerkung

Die Einstellung von "server_port" in nsca.cfg wird ignoriert, wenn Sie inetd/xinetd benutzen.

```
#> kill < /var/run/nsca.pid
#> echo "nsca 5667/tcp # NSCA" >> /etc/services
```

Abhängig vom verwendeten Superserver gibt es drei Alternativen

- **inetd MIT tcpwrappers**

Fügen Sie Einträge zu Ihrer /etc/hosts.allow und /etc/hosts.deny hinzu, um TCP wrapper-Protection für den nsca-Service zu aktivieren. Dies ist optional, allerdings sehr empfohlen. Fügen Sie "nsca stream tcp nowait <user> /usr/sbin/tcpd <nsca-binary> -c <nsca-cfg> --inetd" zur /etc/inetd.conf hinzu, z.B.

```
#> echo "nsca stream tcp nowait icinga /usr/sbin/tcpd /usr/local/icinga/bin/nsca -c /usr/local/icinga/etc/nsca.cfg --inetd" >> /etc/inetd.conf
#> /etc/init.d/inetd restart
```

- **inetd OHNE tcpwrappers**

Fügen Sie "nsca stream tcp nowait <user> <nsca-binary> -c <nsca-cfg> --inetd" zur /etc/inetd.conf hinzu, z.B.

```
#> echo "nsca stream tcp nowait icinga /usr/local/icinga/bin/nsca -c /usr/local/icinga/etc/nsca.cfg --inetd" >> /etc/inetd.conf
#> /etc/init.d/inetd restart
```

- **xinetd**

Denken Sie daran, die Konfigurationsdatei nsca.xinetd im sample-config-Verzeichnis zu editieren und <ipaddress1> durch die IP-Adressen Ihrer Client-Rechner (auf denen send_nsca laufen wird) zu ersetzen. Dies funktioniert nur, wenn xinetd mit Unterstützung für tcpwrapper kompiliert wurde. Wenn Sie DHCP benutzen, funktioniert das nicht und Sie sollten diese Zeile löschen.

Fügen Sie Einträge zu Ihrer `/etc/hosts.allow` und `/etc/hosts.deny` hinzu, um TCP wrapper-Protection für den nsca-Service zu aktivieren. Dies ist optional, allerdings sehr empfohlen. Fügen Sie "nsca stream tcp nowait <user> /usr/sbin/tcpd <nsca-binary> -c <nsca-cfg> --inetd" zur `/etc/inetd.conf` hinzu, z.B.

```
#> cp -p ..../sample-config/nsca.xinetd /etc/xinetd.d/
#> /etc/init.d/xinetd restart
```

Entfernte Systeme

Sie sind mit dem lokalen System fertig, aber natürlich muss `send_nsca` noch auf entfernte Systeme kopiert werden.

Bitte denken Sie daran, dass `send_nsca` für die Zielplattform kompiliert werden muss, so dass Sie ggf. die libmcrypt-Pakete und `configure/make` auf mehreren Servern installieren bzw. ausführen müssen.

Dateien kopieren

Sie können frei entscheiden, wo Sie Binary und Konfigurationsdatei ablegen möchten. Wir nehmen an, dass Sie eine Verzeichnisstruktur haben, die ähnlich zum Icinga-Server ist.

```
#> scp -p <Icinga server>:/usr/local/icinga-nscsa/src/send_nsca /usr/local/icinga/bin/
#> scp -p <Icinga server>:/usr/local/icinga-nscsa/sample-config/send_nsca.cfg /usr/local/icinga/etc/
```

Zweiter Test

Nun können Sie den Test auf dem entfernten System ausführen

```
#> su - icinga
$> echo -e "A\tB\tC\tD\n" | /usr/local/icinga/bin/send_nsca -H <Icinga server> -c /usr/local/icinga/etc/send_nsca.cfg
```

Dies sollte ebenfalls die Meldung "1 data packet(s) sent to host successfully." liefern und es sollte Warnungen im `icinga.log` auf dem Icinga-Server geben, die ähnlich zu den o.g. sind. Wenn es keine Meldungen gibt, dann prüfen Sie die Einstellung von `log_passive_checks` in `icinga.cfg`.

Fehlersuche

Wenn der Daemon nicht berechtigt ist, in die Command-Pipe zu schreiben, dann sind die Daten verloren! Der Daemon sollte mit dem gleichen Benutzer laufen wie Icinga.

Wenn das Objekt (Host und/oder Service) nicht in der laufenden Konfiguration enthalten ist, werden die Daten verworfen.

Host-Name (und Service-Beschreibung, falls zutreffend) sind Case-sensitiv und müssen exakt mit den Definitionen in Icinga übereinstimmen.

Prüfen Sie, ob Sie in `nsca.cfg` und `send_nsca.cfg` das gleiche Passwort angegeben haben. Andernfalls wird die Übertragung fehlschlagen.

Prüfen Sie, ob Sie gleiche Verschlüsselungs-/Entschlüsselungsmethode verwenden. Andernfalls wird die Übertragung fehlschlagen.

Prüfen Sie, ob Ihre Firewall-Einstellungen die Kommunikation über den angegebenen Port zulassen (Default ist 5667)

Wenn Sie xinetd verwenden, dann prüfen Sie, ob die hinter "only_from=" angegebenen IP-Adressen zu den entfernten Systemen passen oder entfernen Sie diese Zeile (und starten Sie xinetd neu).

Aktivieren Sie "debug=1" in nsca.cfg, starten Sie den Daemon (neu) und schauen Sie nach Meldungen im Syslog / in nsca.log.

Sicherheit

Es gibt einige Sicherheitsimplikationen, wenn Sie entfernten Clients erlauben, Prüfergebnisse an Icinga zu senden. Daher gibt es die Möglichkeit, die Pakete zu verschlüsseln, die der NSCA-Client an den NSCA-Daemon sendet. Lesen Sie die SECURITY-Datei, um weitere Informationen über die Sicherheitsrisiken zu erhalten, die durch den Betrieb von NSCA auftreten können, zusammen mit einer Erklärung, welche Art von Schutz die Verschlüsselung Ihnen bietet.

Betrieb

send_nsca wird benutzt, um die Prüfergebnisse vom entfernten Rechner zum Icinga-Server zu senden. Die Syntax hängt vom Objekttyp ab. Um Service-Prüfergebnisse zu versenden, benutzen Sie

<host_name>[tab]<svc_description>[tab]<return_code>[tab]<plugin_output>[newline]

wobei:

<host_name>=der Kurzname des Hosts, mit dem der Service verbunden ist (wie in der host_name-Direktive der Service-Definition angegeben)

<svc_description>=Beschreibung des Service (wie in der service_description-Direktive der Service-Definition angegeben)

<return_code>=numerischer Return-Code (0,1,2,3 wie [hier](#) beschrieben)

<plugin_output>=Ausgabe des Service-Check

Host-Prüfergebnisse werden in einer ähnlichen Form versandt - lassen Sie einfach die Service-Beschreibung weg:

<host_name>[tab]<return_code>[tab]<plugin_output>[newline]

Integration in Icinga

Bisher haben Sie lediglich einige Voraussetzungen geschaffen, um passive Prüfergebnisse zu übermitteln, aber Sie haben noch keinen Host oder Service definiert, der diese Funktionalität nutzt.

Obwohl Sie nur Prüfergebnisse empfangen müssen Sie doch die "check_command"-Direktive in Ihren Definitionen angeben. Es gibt ein Plugin namens "check_dummy", das für diesen Zweck benutzt werden kann. Es kann sein, dass Sie die folgende command-Definition einfügen müssen, falls sie noch nicht vorhanden ist. Das zweite Argument ist optional und kann z.B. einen erklärenden Text enthalten.

```
define command{
    command_name check_dummy
    command_line $USER1$/check_dummy $ARG1$ $ARG2$
}
```

Velleicht möchten Sie ein Service-Template anlegen. Das Host-Template könnte ähnlich aussehen (ersetzen Sie einfach "service" durch "host")

```
define service{
    use generic-service      ; template to inherit from
    name passive-service    ; name of this template
    active_checks_enabled  0   ; no active checks
    passive_checks_enabled 1   ; allow passive checks
    check_command           check_dummy!0 ; use "check_dummy", RC=0 (OK)
    check_period             24x7 ; check active all the time
    check_freshness          0   ; don't check if check result is "stale"
    register                 0   ; this is a template, not a real service
}
```

Zusammen mit dem obigen Template könnte die Service-Definition wie folgt aussehen:

```
define service{
    use     passive-service      ; template to inherit from
    host_name remotehost        ; host where send_nsca is located
    service_description Diskspace ; service to be checked
}
```

Starten Sie Icinga erneut, damit die Änderungen in der laufenden Konfiguration enthalten sind

```
#> /etc/init.d/icinga restart
```

Wechseln Sie auf Ihrem entfernten Host zum Icinga-Benutzer und führen Sie send_nsca aus. Ersetzen Sie dabei <Icinga server> durch die IP-Adresse des Icinga-Servers

```
#> su - icinga
$> echo -e "remotehost\tDiskspace\t0\t/var=78%\n" | /usr/local/icinga/bin/send_nsca -H <Icinga server> -c /usr/local/icinga/etc/send_nsca.cfg
```

Bitte denken Sie daran, dass Host-Name und Service-Beschreibung exakt den Angaben in Ihrer Icinga-Definition entsprechen müssen (die Angaben sind Case-sensitiv). Andernfalls bekommen Sie Meldungen in icinga.log, dass das Objekt nicht gefunden werden kann. Falls es keine Meldungen gibt, kontrollieren Sie die Einstellung von `log_passive_checks` in `icinga.cfg`.

Nach einem kurzen Moment sollten Sie Meldungen in icinga.log sehen, dass die gesendeten Informationen verarbeitet wurden. Sie sollten die Daten dann auch im klassischen Web-Interface sehen sowie feststellen, dass der Service-Status von "Pending" auf "OK" gewechselt ist und die Ausgabe die Daten enthält, die Sie versandt haben.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[NRPE](#)

[Zum Anfang](#)

[Kapitel 11. Entwicklung](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Kapitel 11. Entwicklung

[Zurück](#)

[Weiter](#)

Kapitel 11. Entwicklung

Inhaltsverzeichnis

[Nagios Plugin API](#)

[Entwickeln von Plugins für die Nutzung mit Embedded Perl](#)

[Liste der externen Befehle](#)

[Installation und Benutzung der Icinga-API](#)

[Die Icinga-Web REST API](#)

[Zurück](#)

[Weiter](#)

[NSCA](#)

[Zum Anfang](#)

[Nagios Plugin API](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Nagios Plugin API

[Zurück](#)[Kapitel 11. Entwicklung](#)[Weiter](#)

Nagios Plugin API

Andere Ressourcen

Wenn Sie planen, Ihren eigenen Plugins für Icinga zu schreiben, dann besuchen Sie folgende Ressourcen:

- Die offizielle [Nagios-Plugin-Projekt-Website](#)
- Die offiziellen [Nagios-Plugin-Entwicklungsrichtlinien](#)

Plugin-Überblick

Scripts und ausführbare Programme müssen (mindestens) zwei Dinge tun, um als Icinga-Plugins zu funktionieren:

- mit einem von verschiedenen möglichen Return-Codes enden
- mindestens eine Zeile Textausgabe an STDOUT zurückliefern

Die inneren Abläufe Ihres Plugins sind für Icinga unwichtig. Ihr Plugin könnte den Zustand eines TCP-Ports prüfen, eine Datenbankabfrage durchführen, den freien Plattenplatz ermitteln oder was immer benötigt wird, um etwas zu prüfen. Die Einzelheiten hängen davon ab, was zu prüfen ist - das liegt an Ihnen.

Return-Code

Icinga ermittelt den Zustand eines Hosts oder Service über die Auswertung des Return-Codes des Plugins. Die folgenden Tabellen zeigen eine Liste von gültigen Return-Codes zusammen mit ihren entsprechenden Service- oder Host-Zuständen.

Plugin Return-Code	Service-Zustand	Host-Zustand
0	OK	UP
1	WARNING	UP oder DOWN/UNREACHABLE*
2	CRITICAL	DOWN/UNREACHABLE
3	UNKNOWN	DOWN/UNREACHABLE

 Anmerkung: Wenn die `use_aggressive_host_checking`-Option aktiviert ist, dann ergibt ein Return-Code von 1 einen Host-Zustand "DOWN" oder "UNREACHABLE". Andernfalls ergibt ein Return-Code von 1 einen Host-Zustand "UP". Der Prozess, durch den Icinga ermittelt, ob ein Host DOWN oder UNREACHABLE ist, wird [hier](#) erklärt.

Spezifikation der Plugin-Ausgabe(n)

Als Minimum sollten Plugins mindestens eine Zeile Textausgabe zurückliefern, es können aber auch mehrere Zeilen Ausgaben sein. Plugins können zusätzlich Performance-Daten zurückliefern, die von externen Applikationen verarbeitet werden können. Das grundlegende Format für Plugin-Ausgaben ist wie folgt:

TEXT OUTPUT | OPTIONAL PERFDATA

LONG TEXT LINE 1 LONG TEXT LINE 2 ... LONG TEXT LINE N | PERFDATA LINE 2

PERFDATA LINE 3 ... PERFDATA LINE N

Die Performance-Daten (in orange dargestellt) sind optional. Wenn ein Plugin Performance-Daten in der Ausgabe zurückliefert, dann müssen die Performance-Daten von den anderen Textausgaben mit einem Pipe-Symbol (!) getrennt werden. Zusätzliche Zeilen von langen Textausgaben (in blau dargestellt) sind ebenso optional.

Plugin-Beispielausgaben

Nun ein paar Beispiele von möglichen Plugin-Ausgaben...

Fall 1: Eine Zeile Ausgabe (nur Text)

Angenommen, wir haben ein Plugin, das eine Zeile ausgibt, dann sieht das wie folgt aus:

DISK OK - free space: / 3326 MB (56%); Wenn dieses Plugin benutzt wurde, um eine Service-Prüfung durchzuführen, wird die gesamte Zeile der Ausgabe im `$SERVICEOUTPUT$`-Makro gespeichert.

Fall 2: Eine Zeile Ausgabe (Text und Performance-Daten)

Ein Plugin kann optionale Performance-Daten zurückliefern, die von externen Applikationen benutzt werden. Um dies zu tun, müssen die Performance-Daten von der Textausgabe durch ein Pipe-Symbol (!) wie folgt getrennt werden:

DISK OK - free space: / 3326 MB (56%); ! /=2643MB;5948;5958;0;5968. Wenn dieses Plugin benutzt wurde, um eine Service-Prüfung durchzuführen, wird der **rote** Teil der Ausgabe (links vom Pipe-Symbol) im `$SERVICEOUTPUT$`-Makro und der **orange** Teil der Ausgabe (rechts vom Pipe-Symbol) im `$SERVICEPERFDATA$`-Makro gespeichert.

Fall 3: Mehrere Zeilen Ausgaben (Text und Performance-Daten)

Ein Plugin kann optional mehrere Zeilen von Text und Performance-Daten wie folgt zurückliefern:

DISK OK - free space: / 3326 MB (56%);

! /=2643MB;5948;5958;0;5968

```
/ 15272 MB (77%);  

/boot 68 MB (69%);  

/home 69357 MB (27%);  

/var/log 819 MB (84%); | /boot=68MB;88;93;0;98  

/home=69357MB;253404;253409;0;253414  

/var/log=818MB;970;975;0;980
```

Wenn dieses Plugin benutzt wurde, um eine Service-Prüfung durchzuführen, wird der **rote** Teil der ersten Zeile der Ausgabe (links vom Pipe-Symbol) im **\$SERVICEOUTPUT\$**-Makro gespeichert. Der **orange** Teil der ersten und folgender Zeilen wird (durch Leerzeichen verbunden) im **\$SERVICEPERFDATA\$**-Makro gespeichert. Der **blaue** Teil der zweiten bis fünften Zeile der Ausgabe wird (mit maskierten Newlines) verkettet und im **\$LONGSERVICEOUTPUT\$**-Makro gespeichert.

Der endgültige Inhalt jedes Makros ist wie folgt:

Makro	Wert
\$SERVICEOUTPUTS	DISK OK - free space: / 3326 MB (56%);
\$SERVICEPERFDATA\$	/=2643MB;5948;5958;0;5968 /boot=68MB;88;93;0;98 /home=69357MB;253404;253409;0;253414 /var/log=818MB;970;975;0;980
\$LONGSERVICEOUTPUT\$	/ 15272 MB (77%);\n/boot 68 MB (69%);\n/var/log 819 MB (84%);

Mit Blick auf mehrere Zeilen Ausgaben haben Sie die folgenden Möglichkeiten, Performance-Daten zurückzuliefern:

- Sie können keinerlei Performance-Daten zurückliefern
- Sie können nur in der ersten Zeile Performance-Daten zurückliefern
- Sie können Performance-Daten in nachfolgenden Zeilen zurückliefern (nach der ersten)
- Sie können Performance-Daten in der ersten und folgenden Zeilen zurückliefern (wie oben gezeigt)

Längenbeschränkungen von Plugin-Ausgaben

Icinga wird nur die ersten acht KB an Daten lesen, die ein Plugin zurückliefert. Dies wird getan, um durchgedrehte Plugins davon abzuhalten, Megabyte oder Gigabyte an Daten an Icinga zurückzuliefern. Diese Beschränkung von acht KB kann einfach geändert werden, wenn Sie das brauchen. Ändern Sie einfach den Wert der MAX_PLUGIN_OUTPUT_LENGTH-Definition in der *include/nagios.h.in*-Datei der Source-Code-Distribution und rekomplizieren Sie Icinga. Wenn Sie die Kapazität von acht KB durch Anpassung dieses Wertes ändern, dann stellen Sie sich, dass Sie vor dem Kompilieren auch den Wert von MAX_EXTERNAL_COMMAND_LENGTH in *include/common.h* erhöhen, damit Resultate von passiven Prüfungen in dieser Länge durch das External Command File empfangen werden können.

Beispiele

Wenn Sie nach Beispiel-Plugins suchen, um sie zu studieren, würden wir empfehlen, dass Sie die offiziellen Icinga-Plugins herunterladen und den Code von verschiedenen C-, Perl- und Shell-Script-Plugins ansehen. Informationen, wie Sie die offiziellen Plugins besorgen können, finden Sie [hier](#).

Perl-Plugins

Icinga bietet einen optionalen [eingebauten Perl-Interpreter](#) (embedded Perl interpreter), der die Ausführung von Perl-Plugins beschleunigen kann. Mehr Informationen zur Entwicklung von Perl-Plugins zur Nutzung mit dem eingebauten Perl-Interpreter finden Sie [hier](#).

[Zurück](#)[Nach oben](#)[Weiter](#)[Kapitel 11. Entwicklung](#)[Zum Anfang](#)[Entwickeln von Plugins für die Nutzung mit Embedded Perl](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>

**Entwickeln von Plugins für die Nutzung mit Embedded Perl**[Zurück](#)**Kapitel 11. Entwicklung**[Weiter](#)**Entwickeln von Plugins für die Nutzung mit Embedded Perl****Einführung**

Stanley Hopcroft hat ziemlich viel mit dem eingebetteten Perl-Interpreter gearbeitet und die Vor- und Nachteile der Nutzung kommentiert. Er hat auch verschiedene Hinweise gegeben, um Perl-Plugins zu erstellen, die sauber mit dem eingebetteten Interpreter laufen. Der überwiegende Teil dieser Dokumentation stammt aus seinen Kommentaren.

Es ist anzumerken, dass sich "ePN", wie in dieser Dokumentation verwendet, auf den eingebetteten Perl-Interpreter, oder wenn Ihnen das lieber ist, auf Icinga kompiliert mit einem eingebetteten Perl-Interpreter bezieht.

Zielgruppe

- Durchschnittliche Perl-Entwickler mit einem Verständnis für die mächtigen Eigenschaften der Sprache ohne Wissen der Interna bzw. einem vertieften Wissen dieser Eigenschaften.
- die mit einem benutzenden Wissen statt einem tiefen Verständnis
- wenn Sie glücklich sind mit Perl-Objekten, sprich Verwaltung, Datenstrukturen und dem Debugger, dann ist das wahrscheinlich ausreichend.

Dinge, die Sie tun sollten, wenn Sie ein Perl-Plugin entwickeln (mit ePN oder ohne)

- generieren Sie immer etwas Output
- Verwenden Sie 'use utils' und importieren Sie die Dinge, die es exportiert (\$TIMEOUT %ERRORS &print_revision &support)
- Werfen Sie einen Blick darauf, wie die Standard-Plugins ihren Kram erledigen
 - beenden Sie immer mit \$ERRORS{CRITICAL}, \$ERRORS{OK}, etc.
 - verwenden Sie getopt, um Kommandozeilenparameter einzulesen
 - denken Sie an Timeout-Verwaltung
 - rufen Sie print_usage auf (das Sie liefern müssen), wenn keine Kommandozeilenparameter übergeben wurden

- benutzen Sie Standard-Optionen (eg H 'host', V 'version')

Dinge, die Sie tun müssen, um ein Perl-Plugin für ePN zu entwickeln

1. <DATA> kann nicht verwendet werden, benutzen Sie statt dessen here-Dokumente, z.B.

```
my $data = <<DATA;
portmapper 100000
portmap 100000
sunrpc 100000
rpcbind 100000
rstatd 100001
rstat 100001
rup 100001
..
DATA
%proignum = map { my($a, $b) = split; ($a, $b) } split(/\n/, $data) ;
```

2. BEGIN-Blöcke werden nicht so funktionieren, wie Sie das erwarten. Es wird das Beste sein, wenn Sie darauf verzichten.

3. stellen Sie sicher, dass es während des Compile absolut sauber ist, d.h.

- use strict
- use perl -w (andere Switches [namentlich T] könnten nicht weiterhelfen)
- use perl -c

4. Vermeiden Sie lexikalische Variablen (my) mit globalem Geltungsbereich, um damit variable Daten in Unterrountinen zu übergeben. Das ist in der Tat fatal, wenn die Unterroutine mehrfach aufgerufen wird, während die Prüfung läuft. Solche Unterrountinen arbeiten als 'closures', die den ersten Wert der globalen lexikalischen Variable bei folgenden Aufrufen der Unterroutine beibehalten. Wenn die globale Variable allerdings read-only ist (bei einer komplizierten Struktur zum Beispiel), dann ist das kein Problem. Was Bekman Ihnen statt dessen rät, ist eines der folgenden Dinge:

- machen Sie die Unterroutine anonym und rufen Sie sie z.B. über eine Code-Referenz auf

```
ändern Sie dies           in
my $x = 1 ;
sub a { .. Process $x ... } $a_cr = sub { ... Process $x ... } ;
.
.
a ;
$x = 2
a ;
# anonyme Closures binden immer den aktuellen lexikalischen Wert ein
&$a_cr ;
```

- packen Sie globale Lexikale und die Unterroutine, die sie benutzt, in ihr eigenes Package (als ein Objekt oder Modul)
- übergeben Sie Informationen an Unterrountinen als Referenzen oder Aliases (\\$lex_var oder \$_[n])
- ersetzen Sie Lexikale durch Package Globals und schließen Sie diese von 'use strict'-Beanstandungen durch 'use vars qw(global1 global2 ..)' aus

5. Seien Sie sich bewusst, woher Sie mehr Informationen bekommen können.

Nützliche Informationen können Sie von den üblichen Verdächtigen bekommen (die O'Reilly-Buch, plus Damien Conways "Object Oriented Perl"), aber um den wirklich nützlichen Kram im richtigen Kontext zu bekommen, starten Sie mit Stas Bekman's mod_perl guide unter <http://perl.apache.org/guide/>.

Dieses wundervolle Dokument in Buchgröße hat überhaupt nichts mit Icinga zu tun, aber dafür umso mehr mit dem Schreiben von Perl-Programmen für den eingebetteten Perl-Interpreter in Apache (d.h. Doug MacEacherns mod_perl).

Die perlembed-Manpage ist wichtig für den Zusammenhang und die Ermunterung..

Auf der Basis, dass Lincoln Stein und Doug MacEachern ein oder zwei Dinge über Perl und eingebettetes Perl wissen, ist ihr Buch 'Writing Apache Modules with Perl and C' ziemlich sicher einen Blick wert.

6. Achten Sie darauf, dass Ihr Plugin mit ePN vielleicht merkwürdige Werte zurückliefert und dass das wahrscheinlich an dem unter Punkt 4 angesprochenen Problem liegt

7. Seien Sie darauf vorbereitet, dass Sie debuggen über:

- ein Test-ePN und
- print-Befehle in Ihr Plugin einfügen, um Variablenwerte auf STDERR auszugeben (da Sie STDOUT nicht verwenden können)
- print-Befehle in p1.pl einfügen, um anzuzeigen, was ePN glaubt, was Ihr Plugin ist, bevor es versucht, das auszuführen (vi)
- ePN im Vordergrund-Modus auszuführen (möglicherweise in Verbindung mit den obigen Empfehlungen)
- das 'Deparse'-Modul in Ihrem Modul zu benutzen, um zu sehen, wie der Parser es optimiert hat und was der Interpreter wirklich bekommt (lesen Sie 'Constants in Perl' von Sean M. Burke, The Perl Journal, Fall 2001)

```
perl -MO::Deparse <your_program>
```

8. Beachten Sie, in was ePN Ihr Plugin transformiert, und falls alles andere fehlschlägt, debuggen Sie die transformierte Version.

Wie Sie unten sehen können, schreibt p1.pl Ihr Plugin um in eine Unterroutine namens 'hndlr' im Package 'Embed::<something_related_to_your_plugin_file_name>'.

Ihr Plugin wird ggf. Kommandozeilenparameter in @ARGV erwarten, so dass p1.pl auch @_ an @ARGV zuweist.

Dies wiederum wird 'eval'-t und falls dieser Test mit einem Fehler fehlschlägt (jeder Parse- oder Laufzeitfehler), wird das Plugin 'rausgeschmissen'.

Die folgenden Ausgaben zeigen, wie ein Test-ePN das *check_rpc*-Plugin transformiert hat, bevor es versucht, es auszuführen. Der meiste Code des eigentlichen Plugins wird nicht gezeigt, weil wir nur an den Umformungen interessiert sind, die der ePN am Plugin vorgenommen hat). Zur Verdeutlichung sind die Umformungen in rot dargestellt:

```

package main;
use subs 'CORE::GLOBAL::exit';
sub CORE::GLOBAL::exit { die "ExitTrap: $_[0] (Embed:::check_5frpc)"; }
package Embed:::check_5frpc; sub hndlr { shift(@_) ;
@ARGV=@_;
#! /usr/bin/perl -w
#
# check_rpc plugin for Icinga
#
# usage:
#   check_rpc host service
#
# Check if an rpc service is registered and running
# using rpcinfo - $proto $host $proignum 2>&1 |";
#
# Use these hosts.cfg entries as examples
#
# command[check_nfs]=/some/path/libexec/check_rpc $HOSTADDRESS$ nfs
# service[check_nfs]=NFS;24x7;3;5;5;unix-admin;60;24x7;1;1;1;;check_rpc
#
# initial version: 3 May 2000 by Truongchinh Nguyen and Karl DeBisschop
# current status: $Revision: 1.18 $
#
# Copyright Notice: GPL
#
... der Rest des Plugin-Codes folgt (und wurde aus Gründen der Kürze entfernt) ...
}

```

9. Nutzen Sie 'use diagnostics' nicht in einem produktiven ePN. Wir glauben, es sorgt dafür, dass alle Perl-Plugins CRITICAL zurückliefern.
 10. Überlegen Sie, ob Sie ein Mini-ePN benutzen, um Ihr Plugin zu testen. Das ist nicht ausreichend, um zu garantieren, dass Ihr Plugin mit einem ePN fehlerfrei ausgeführt wird, aber wenn bereits der Plugin-Test fehlschlägt, dann wird er auf jeden Fall mit Ihrem ePN fehlschlagen. [Ein Beispiel-Mini-ePN ist im *contrib*-Verzeichnis der Icinga-Distribution zu finden. Wechseln Sie in das *contrib*-Verzeichnis und tippen Sie 'make mini_epn', um es zu kompilieren. Es muss im gleichen Verzeichnis ausgeführt werden, in dem die p1.pl-Datei steht (diese Datei wird mit Icinga ausgeliefert).]
-

[Zurück](#)[Nach oben](#)[Weiter](#)[Nagios Plugin API](#)[Zum Anfang](#)[Liste der externen Befehle](#)© 2009-2011 Icinga Development Team, <http://www.icinga.org>

**Liste der externen Befehle**[Zurück](#)[Kapitel 11. Entwicklung](#)[Weiter](#)

Liste der externen Befehle

Nachfolgend finden Sie Beschreibungen für jeden externen Befehl. Es ist ziemlich einfach, externe Befehle an Icinga zu senden und Sie müssen jeweils nur die letzte Zeile des Beispiel-Scripts anpassen, um einen anderen Befehl zu nutzen (so dass wir auf Beispiele in den Beschreibungen verzichtet haben).

Example:

```
#!/bin/sh
# Passen Sie ggf. die Variablen auf Ihre Umgebung an.

now=`date +%s`
commandfile='/usr/local/icinga/var/rw/icinga.cmd'

/bin/printf "[%lu] ACKNOWLEDGE_HOST_PROBLEM;Host1;1;1;1;Max Mustermann;ein Kommentar\n" $now > $commandfile
```

ACKNOWLEDGE_HOST_PROBLEM

ACKNOWLEDGE_HOST_PROBLEM; <host_name>;<sticky>;<notify>;<persistent>;<author>;<comment>

Erlaubt Ihnen, das aktuelle Problem für den angegebenen Host zu bestätigen. Durch Bestätigung des aktuellen Problems werden weitere Benachrichtigungen (für den gleichen Host-Status) deaktiviert. Wenn die "sticky"-Option auf zwei (2) gesetzt wird, bleibt die Bestätigung bestehen, bis der Host in einen UP-Status zurückkehrt. Andernfalls wird die Bestätigung automatisch entfernt, wenn sich der Host-Status ändert. Wenn die "notify"-Option auf eins (1) gesetzt wird, werden die Kontakte über die Bestätigung informiert. Wenn die "persistent"-Option auf eins (1) gesetzt wird, wird der Kommentar zu dieser Bestätigung auch über Neustarts des Icinga-Prozesses hinweg aufbewahrt. Andernfalls wird der Kommentar beim nächsten Neustart des Icinga-Prozesses gelöscht.

ACKNOWLEDGE_HOST_PROBLEM_EXPIRE

ACKNOWLEDGE_HOST_PROBLEM_EXPIRE; <host_name>;<sticky>;<notify>;<persistent>;<timestamp>;<author>;<comment>

Erlaubt Ihnen, die Verfallszeit (in Sekunden seit der UNIX-Epoche) der Bestätigung (acknowledgement) zu setzen.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.6.

ACKNOWLEDGE_SVC_PROBLEM

`ACKNOWLEDGE_SVC_PROBLEM ;<host_name>;<service_description>;<sticky>;<notify>;<persistent>;<author>;<comment>`

Erlaubt Ihnen, das aktuelle Problem für den angegebenen Service zu bestätigen. Durch Bestätigung des aktuellen Problems werden weitere Benachrichtigungen (für den gleichen Service-Status) deaktiviert. Wenn die "sticky"-Option auf zwei (2) gesetzt wird, bleibt die Bestätigung bestehen, bis der Service in einen OK-Status zurückkehrt. Andernfalls wird die Bestätigung automatisch entfernt, wenn sich der Service-Status ändert. Wenn die "notify"-Option auf eins (1) gesetzt wird, werden die Kontakte über die Bestätigung informiert. Wenn die "persistent"-Option auf eins (1) gesetzt wird, wird der Kommentar zu dieser Bestätigung auch über Neustarts des Icinga-Prozesses hinweg aufbewahrt. Andernfalls wird der Kommentar beim nächsten Neustart des Icinga-Prozesses gelöscht.

ACKNOWLEDGE_SVC_PROBLEM_EXPIRE

`ACKNOWLEDGE_SVC_PROBLEM_EXPIRE ;<host_name>;<service_description>;<sticky>;<notify>;<persistent>;<timestamp>;<author>;<comment>`

Erlaubt Ihnen, die Verfallszeit (in Sekunden seit der UNIX-Epoche) der Bestätigung (acknowledgement) zu setzen.



Anmerkung

Diese Option ist verfügbar ab Icinga 1.6.

ADD_HOST_COMMENT

`ADD_HOST_COMMENT ;<host_name>;<persistent>;<author>;<comment>`

Fügt einen Kommentar zu einem bestimmten Host hinzu. Wenn die "persistent"-Option auf eins (1) gesetzt wird, wird der Kommentar zu diesem Host auch über Neustarts des Icinga-Prozesses hinweg aufbewahrt. Andernfalls wird der Kommentar beim nächsten Neustart des Icinga-Prozesses gelöscht.

ADD_SVC_COMMENT

`ADD_SVC_COMMENT ;<host_name>;<service_description>;<persistent>;<author>;<comment>`

Fügt einen Kommentar zu einem bestimmten Service hinzu. Wenn die "persistent"-Option auf eins (1) gesetzt wird, wird der Kommentar zu diesem Service auch über Neustarts des Icinga-Prozesses hinweg aufbewahrt. Andernfalls wird der Kommentar beim nächsten Neustart des Icinga-Prozesses gelöscht.

CHANGE_CONTACT_HOST_NOTIFICATION_TIMEPERIOD

`CHANGE_CONTACT_HOST_NOTIFICATION_TIMEPERIOD ;<contact_name>;<notification_timeperiod>`

Ändert das Host-Benachrichtigungs-Zeitfenster (timeperiod) für einen bestimmten Host auf das, was durch die "notification_timeperiod"-Option angegeben wird. Diese Option gibt den Kurznamen des Zeitfensters an, das als Host-Benachrichtigungs-Zeitfenster für den Kontakt benutzt werden soll. Das Zeitfenster muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_CONTACT_MODATTR

CHANGE_CONTACT_MODATTR ; <contact_name>;<value>

Dieser Befehl ändert den Wert für "modified attributes" für den angegebenen Kontakt. "Modified attributes"-Werte werden von Icinga benutzt, um festzulegen, welche Objekteigenschaften über Programmneustarts hinweg aufbewahrt werden sollen. Das bedeutet, dass die Änderung des Wertes die Datenaufbewahrung beeinflussen kann. Dies ist eine fortgeschrittene Option und sollte nur von Personen benutzt werden, die fundierte Kenntnisse von der Aufbewahrungslogik in Icinga besitzen.

CHANGE_CONTACT_MODHATTR

CHANGE_CONTACT_MODHATTR ; <contact_name>;<value>

Dieser Befehl ändert den Wert für "modified host attributes" für den angegebenen Kontakt. "Modified attributes"-Werte werden von Icinga benutzt, um festzulegen, welche Objekteigenschaften über Programmneustarts hinweg aufbewahrt werden sollen. Das bedeutet, dass die Änderung des Wertes die Datenaufbewahrung beeinflussen kann. Dies ist eine fortgeschrittene Option und sollte nur von Personen benutzt werden, die fundierte Kenntnisse von der Aufbewahrungslogik in Icinga besitzen.

CHANGE_CONTACT_MODSATTR

CHANGE_CONTACT_MODSATTR ; <contact_name>;<value>

Dieser Befehl ändert den Wert für "modified service attributes" für den angegebenen Kontakt. "Modified attributes"-Werte werden von Icinga benutzt, um festzulegen, welche Objekteigenschaften über Programmneustarts hinweg aufbewahrt werden sollen. Das bedeutet, dass die Änderung des Wertes die Datenaufbewahrung beeinflussen kann. Dies ist eine fortgeschrittene Option und sollte nur von Personen benutzt werden, die fundierte Kenntnisse von der Aufbewahrungslogik in Icinga besitzen.

CHANGE_CONTACT_SVC_NOTIFICATION_TIMEPERIOD

CHANGE_CONTACT_SVC_NOTIFICATION_TIMEPERIOD ; <contact_name>;<notification_timeperiod>

Ändert das Service-Benachrichtigungs-Zeitfenster (timeperiod) für einen bestimmten Service auf das, was durch die "notification_timeperiod"-Option angegeben wird. Diese Option gibt den Kurznamen des Zeitfensters an, das als Service-Benachrichtigungs-Zeitfenster für den Kontakt benutzt werden soll. Das Zeitfenster muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_CUSTOM_CONTACT_VAR

CHANGE_CUSTOM_CONTACT_VAR ; <contact_name>;<varname>;<varvalue>

Ändert den Wert einer benutzerdefinierten Kontakt-Variable.

CHANGE_CUSTOM_HOST_VAR

CHANGE_CUSTOM_HOST_VAR ; <host_name>;<varname>;<varvalue>

Ändert den Wert einer benutzerdefinierten Host-Variable.

CHANGE_CUSTOM_SVC_VAR

CHANGE_CUSTOM_SVC_VAR ;<host_name>;<service_description>;<varname>;<varvalue>

Ändert den Wert einer benutzerdefinierten Service-Variable.

CHANGE_GLOBAL_HOST_EVENT_HANDLER

CHANGE_GLOBAL_HOST_EVENT_HANDLER ;<event_handler_command>

Ändert den globalen Host-Eventhandler-Befehl auf den Befehl, der über "event_handler_command" angegeben wird. Diese Option gibt den Kurznamen des Befehls an, der als globaler Host-Eventhandler-Befehl benutzt werden soll. Der Befehl muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_GLOBAL_SVC_EVENT_HANDLER

CHANGE_GLOBAL_SVC_EVENT_HANDLER ;<event_handler_command>

Ändert den globalen Service-Eventhandler-Befehl auf den Befehl, der über "event_handler_command" angegeben wird. Diese Option gibt den Kurznamen des Befehls an, der als globaler Service-Eventhandler-Befehl benutzt werden soll. Der Befehl muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_HOST_CHECK_COMMAND

CHANGE_HOST_CHECK_COMMAND ;<host_name>;<check_command>

Ändert den Prüfbefehl für einen bestimmten Host auf das, was durch die "check_command"-Option angegeben wird. Diese Option gibt den Kurznamen des Befehls enthalten, der als Prüfbefehl für den Host benutzt werden soll. Der Befehl muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_HOST_CHECK_TIMEPERIOD

CHANGE_HOST_CHECK_TIMEPERIOD ;<host_name>;<timeperiod>

Ändert das Host-Prüf-Zeitfenster (timeperiod) für einen bestimmten Host auf das, was durch die "timeperiod"-Option angegeben wird. Diese Option gibt den Kurznamen des Zeitfensters an, das als Prüf-Zeitfenster für den Host benutzt werden soll. Das Zeitfenster muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_HOST_EVENT_HANDLER

CHANGE_HOST_EVENT_HANDLER ;<host_name>;<event_handler_command>

Ändert den Eventhandler-Befehl für einen bestimmten Host auf den Befehl, der über "event_handler_command" angegeben wird. Diese Option gibt den Kurznamen des Befehls an, der als neuer Eventhandler-Befehl für diesen Host benutzt werden soll. Der Befehl muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_HOST_MODATTR

CHANGE_HOST_MODATTR ;<host_name>;<value>

Dieser Befehl ändert den Wert für "modified attributes" für den angegebenen Host. "Modified attributes"-Werte werden von Icinga benutzt, um festzulegen, welche Objekteigenschaften über Programmneustarts hinweg aufbewahrt werden sollen. Das bedeutet, dass die Änderung des Wertes die Datenaufbewahrung beeinflussen kann. Dies ist eine fortgeschrittene Option und sollte nur von Personen benutzt werden, die fundierte Kenntnisse von der Aufbewahrungslogik

in Icinga besitzen

CHANGE_HOST_NOTIFICATION_TIMEPERIOD

CHANGE_HOST_NOTIFICATION_TIMEPERIOD ; <host_name>;<notification_timeperiod>

Ändert das Host-Benachrichtigungs-Zeitfenster (timeperiod) für einen bestimmten Host auf das, was durch die "notification_timeperiod"-Option angegeben wird. Diese Option gibt den Kurznamen des Zeitfensters an, das als Benachrichtigungs-Zeitfenster für den Host benutzt werden soll. Das Zeitfenster muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_MAX_HOST_CHECK_ATTEMPTS

CHANGE_MAX_HOST_CHECK_ATTEMPTS ; <host_name>;<check_attempts>

Ändert die maximale Anzahl von erneuten Prüf-Versuchen (Wiederholungen) für einen bestimmten Host.

CHANGE_MAX_SVC_CHECK_ATTEMPTS

CHANGE_MAX_SVC_CHECK_ATTEMPTS ; <host_name>;<service_description>;<check_attempts>

Ändert die maximale Anzahl von erneuten Prüf-Versuchen (Wiederholungen) für einen bestimmten Service.

CHANGE_NORMAL_HOST_CHECK_INTERVAL

CHANGE_NORMAL_HOST_CHECK_INTERVAL ; <host_name>;<check_interval>

Ändert das normale Prüfintervall (für regelmäßige Prüfungen) für einen bestimmten Host.

CHANGE_NORMAL_SVC_CHECK_INTERVAL

CHANGE_NORMAL_SVC_CHECK_INTERVAL ; <host_name>;<service_description>;<check_interval>

Ändert das normale Prüfintervall (für regelmäßige Prüfungen) für einen bestimmten Service

CHANGE_RETRY_HOST_CHECK_INTERVAL

CHANGE_RETRY_HOST_CHECK_INTERVAL ; <host_name>;<service_description>;<check_interval>

Ändert das Prüfintervall (für Wiederholungsprüfungen) für einen bestimmten Host.

CHANGE_RETRY_SVC_CHECK_INTERVAL

CHANGE_RETRY_SVC_CHECK_INTERVAL ; <host_name>;<service_description>;<check_interval>

Ändert das Prüfintervall (für Wiederholungsprüfungen) für einen bestimmten Service.

CHANGE_SVC_CHECK_COMMAND

CHANGE_SVC_CHECK_COMMAND ; <host_name>;<service_description>;<check_command>

Ändert den Prüfbefehl für einen bestimmten Service auf das, was durch die "check_command"-Option angegeben wird. Diese Option gibt den Kurznamen des Befehls enthalten, der als Prüfbefehl für den Service benutzt werden soll. Der Befehl muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_SVC_CHECK_TIMEPERIOD

CHANGE_SVC_CHECK_TIMEPERIOD ;<host_name>;<service_description>;<check_timeperiod>

Ändert das Service-Prüf-Zeitfenster (timeperiod) für einen bestimmten Service auf das, was durch die "timeperiod"-Option angegeben wird. Diese Option gibt den Kurznamen des Zeitfensters an, das als Prüf-Zeitfenster für den Service benutzt werden soll. Das Zeitfenster muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_SVC_EVENT_HANDLER

CHANGE_SVC_EVENT_HANDLER ;<host_name>;<service_description>;<event_handler_command>

Ändert den Eventhandler-Befehl für einen bestimmten Service auf den Befehl, der über "event_handler_command" angegeben wird. Diese Option gibt den Kurznamen des Befehls an, der als neuer Eventhandler-Befehl für diesen Service benutzt werden soll. Der Befehl muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

CHANGE_SVC_MODATTR

CHANGE_SVC_MODATTR ;<host_name>;<service_description>;<value>

Dieser Befehl ändert den Wert für "modified attributes" für den angegebenen Service. "Modified attributes"-Werte werden von Icinga benutzt, um festzulegen, welche Objekteigenschaften über Programmneustarts hinweg aufbewahrt werden sollen. Das bedeutet, dass die Änderung des Wertes die Datenaufbewahrung beeinflussen kann. Dies ist eine fortgeschrittene Option und sollte nur von Personen benutzt werden, die fundierte Kenntnisse von der Aufbewahrungslogik in Icinga besitzen.

CHANGE_SVC_NOTIFICATION_TIMEPERIOD

CHANGE_SVC_NOTIFICATION_TIMEPERIOD ;<host_name>;<service_description>;<notification_timeperiod>

Ändert das Service-Benachrichtigungs-Zeitfenster (timeperiod) für einen bestimmten Host auf das, was durch die "notification_timeperiod"-Option angegeben wird. Diese Option gibt den Kurznamen des Zeitfensters an, das als Benachrichtigungs-Zeitfenster für den Host benutzt werden soll. Das Zeitfenster muss bereits existieren, also beim letzten (Neu-)Start des Icinga-Prozesses konfiguriert gewesen sein.

DEL_ALL_HOST_COMMENTS

DEL_ALL_HOST_COMMENTS ;<host_name>

Löscht alle Kommentare, die mit einem bestimmten Host verbunden sind.

DEL_ALL_SVC_COMMENTS

DEL_ALL_SVC_COMMENTS ;<host_name>;<service_description>

Löscht alle Kommentare, die mit einem bestimmten Service verbunden sind.

DEL_HOST_COMMENT

DEL_HOST_COMMENT ;<comment_id>

Löscht einen Host-Kommentar. Die ID-Nummer des zu löschen Kommentars muss angegeben werden.

DEL_DOWNTIME_BY_HOST_NAME

DEL_DOWNTIME_BY_HOST_NAME ; <host_name>

Löscht die Ausfallzeiten des Hosts, der über "host_name" angegeben wurde.



Anmerkung

Dieser Befehl ist ab Icinga 1.4 verfügbar. Die Änderungen stammen vom [Opsview](#)-Team.

DEL_DOWNTIME_BY_HOSTGROUP_NAME

DEL_DOWNTIME_BY_HOSTGROUP_NAME ; <hostgroup_name>

Löscht die Ausfallzeiten aller Hosts der Hostgruppe, die über "hostgroup_name" angegeben wurde.



Anmerkung

Dieser Befehl ist ab Icinga 1.4 verfügbar. Die Änderungen stammen vom [Opsview](#)-Team.

DEL_DOWNTIME_BY_START_TIME_COMMENT

DEL_DOWNTIME_BY_START_TIME_COMMENT ; <start time[;comment_id]>

Löscht die Ausfallzeiten mit Startzeiten, die dem Zeitstempel entsprechen, der über "start time" angegeben wurde. Optional wird zusätzlich auch die Kommentar-ID überprüft.



Anmerkung

Dieser Befehl ist ab Icinga 1.4 verfügbar. Die Änderungen stammen vom [Opsview](#)-Team.

DEL_HOST_DOWNTIME

DEL_HOST_DOWNTIME ; <downtime_id>

Löscht den Host-Ausfallzeit-Eintrag mit der durch "downtime_id" angegebenen Nummer. Falls die Ausfallzeit gerade aktiv ist, wird die Ausfallzeit des Hosts damit beendet (solange es keine weiteren aktiven überlappenden Ausfallzeiteinträge für diesen Host gibt).

DEL_SVC_COMMENT

DEL_SVC_COMMENT ; <comment_id>

Löscht einen Service-Kommentar. Die ID-Nummer des zu löschen Kommentars muss angegeben werden.

DEL_SVC_DOWNTIME

DEL_SVC_DOWNTIME ;<downtime_id>

Löscht den Service-Ausfallzeit-Eintrag mit der durch "downtime_id" angegebenen Nummer. Falls die Ausfallzeit gerade aktiv ist, wird die Ausfallzeit des Service damit beendet (solange es keine weiteren aktiven überlappenden Ausfallzeiteinträge für diesen Service gibt).

DELAY_HOST_NOTIFICATION

DELAY_HOST_NOTIFICATION ;<host_name>;<notification_time>

Verzögert die nächste Benachrichtigung für einen bestimmten Host bis zur Erreichung des Wertes, der durch "notification_time" angegeben wird. Der "notification_time"-Parameter wird im time_t-Format angegeben (Sekunden seit der UNIX-Epoche). Beachten Sie, dass dies nur dann zutrifft, wenn der Host im gleichen Problemzustand wie momentan bleibt. Wenn der Host-Zustand in einen anderen Status wechselt, wird ggf. eine neue Benachrichtigung versandt, bevor die Zeit erreicht ist, die Sie als "notification_time"-Argument angegeben haben.

DELAY_SVC_NOTIFICATION

DELAY_SVC_NOTIFICATION ;<host_name>;<service_description>;<notification_time>

Verzögert die nächste Benachrichtigung für einen bestimmten Service bis zur Erreichung des Wertes, der durch "notification_time" angegeben wird. Der "notification_time"-Parameter wird im time_t-Format angegeben (Sekunden seit der UNIX-Epoche). Beachten Sie, dass dies nur dann zutrifft, wenn der Service im gleichen Problemzustand wie momentan bleibt. Wenn der Service-Zustand in einen anderen Status wechselt, wird ggf. eine neue Benachrichtigung versandt, bevor die Zeit erreicht ist, die Sie als "notification_time"-Argument angegeben haben.

DISABLE_ALL_NOTIFICATIONS_BEYOND_HOST

DISABLE_ALL_NOTIFICATIONS_BEYOND_HOST ;<host_name>

Deaktiviert die Benachrichtigungen für alle Host und Services "jenseits" (d.h. auf alles "Child-"Hosts) des angegebenen Hosts. Die aktuelle Benachrichtigungseinstellung auf dem angegebenen Host ist davon nicht betroffen.

DISABLE_CONTACT_HOST_NOTIFICATIONS

DISABLE_CONTACT_HOST_NOTIFICATIONS ;<contact_name>

Deaktiviert Host-Benachrichtigungen für einen bestimmten Kontakt.

DISABLE_CONTACT_SVC_NOTIFICATIONS

DISABLE_CONTACT_SVC_NOTIFICATIONS ;<contact_name>

Deaktiviert Service-Benachrichtigungen für einen bestimmten Kontakt.

DISABLE_CONTACTGROUP_HOST_NOTIFICATIONS

DISABLE_CONTACTGROUP_HOST_NOTIFICATIONS ;<contactgroup_name>

Deaktiviert Host-Benachrichtigungen für alle Kontakte einer bestimmten Kontaktgruppe.

DISABLE_CONTACTGROUP_SVC_NOTIFICATIONS

DISABLE_CONTACTGROUP_SVC_NOTIFICATIONS ;<contactgroup_name>

Deaktiviert Service-Benachrichtigungen für alle Kontakte einer bestimmten Kontaktgruppe.

DISABLE_EVENT_HANDLERS

DISABLE_EVENT_HANDLERS

Deaktiviert Host- und Service-Eventhandler auf programmweiter Ebene.

DISABLE_FAILURE_PREDICTION

DISABLE_FAILURE_PREDICTION

Deaktiviert Fehlervorhersage auf programmweiter Ebene. Diese Funktion ist in Icinga (noch) nicht implementiert.

DISABLE_FLAP_DETECTION

DISABLE_FLAP_DETECTION

Deaktiviert Host- und Service-Flattererkennung auf programmweiter Ebene.

DISABLE_HOST_AND_CHILD_NOTIFICATIONS

DISABLE_HOST_AND_CHILD_NOTIFICATIONS ;<host_name>

Deaktiviert Benachrichtigungen für den angegebenen Host sowie die "Child"-Hosts des angegebenen Hosts.

DISABLE_HOST_CHECK

DISABLE_HOST_CHECK ;<host_name>

Deaktiviert (regelmäßig geplante) aktive Prüfungen des angegebenen Hosts. "On-demand"-Prüfungen finden weiterhin statt (?).

DISABLE_HOST_EVENT_HANDLER

DISABLE_HOST_EVENT_HANDLER ;<host_name>

Deaktiviert den Eventhandler für den angegebenen Host.

DISABLE_HOST_FLAP_DETECTION

DISABLE_HOST_FLAP_DETECTION ;<host_name>

Deaktiviert die Flattererkennung für den angegebenen Host.

DISABLE_HOST_FRESHNESS_CHECKS

DISABLE_HOST_FRESHNESS_CHECKS

Deaktiviert Frische-Prüfungen für alle Hosts auf programmweiter Ebene.

DISABLE_HOST_NOTIFICATIONS

DISABLE_HOST_NOTIFICATIONS ;<host_name>

Deaktiviert Benachrichtigungen für den angegebenen Host.

DISABLE_HOST_SVC_CHECKS

DISABLE_HOST_SVC_CHECKS ;<host_name>

Deaktiviert aktive Prüfungen für alle Services des angegebenen Hosts.

DISABLE_HOST_SVC_NOTIFICATIONS

DISABLE_HOST_SVC_NOTIFICATIONS ;<host_name>

Deaktiviert Benachrichtigungen für alle Services des angegebenen Hosts.

DISABLE_HOSTGROUP_HOST_CHECKS

DISABLE_HOSTGROUP_HOST_CHECKS ;<hostgroup_name>

Deaktiviert aktive Prüfungen für alle Hosts der angegebenen Hostgruppe.

DISABLE_HOSTGROUP_HOST_NOTIFICATIONS

DISABLE_HOSTGROUP_HOST_NOTIFICATIONS ;<hostgroup_name>

Deaktiviert Benachrichtigungen für alle Hosts der angegebenen Hostgruppe. Dies deaktiviert NICHT die Benachrichtigungen für die Services der betroffenen Hosts - siehe DISABLE_HOSTGROUP_SVC_NOTIFICATIONS.

DISABLE_HOSTGROUP_PASSIVE_HOST_CHECKS

DISABLE_HOSTGROUP_PASSIVE_HOST_CHECKS ;<hostgroup_name>

Deaktiviert passive Prüfungen für alle Hosts der angegebenen Hostgruppe.

DISABLE_HOSTGROUP_PASSIVE_SVC_CHECKS

DISABLE_HOSTGROUP_PASSIVE_SVC_CHECKS ;<hostgroup_name>

Deaktiviert passive Prüfungen für alle Services, die mit den Hosts der angegebenen Hostgruppe verbunden sind.

DISABLE_HOSTGROUP_SVC_CHECKS

DISABLE_HOSTGROUP_SVC_CHECKS ;<hostgroup_name>

Deaktiviert aktive Prüfungen für alle Services, die mit den Hosts der angegebenen Hostgruppe verbunden sind.

DISABLE_HOSTGROUP_SVC_NOTIFICATIONS

DISABLE_HOSTGROUP_SVC_NOTIFICATIONS ;<hostgroup_name>

Deaktiviert Benachrichtigungen für alle Services, die mit den Hosts der angegebenen Hostgruppe verbunden sind. Dies deaktiviert NICHT die Benachrichtigungen für die Hosts der angegebenen Hostgruppe - siehe DISABLE_HOSTGROUP_HOST_NOTIFICATIONS.

DISABLE_NOTIFICATIONS

DISABLE_NOTIFICATIONS

Deaktiviert Host- und Service-Benachrichtigungen auf programmweiter Ebene.

DISABLE_PASSIVE_HOST_CHECKS

DISABLE_PASSIVE_HOST_CHECKS ;<host_name>

Deaktiviert Annahme und Verarbeitung von passiven Host-Prüfungen für den angegebenen Host.

DISABLE_PASSIVE_SVC_CHECKS

DISABLE_PASSIVE_SVC_CHECKS ;<host_name>;<service_description>

Deaktiviert Annahme und Verarbeitung von passiven Service-Prüfungen für den angegebenen Service.

DISABLE_PERFORMANCE_DATA

DISABLE_PERFORMANCE_DATA

Deaktiviert die Verarbeitung von Host- und Service-Performance-Daten auf programmweiter Ebene.

DISABLE_SERVICE_FLAP_DETECTION

DISABLE_SERVICE_FLAP_DETECTION ;<host_name>;<service_description>

Deaktiviert Flattererkennung für den angegebenen Service.

DISABLE_SERVICE_FRESHNESS_CHECKS

DISABLE_SERVICE_FRESHNESS_CHECKS

Deaktiviert Frische-Prüfungen auf programmweiter Ebene.

DISABLE_SERVICEGROUP_HOST_CHECKS

DISABLE_SERVICEGROUP_HOST_CHECKS ;<servicegroup_name>

Deaktiviert aktive Prüfungen für alle Hosts, die mit Services aus der angegebenen Servicegruppe verbunden sind.

DISABLE_SERVICEGROUP_HOST_NOTIFICATIONS

DISABLE_SERVICEGROUP_HOST_NOTIFICATIONS ;<servicegroup_name>

Deaktiviert Benachrichtigungen für alle Hosts, die mit Services aus der angegebenen Servicegruppe verbunden sind.

DISABLE_SERVICEGROUP_PASSIVE_HOST_CHECKS

DISABLE_SERVICEGROUP_PASSIVE_HOST_CHECKS ;<servicegroup_name>

Deaktiviert Annahme und Verarbeitung von passiven Prüfungen für alle Hosts, die mit Services aus der angegebenen Servicegruppe verbunden sind.

DISABLE_SERVICEGROUP_PASSIVE_SVC_CHECKS

DISABLE_SERVICEGROUP_PASSIVE_SVC_CHECKS ; <servicegroup_name>

Deaktiviert Annahme und Verarbeitung von passiven Prüfungen für alle Services aus der angegebenen Servicegruppe.

DISABLE_SERVICEGROUP_SVC_CHECKS

DISABLE_SERVICEGROUP_SVC_CHECKS ; <servicegroup_name>

Deaktiviert aktive Prüfungen für alle Services aus der angegebenen Servicegruppe.

DISABLE_SERVICEGROUP_SVC_NOTIFICATIONS

DISABLE_SERVICEGROUP_SVC_NOTIFICATIONS ; <servicegroup_name>

Deaktiviert Benachrichtigungen für alle Services aus der angegebenen Servicegruppe.

DISABLE_SVC_CHECK

DISABLE_SVC_CHECK ; <host_name>;<service_description>

Deaktiviert aktive Prüfungen für den angegebenen Service.

DISABLE_SVC_EVENT_HANDLER

DISABLE_SVC_EVENT_HANDLER ; <host_name>;<service_description>

Deaktiviert den Eventhandler für den angegebenen Service.

DISABLE_SVC_FLAP_DETECTION

DISABLE_SVC_FLAP_DETECTION ; <host_name>;<service_description>

Deaktiviert die Flattererkennung für den angegebenen Service.

DISABLE_SVC_NOTIFICATIONS

DISABLE_SVC_NOTIFICATIONS ; <host_name>;<service_description>

Deaktiviert Benachrichtigungen für den angegebenen Service.

ENABLE_ALL_NOTIFICATIONS_BEYOND_HOST

ENABLE_ALL_NOTIFICATIONS_BEYOND_HOST ; <host_name>

Aktiviert die Benachrichtigungen für alle Host und Services "jenseits" (d.h. auf alles "Child-"Hosts) des angegebenen Hosts. Die aktuelle Benachrichtigungseinstellung auf dem angegebenen Host ist davon nicht betroffen. Benachrichtigungen für diese Hosts und Services werden nur versandt, wenn Benachrichtigungen auch auf programmweiter Ebene aktiviert sind.

ENABLE_CONTACT_HOST_NOTIFICATIONS

ENABLE_CONTACT_HOST_NOTIFICATIONS ;<contact_name>

Aktiviert Host-Benachrichtigungen für einen bestimmten Kontakt..

ENABLE_CONTACT_SVC_NOTIFICATIONS

ENABLE_CONTACT_SVC_NOTIFICATIONS ;<contact_name>

Aktiviert Service-Benachrichtigungen für einen bestimmten Kontakt.

ENABLE_CONTACTGROUP_HOST_NOTIFICATIONS

ENABLE_CONTACTGROUP_HOST_NOTIFICATIONS ;<contactgroup_name>

Aktiviert Host-Benachrichtigungen für alle Kontakte der angegebenen Kontaktgruppe.

ENABLE_CONTACTGROUP_SVC_NOTIFICATIONS

ENABLE_CONTACTGROUP_SVC_NOTIFICATIONS ;<contactgroup_name>

Aktiviert Service-Benachrichtigungen für alle Kontakte der angegebenen Kontaktgruppe.

ENABLE_EVENT_HANDLERS

ENABLE_EVENT_HANDLERS

Aktiviert Host- und Service-Eventhandler auf programmweiter Ebene.

ENABLE_FAILURE_PREDICTION

ENABLE_FAILURE_PREDICTION

Aktiviert Fehlervorhersage auf programmweiter Ebene. Diese Funktion ist in Icinga (noch) nicht implementiert.

ENABLE_FLAP_DETECTION

ENABLE_FLAP_DETECTION

Aktiviert Host- und Service-Flattererkennung auf programmweiter Ebene.

ENABLE_HOST_AND_CHILD_NOTIFICATIONS

ENABLE_HOST_AND_CHILD_NOTIFICATIONS ;<host_name>

Aktiviert Benachrichtigungen für den angegebenen Host sowie die "Child"-Hosts des angegebenen Hosts. Benachrichtigungen für diese Hosts werden nur versandt, wenn Benachrichtigungen auch auf programmweiter Ebene aktiviert sind.

ENABLE_HOST_CHECK

ENABLE_HOST_CHECK ;<host_name>

Aktiviert (regelmäßig geplante) aktive Prüfungen des angegebenen Hosts.

ENABLE_HOST_EVENT_HANDLER

ENABLE_HOST_EVENT_HANDLER ;<host_name>

Aktiviert den Eventhandler für den angegebenen Host.

ENABLE_HOST_FLAP_DETECTION

ENABLE_HOST_FLAP_DETECTION ;<host_name>

Aktiviert die Flattererkennung für den angegebenen Host. Die Flattererkennung muss auch auf programmweiter Ebene aktiviert sein.

ENABLE_HOST_FRESHNESS_CHECKS

ENABLE_HOST_FRESHNESS_CHECKS

Aktiviert Frische-Prüfungen für alle Hosts auf programmweiter Ebene. Einzelne Hosts, bei denen die Frische-Prüfung deaktiviert ist, sind davon nicht betroffen. Sie werden weiterhin nicht auf Frische geprüft.

ENABLE_HOST_NOTIFICATIONS

ENABLE_HOST_NOTIFICATIONS ;<host_name>

Aktiviert Benachrichtigungen für den angegebenen Host. Benachrichtigungen für diesen Host werden nur versandt, wenn Benachrichtigungen auch auf programmweiter Ebene aktiviert sind.

ENABLE_HOST_SVC_CHECKS

ENABLE_HOST_SVC_CHECKS ;<host_name>

Aktiviert aktive Prüfungen für alle Services des angegebenen Hosts.

ENABLE_HOST_SVC_NOTIFICATIONS

ENABLE_HOST_SVC_NOTIFICATIONS ;<host_name>

Aktiviert Benachrichtigungen für alle Services des angegebenen Hosts. Benachrichtigungen werden nur versandt, wenn Benachrichtigungen auch auf programmweiter Ebene aktiviert sind.

ENABLE_HOSTGROUP_HOST_CHECKS

ENABLE_HOSTGROUP_HOST_CHECKS ;<hostgroup_name>

Aktiviert aktive Prüfungen für alle Hosts der angegebenen Hostgruppe.

ENABLE_HOSTGROUP_HOST_NOTIFICATIONS

ENABLE_HOSTGROUP_HOST_NOTIFICATIONS ;<hostgroup_name>

Aktiviert Benachrichtigungen für alle Hosts der angegebenen Hostgruppe. Dies aktiviert nicht die Benachrichtigungen für die Services, die mit den Hosts der angegebenen Hostgruppe verbunden sind. - siehe ENABLE_HOSTGROUP_SVC_NOTIFICATIONS. Benachrichtigungen für diese Hosts werden nur versandt, wenn Benachrichtigungen auch auf programmweiter Ebene aktiviert sind.

ENABLE_HOSTGROUP_PASSIVE_HOST_CHECKS

`ENABLE_HOSTGROUP_PASSIVE_HOST_CHECKS ; <hostgroup_name>`

Aktiviert passive Prüfungen für alle Hosts der angegebenen Hostgruppe.

ENABLE_HOSTGROUP_PASSIVE_SVC_CHECKS

`ENABLE_HOSTGROUP_PASSIVE_SVC_CHECKS ; <hostgroup_name>`

Aktiviert passive Prüfungen für alle Host der angegebenen Hostgruppe.

ENABLE_HOSTGROUP_SVC_CHECKS

`ENABLE_HOSTGROUP_SVC_CHECKS ; <hostgroup_name>`

Aktiviert aktive Prüfungen für alle Services, die mit den Hosts der angegebenen Hostgruppe verbunden sind.

ENABLE_HOSTGROUP_SVC_NOTIFICATIONS

`ENABLE_HOSTGROUP_SVC_NOTIFICATIONS ; <hostgroup_name>`

Aktiviert Benachrichtigungen für alle Services, die mit den Hosts der angegebenen Hostgruppe verbunden sind. Dies aktiviert nicht die Benachrichtigungen für die Hosts der angegebenen Hostgruppe. - siehe `ENABLE_HOSTGROUP_HOST_NOTIFICATIONS`. Benachrichtigungen für diese Services werden nur versandt, wenn Benachrichtigungen auch auf programmweiter Ebene aktiviert sind

ENABLE_NOTIFICATIONS

`ENABLE_NOTIFICATIONS`

Aktiviert Host- und Service-Benachrichtigungen auf programmweiter Ebene.

ENABLE_PASSIVE_HOST_CHECKS

`ENABLE_PASSIVE_HOST_CHECKS ; <host_name>`

Aktiviert Annahme und Verarbeitung von passiven Host-Prüfungen für den angegebenen Hosts.

ENABLE_PASSIVE_SVC_CHECKS

`ENABLE_PASSIVE_SVC_CHECKS ; <host_name>;<service_description>`

Aktiviert passive Prüfungen für den angegebenen Service.

ENABLE_PERFORMANCE_DATA

`ENABLE_PERFORMANCE_DATA`

Aktiviert die Verarbeitung von Host- und Service-Performance-Daten auf programmweiter Ebene.

ENABLE_SERVICE_FRESHNESS_CHECKS

ENABLE_SERVICE_FRESHNESS_CHECKS

Aktiviert Frische-Prüfungen für alle Services auf programmweiter Ebene. Einzelne Services, bei denen die Frische-Prüfung deaktiviert ist, sind davon nicht betroffen. Sie werden weiterhin nicht auf Frische geprüft.

ENABLE_SERVICEGROUP_HOST_CHECKS

ENABLE_SERVICEGROUP_HOST_CHECKS ;<servicegroup_name>

Aktiviert aktive Prüfungen für alle Hosts, die mit Services aus der angegebenen Servicegruppe verbunden sind.

ENABLE_SERVICEGROUP_HOST_NOTIFICATIONS

ENABLE_SERVICEGROUP_HOST_NOTIFICATIONS ;<servicegroup_name>

Aktiviert Benachrichtigungen für alle Hosts, die mit Services aus der angegebenen Servicegruppe verbunden sind. Benachrichtigungen für diese Hosts werden nur versandt, wenn Benachrichtigungen auch auf programmweiter Ebene aktiviert sind.

ENABLE_SERVICEGROUP_PASSIVE_HOST_CHECKS

ENABLE_SERVICEGROUP_PASSIVE_HOST_CHECKS ;<servicegroup_name>

Aktiviert Annahme und Verarbeitung von passiven Prüfungen für alle Hosts, die mit Services aus der angegebenen Servicegruppe verbunden sind.

ENABLE_SERVICEGROUP_PASSIVE_SVC_CHECKS

ENABLE_SERVICEGROUP_PASSIVE_SVC_CHECKS ;<servicegroup_name>

Aktiviert Annahme und Verarbeitung von passiven Prüfungen für alle Services aus der angegebenen Servicegruppe.

ENABLE_SERVICEGROUP_SVC_CHECKS

ENABLE_SERVICEGROUP_SVC_CHECKS ;<servicegroup_name>

Aktiviert aktive Prüfungen für alle Services aus der angegebenen Servicegruppe.

ENABLE_SERVICEGROUP_SVC_NOTIFICATIONS

ENABLE_SERVICEGROUP_SVC_NOTIFICATIONS ;<servicegroup_name>

Aktiviert Benachrichtigungen für alle Services aus der angegebenen Servicegruppe. Benachrichtigungen für diese Service werden nur versandt, wenn Benachrichtigungen auch auf programmweiter Ebene aktiviert sind.

ENABLE_SVC_CHECK

ENABLE_SVC_CHECK ;<host_name>;<service_description>

Aktiviert aktive Prüfungen für den angegebenen Service.

ENABLE_SVC_EVENT_HANDLER

ENABLE_SVC_EVENT_HANDLER ;<host_name>;<service_description>

Aktiviert den Eventhandler für den angegebenen Service.

ENABLE_SVC_FLAP_DETECTION

ENABLE_SVC_FLAP_DETECTION ;<host_name>;<service_description>

Aktiviert die Flattererkennung für den angegebenen Service. Die Flattererkennung muss auch auf programmweiter Ebene aktiviert sind.

ENABLE_SVC_NOTIFICATIONS

ENABLE_SVC_NOTIFICATIONS ;<host_name>;<service_description>

Aktiviert Benachrichtigungen für den angegebenen Service. Benachrichtigungen für diesen Service werden nur versandt, wenn Benachrichtigungen auch auf programmweiter Ebene aktiviert sind.

PROCESS_FILE

PROCESS_FILE ;<file_name>;<delete>

Weist Icinga an, alle externen Befehle zu verarbeiten, die in der Datei zu finden sind, die mit dem <file_name>-Argument angegeben wird. Falls die <delete>-Option nicht-Null ist, wird die Datei nach der Verarbeitung gelöscht. Falls die <delete>-Option Null (0) ist, bleibt die Datei erhalten.

PROCESS_HOST_CHECK_RESULT

PROCESS_HOST_CHECK_RESULT ;<host_name>;<status_code>;<plugin_output>

Dies wird benutzt, um ein passives Prüfergebnis für einen bestimmten Host einzuliefern. Der "status_code" gibt den Status der Host-Prüfung an und sollte einen der folgenden Werte haben: 0=UP, 1=DOWN, 2=UNREACHABLE. Das "plugin_output"-Argument enthält den Text der Host-Prüfung, zusammen mit optionalen Performance-Daten.

PROCESS_SERVICE_CHECK_RESULT

PROCESS_SERVICE_CHECK_RESULT ;<host_name>;<service_description>;<return_code>;<plugin_output>

Dies wird benutzt, um ein passives Prüfergebnis für einen bestimmten Service einzuliefern. Der "status_code" gibt den Status der Service-Prüfung an und sollte einen der folgenden Werte haben: 0=OK, 1=WARNING, 2=CRITICAL, 3=UNKNOWN. Das "plugin_output"-Argument enthält den Text der Service-Prüfungen, zusammen mit optionalen Performance-Daten.

READ_STATE_INFORMATION

READ_STATE_INFORMATION

Bewirkt, dass Icinga alle aktuellen Überwachungs-Statusinformationen aus der Statusaufbewahrungsdatei (state retention file) liest. Normalerweise werden diese Informationen beim Start des Icinga-Prozesses vor dem Start der eigentlichen Überwachung geladen. Dieser Befehl bewirkt, dass Icinga die aktuellen Überwachungs-Statusinformationen verwirft und stattdessen die Informationen aus der Statusaufbewahrungsdatei liest. *Vorsicht!*

REMOVE_HOST_ACKNOWLEDGEMENT

```
REMOVE_HOST_ACKNOWLEDGEMENT ;<host_name>
```

Dies entfernt die Problem-Bestätigung für den angegebenen Host. Sobald die Bestätigung entfernt ist, können wieder Benachrichtigungen für den Host versandt werden.

REMOVE_SVC_ACKNOWLEDGEMENT

```
REMOVE_SVC_ACKNOWLEDGEMENT ;<host_name>;<service_description>
```

Dies entfernt die Problem-Bestätigung für den angegebenen Service. Sobald die Bestätigung entfernt ist, können wieder Benachrichtigungen für den Service versandt werden.

RESTART_PROGRAM

```
RESTART_PROGRAM
```

Veranlasst einen Restart des Icinga-Prozesses.

SAVE_STATE_INFORMATION

```
SAVE_STATE_INFORMATION
```

Bewirkt, dass Icinga alle aktuellen Überwachungs-Statusinformationen in die Statusaufbewahrungsdatei (state retention file) schreibt. Normalerweise werden diese Informationen vor Beendigung des Icinga-Prozesses (und möglicherweise in bestimmten regelmäßig geplanten Intervallen) geschrieben. Dieser Befehl erlaubt Ihnen, dass Icinga diese Informationen sofort in die Datei schreibt. Dies beeinflusst nicht die aktuellen Statusinformationen des Icinga-Prozesses.

SCHEDULE_AND_PROPAGATE_HOST_DOWNTIME

```
SCHEDULE_AND_PROPAGATE_HOST_DOWNTIME ;<host_name>;<start_time>;<end_time>;<fixed>;<trigger_id>;<duration>;<author>;<comment>
```

Plant eine Ausfallzeit für den angegebenen Host und seine "Kinder"(-Hosts). Wenn das "fixed"-Argument auf eins (1) gesetzt ist, wird die Ausfallzeit zu den durch die "start"- und "end"-Argumente angegebenen Zeiten starten bzw. enden. Andernfalls wird die Ausfallzeit zwischen den Start- und Endzeiten beginnen und die Anzahl von Sekunden dauern, die durch das "duration"-Argument angegeben wird. Die "start"- und "end"-Argumente werden im time_t-Format (Sekunden seit der UNIX-Epoche) angegeben. Die Ausfallzeit des angegebenen (Eltern-) Hosts kann durch einen anderen Ausfallzeiteintrag ausgelöst werden, wenn die "trigger_id" den Wert der ID eines anderen Ausfallzeiteintrag hat. Setzen Sie den Wert von "trigger_id" auf Null (0), wenn die Ausfallzeit des angegebenen (Eltern-) Hosts nicht von einem anderen Ausfallzeiteintrag ausgelöst werden soll.

SCHEDULE_AND_PROPAGATE_TRIGGERED_HOST_DOWNTIME

```
SCHEDULE_AND_PROPAGATE_TRIGGERED_HOST_DOWNTIME ;<host_name>;<start_time>;<end_time>;<fixed>;<trigger_id>;<duration>;<author>;<comment>
```

Plant eine Ausfallzeit für den angegebenen Host und seine "Kinder"(-Hosts). Wenn das "fixed"-Argument auf eins (1) gesetzt ist, wird die Ausfallzeit zu den durch die "start"- und "end"-Argumente angegebenen Zeiten starten bzw. enden. Andernfalls wird die Ausfallzeit zwischen den Start- und Endzeiten beginnen und die Anzahl von Sekunden dauern, die durch das "duration"-Argument angegeben wird. Die "start"- und "end"-Argumente werden im time_t-Format (Sekunden seit der UNIX-Epoche) angegeben. Die Ausfallzeit der "Kinder"-Hosts wird durch die Ausfallzeit des "Eltern"-Hosts ausgelöst. Die Ausfallzeit des angegebenen (Eltern-) Hosts kann durch einen anderen Ausfallzeiteintrag ausgelöst werden, wenn die

"trigger_id" den Wert der ID eines anderen Ausfallzeiteintrag hat. Setzen Sie den Wert von "trigger_id" auf Null (0), wenn die Ausfallzeit des angegebenen (Eltern-) Hosts nicht von einem anderen Ausfallzeiteintrag ausgelöst werden soll.

SCHEDULE_FORCED_HOST_CHECK

SCHEDULE_FORCED_HOST_CHECK ;<host_name>;<check_time>

Plant eine erzwungene aktive Prüfung des angegebenen Hosts zur "check_time". Das "check_time"-Argument gibt dem Wert im time_t-Format (Sekunden seit der UNIX-Epoche) an. Erzwungene Prüfungen in jedem Fall ausgeführt, also unabhängig von der Zeit (z.B. werden Zeitfenster-Einschränkungen ignoriert) und davon, ob aktive Prüfungen für bestimmte Hosts oder auf programmweiter Ebene aktiviert sind oder nicht.

SCHEDULE_FORCED_HOST_SVC_CHECKS

SCHEDULE_FORCED_HOST_SVC_CHECKS ;<host_name>;<check_time>

Plant eine erzwungene aktive Prüfung von allen Services des angegebenen Hosts zur "check_time". Das "check_time"-Argument gibt dem Wert im time_t-Format (Sekunden seit der UNIX-Epoche) an. Erzwungene Prüfungen in jedem Fall ausgeführt, also unabhängig von der Zeit (z.B. werden Zeitfenster-Einschränkungen ignoriert) und davon, ob aktive Prüfungen für bestimmte Hosts oder auf programmweiter Ebene aktiviert sind oder nicht.

SCHEDULE_FORCED_SVC_CHECK

SCHEDULE_FORCED_SVC_CHECK ;<host_name>;<service_description>;<check_time>

Plant eine erzwungene aktive Prüfung eines bestimmten Service des angegebenen Hosts zur "check_time". Das "check_time"-Argument gibt dem Wert im time_t-Format (Sekunden seit der UNIX-Epoche) an. Erzwungene Prüfungen in jedem Fall ausgeführt, also unabhängig von der Zeit (z.B. werden Zeitfenster-Einschränkungen ignoriert) und davon, ob aktive Prüfungen für bestimmte Hosts oder auf programmweiter Ebene aktiviert sind oder nicht.

SCHEDULE_HOST_CHECK

SCHEDULE_HOST_CHECK ;<host_name>;<check_time>

Plant die nächste aktive Prüfung des angegebenen Hosts zur "check_time". Das "check_time"-Argument gibt dem Wert im time_t-Format (Sekunden seit der UNIX-Epoche) an. Beachten Sie, dass der Host ggf. nicht zu der Zeit geprüft wird, die Sie angegeben haben. Das kann eine Reihe von Gründen haben: Aktive Prüfungen sind auf Host- oder programmweiter Ebene deaktiviert, die Host-Prüfung ist bereits für einen früheren Zeitpunkt geplant, usw. Wenn Sie die Host-Prüfung erzwingen wollen, dann schauen Sie beim SCHEDULE_FORCED_HOST_CHECK-Befehl.

SCHEDULE_HOST_DOWNTIME

SCHEDULE_HOST_DOWNTIME ;<host_name>;<start_time>;<end_time>;<fixed>;<trigger_id>;<duration>;<author>;<comment>

Plant eine Ausfallzeit für den angegebenen Host. Wenn das "fixed"-Argument auf eins (1) gesetzt ist, wird die Ausfallzeit zu den durch die "start"- und "end"-Argumente angegebenen Zeiten starten bzw. enden. Andernfalls wird die Ausfallzeit zwischen den Start- und Endzeiten beginnen und die Anzahl von Sekunden dauern, die durch das "duration"-Argument angegeben wird. Die "start"- und "end"-Argumente werden im time_t-Format (Sekunden seit der UNIX-Epoche) angegeben. Die Ausfallzeit des angegebenen Hosts kann durch einen anderen Ausfallzeiteintrag ausgelöst werden, wenn die "trigger_id" den Wert der ID eines

anderen Ausfallzeiteintrag hat. Setzen Sie den Wert von "trigger_id" auf Null (0), wenn die Ausfallzeit des angegebenen Hosts nicht von einem anderen Ausfallzeiteintrag ausgelöst werden soll.

SCHEDULE_HOST_SVC_CHECKS

SCHEDULE_HOST_SVC_CHECKS ;<host_name>;<check_time>

Plant die nächste aktive Prüfung aller Services des angegebenen Hosts zur "check_time". Das "check_time"-Argument gibt dem Wert im time_t-Format (Sekunden seit der UNIX-Epoche) an. Beachten Sie, dass die Services ggf. nicht zu der Zeit geprüft wird, die Sie angegeben haben. Das kann eine Reihe von Gründen haben: Aktive Prüfungen sind auf Service- oder programmweiter Ebene deaktiviert, die Service-Prüfungen sind bereits für einen früheren Zeitpunkt geplant, usw. Wenn Sie die Service-Prüfungen erzwingen wollen, dann schauen Sie beim SCHEDULE_FORCED_HOST_SVC_CHECKS-Befehl.

SCHEDULE_HOST_SVC_DOWNTIME

SCHEDULE_HOST_SVC_DOWNTIME ;<host_name>;<start_time>;<end_time>;<fixed>;<trigger_id>;<duration>;<author>;<comment>

Plant eine Ausfallzeit für alle Services des angegebenen Hosts. Wenn das "fixed"-Argument auf eins (1) gesetzt ist, wird die Ausfallzeit zu den durch die "start"- und "end"-Argumente angegebenen Zeiten starten bzw. enden. Andernfalls wird die Ausfallzeit zwischen den Start- und Endzeiten beginnen und die Anzahl von Sekunden dauern, die durch das "duration"-Argument angegeben wird. Die "start"- und "end"-Argumente werden im time_t-Format (Sekunden seit der UNIX-Epoche) angegeben. Die Ausfallzeit der Services kann durch einen anderen Ausfallzeiteintrag ausgelöst werden, wenn die "trigger_id" den Wert der ID eines anderen Ausfallzeiteintrag hat. Setzen Sie den Wert von "trigger_id" auf Null (0), wenn die Ausfallzeit der Services nicht von einem anderen Ausfallzeiteintrag ausgelöst werden soll.

SCHEDULE_HOSTGROUP_HOST_DOWNTIME

SCHEDULE_HOSTGROUP_HOST_DOWNTIME ;<hostgroup_name>;<start_time>;<end_time>;<fixed>;<trigger_id>;<duration>;<author>;<comment>

Plant eine Ausfallzeit für alle Hosts der angegebenen Hostgruppe. Wenn das "fixed"-Argument auf eins (1) gesetzt ist, wird die Ausfallzeit zu den durch die "start"- und "end"-Argumente angegebenen Zeiten starten bzw. enden. Andernfalls wird die Ausfallzeit zwischen den Start- und Endzeiten beginnen und die Anzahl von Sekunden dauern, die durch das "duration"-Argument angegeben wird. Die "start"- und "end"-Argumente werden im time_t-Format (Sekunden seit der UNIX-Epoche) angegeben. Die Ausfallzeit der Hosts kann durch einen anderen Ausfallzeiteintrag ausgelöst werden, wenn die "trigger_id" den Wert der ID eines anderen Ausfallzeiteintrag hat. Setzen Sie den Wert von "trigger_id" auf Null (0), wenn die Ausfallzeit der Hosts nicht von einem anderen Ausfallzeiteintrag ausgelöst werden soll.

SCHEDULE_HOSTGROUP_SVC_DOWNTIME

SCHEDULE_HOSTGROUP_SVC_DOWNTIME ;<hostgroup_name>;<start_time>;<end_time>;<fixed>;<trigger_id>;<duration>;<author>;<comment>

Plant eine Ausfallzeit für alle Services der angegebenen Hostgruppe. Wenn das "fixed"-Argument auf eins (1) gesetzt ist, wird die Ausfallzeit zu den durch die "start"- und "end"-Argumente angegebenen Zeiten starten bzw. enden. Andernfalls wird die Ausfallzeit zwischen den Start- und Endzeiten beginnen und die Anzahl von Sekunden dauern, die durch das "duration"-Argument angegeben wird. Die "start"- und "end"-Argumente werden im time_t-Format (Sekunden seit der UNIX-Epoche) angegeben. Die Ausfallzeit der Services kann durch einen anderen Ausfallzeiteintrag ausgelöst werden, wenn die "trigger_id" den Wert der ID eines anderen Ausfallzeiteintrag hat. Setzen Sie den Wert von "trigger_id" auf Null (0), wenn die Ausfallzeit der Services nicht von einem anderen Ausfallzeiteintrag ausgelöst werden soll.

SCHEDULE_SERVICEGROUP_HOST_DOWNTIME

`SCHEDULE_SERVICEGROUP_HOST_DOWNTIME ;<servicegroup_name>;<start_time>;<end_time>;<fixed>;<trigger_id>;<duration>;<author>;<comment>`

Plant eine Ausfallzeit für alle Hosts, die Services in der angegebenen Servicegruppe haben. Wenn das "fixed"-Argument auf eins (1) gesetzt ist, wird die Ausfallzeit zu den durch die "start"- und "end"-Argumente angegebenen Zeiten starten bzw. enden. Andernfalls wird die Ausfallzeit zwischen den Start- und Endzeiten beginnen und die Anzahl von Sekunden dauern, die durch das "duration"-Argument angegeben wird. Die "start"- und "end"-Argumente werden im time_t-Format (Sekunden seit der UNIX-Epoche) angegeben. Die Ausfallzeit der Hosts kann durch einen anderen Ausfallzeiteintrag ausgelöst werden, wenn die "trigger_id" den Wert der ID eines anderen Ausfallzeiteintrag hat. Setzen Sie den Wert von "trigger_id" auf Null (0), wenn die Ausfallzeit der Hosts nicht von einem anderen Ausfallzeiteintrag ausgelöst werden soll.

SCHEDULE_SERVICEGROUP_SVC_DOWNTIME

`SCHEDULE_SERVICEGROUP_SVC_DOWNTIME ;<servicegroup_name>;<start_time>;<end_time>;<fixed>;<trigger_id>;<duration>;<author>;<comment>`

Plant eine Ausfallzeit für alle Services der angegebenen Servicegruppe. Wenn das "fixed"-Argument auf eins (1) gesetzt ist, wird die Ausfallzeit zu den durch die "start"- und "end"-Argumente angegebenen Zeiten starten bzw. enden. Andernfalls wird die Ausfallzeit zwischen den Start- und Endzeiten beginnen und die Anzahl von Sekunden dauern, die durch das "duration"-Argument angegeben wird. Die "start"- und "end"-Argumente werden im time_t-Format (Sekunden seit der UNIX-Epoche) angegeben. Die Ausfallzeit der Services kann durch einen anderen Ausfallzeiteintrag ausgelöst werden, wenn die "trigger_id" den Wert der ID eines anderen Ausfallzeiteintrag hat. Setzen Sie den Wert von "trigger_id" auf Null (0), wenn die Ausfallzeit der Services nicht von einem anderen Ausfallzeiteintrag ausgelöst werden soll.

SCHEDULE_SVC_CHECK

`SCHEDULE_SVC_CHECK ;<host_name>;<service_description>;<check_time>`

Plant die nächste aktive Prüfung des angegebenen Service zur "check_time". Das "check_time"-Argument gibt dem Wert im time_t-Format (Sekunden seit der UNIX-Epoche) an. Beachten Sie, dass der Service ggf. nicht zu der Zeit geprüft wird, die Sie angegeben haben. Das kann eine Reihe von Gründen haben: Aktive Prüfungen sind auf Service- oder programmweiter Ebene deaktiviert, die Service-Prüfung ist bereits für einen früheren Zeitpunkt geplant, usw. Wenn Sie die Service-Prüfung erzwingen wollen, dann schauen Sie beim SCHEDULE_FORCED_SVC_CHECK-Befehl.

SCHEDULE_SVC_DOWNTIME

`SCHEDULE_SVC_DOWNTIME ;<host_name>;<service_desription>;<start_time>;<end_time>;<fixed>;<trigger_id>;<duration>;<author>;<comment>`

Plant eine Ausfallzeit für den angegebenen Service. Wenn das "fixed"-Argument auf eins (1) gesetzt ist, wird die Ausfallzeit zu den durch die "start"- und "end"-Argumente angegebenen Zeiten starten bzw. enden. Andernfalls wird die Ausfallzeit zwischen den Start- und Endzeiten beginnen und die Anzahl von Sekunden dauern, die durch das "duration"-Argument angegeben wird. Die "start"- und "end"-Argumente werden im time_t-Format (Sekunden seit der UNIX-Epoche) angegeben. Die Ausfallzeit des angegebenen Service kann durch einen anderen Ausfallzeiteintrag ausgelöst werden, wenn die "trigger_id" den Wert der ID eines anderen Ausfallzeiteintrag hat. Setzen Sie den Wert von "trigger_id" auf Null (0), wenn die Ausfallzeit des angegebenen Service nicht von einem anderen Ausfallzeiteintrag ausgelöst werden soll.

SEND_CUSTOM_HOST_NOTIFICATION

SEND_CUSTOM_HOST_NOTIFICATION ;<host_name>;<options>;<author>;<comment>

Erlaubt Ihnen den Versand einer angepassten Host-Benachrichtigung. Das ist sehr nützlich in schlimmen Situationen, Notfällen oder um mit allen Admins zu kommunizieren, die für einen bestimmten Host zuständig sind. Beim Versand der Host-Benachrichtigung wird das \$NOTIFICATIONTYPE\$-Makro auf "CUSTOM" gesetzt. Das <options>-Feld ist ein logisches ODER der folgenden Ganzzahl-Werte, die beeinflussen, wie die Benachrichtigung versandt wird: 0 = keine Option (Default), 1 = Broadcast (Benachrichtigungen an alle normalen und alle Eskalationskontakte des Hosts versenden), 2 = erzwungen (Benachrichtigung wird versandt, unabhängig von der aktuellen Zeit, ob Benachrichtigungen aktiviert sind oder nicht, etc.), 4 = inkrementieren der Benachrichtigungsnummer für den Host (dies wird nicht automatisch bei angepassten Benachrichtigungen gemacht). Das Kommentarfeld kann über das \$NOTIFICATIONCOMMENT\$-Makro in Benachrichtigungsbefehlen benutzt werden.

SEND_CUSTOM_SVC_NOTIFICATION

SEND_CUSTOM_SVC_NOTIFICATION ;<host_name>;<service_description>;<options>;<author>;<comment>

Erlaubt Ihnen den Versand einer angepassten Service-Benachrichtigung. Das ist sehr nützlich in schlimmen Situationen, Notfällen oder um mit allen Admins zu kommunizieren, die für einen bestimmten Service zuständig sind. Beim Versand der Service-Benachrichtigung wird das \$NOTIFICATIONTYPE\$-Makro auf "CUSTOM" gesetzt. Das <options>-Feld ist ein logisches ODER der folgenden Ganzzahl-Werte, die beeinflussen, wie die Benachrichtigung versandt wird: 0 = keine Option (Default), 1 = Broadcast (Benachrichtigungen an alle normalen und alle Eskalationskontakte des Service versenden), 2 = erzwungen (Benachrichtigung wird versandt, unabhängig von der aktuellen Zeit, ob Benachrichtigungen aktiviert sind oder nicht, etc.), 4 = inkrementieren der Benachrichtigungsnummer für den Service (dies wird nicht automatisch bei angepassten Benachrichtigungen gemacht). Das Kommentarfeld kann über das \$NOTIFICATIONCOMMENT\$-Makro in Benachrichtigungsbefehlen benutzt werden.

SET_HOST_NOTIFICATION_NUMBER

SET_HOST_NOTIFICATION_NUMBER ;<host_name>;<notification_number>

Setzt die aktuellen Benachrichtigungsnummer für den angegebenen Host. Ein Wert von Null (0) zeigt an, dass bisher keine Benachrichtigung für das aktuelle Host-Problem versandt wurde. Das ist nützlich zum Erzwingen einer Eskalation (basierend auf der Benachrichtigungsnummer) oder zur Replizierung von Benachrichtigungsinformationen in redundanten Überwachungsumgebungen. Benachrichtigungsnummern größer Null haben keinen spürbaren Einfluss auf den Benachrichtigungsprozess, falls sich der Host gerade in einem UP-Zustand befindet.

SET_SVC_NOTIFICATION_NUMBER

SET_SVC_NOTIFICATION_NUMBER ;<host_name>;<service_description>;<notification_number>

Setzt die aktuellen Benachrichtigungsnummer für den angegebenen Service. Ein Wert von Null (0) zeigt an, dass bisher keine Benachrichtigung für das aktuelle Service-Problem versandt wurde. Das ist nützlich zum Erzwingen einer Eskalation (basierend auf der Benachrichtigungsnummer) oder zur Replizierung von Benachrichtigungsinformationen in redundanten Überwachungsumgebungen. Benachrichtigungsnummern größer Null haben keinen spürbaren Einfluss auf den Benachrichtigungsprozess, falls sich der Service gerade in einem OK-Zustand befindet.

SHUTDOWN_PROGRAM

SHUTDOWN_PROGRAM

Stoppt den Icinga-Prozess.

START_ACCEPTING_PASSIVE_HOST_CHECKS

START_ACCEPTING_PASSIVE_HOST_CHECKS

Aktiviert Annahme und Verarbeitung von passiven Host-Prüfungen auf programmweiter Ebene.

START_ACCEPTING_PASSIVE_SVC_CHECKS

START_ACCEPTING_PASSIVE_SVC_CHECKS

Aktiviert Annahme und Verarbeitung von passiven Service-Prüfungen auf programmweiter Ebene.

START_EXECUTING_HOST_CHECKS

START_EXECUTING_HOST_CHECKS

Aktiviert aktive Host-Prüfungen auf programmweiter Ebene.

START_EXECUTING_SVC_CHECKS

START_EXECUTING_SVC_CHECKS

Aktiviert aktive Service-Prüfungen auf programmweiter Ebene.

START_OBSESSING_OVER_HOST

START_OBSESSING_OVER_HOST; <host_name>

Aktiviert die Verarbeitung von Host-Prüfungen mit Hilfe des OCHP-Befehls für den angegebenen Host.

START_OBSESSING_OVER_HOST_CHECKS

START_OBSESSING_OVER_HOST_CHECKS

Aktiviert die Verarbeitung von Host-Prüfungen mit Hilfe des OCHP-Befehls auf programmweiter Ebene.

START_OBSESSING_OVER_SVC

START_OBSESSING_OVER_SVC; <host_name>; <service_description>

Aktiviert die Verarbeitung von Service-Prüfungen mit Hilfe des OCSP-Befehls für den angegebenen Service.

START_OBSESSING_OVER_SVC_CHECKS

START_OBSESSING_OVER_SVC_CHECKS

Aktiviert die Verarbeitung von Service-Prüfungen mit Hilfe des OCSP-Befehls auf programmweiter Ebene.

STOP_ACCEPTING_PASSIVE_HOST_CHECKS

STOP_ACCEPTING_PASSIVE_HOST_CHECKS

Deaktiviert Annahme und Verarbeitung von passiven Host-Prüfungen auf programmweiter Ebene.

STOP_ACCEPTING_PASSIVE_SVC_CHECKS

STOP_ACCEPTING_PASSIVE_SVC_CHECKS

Deaktiviert Annahme und Verarbeitung von passiven Service-Prüfungen auf programmweiter Ebene.

STOP_EXECUTING_HOST_CHECKS

STOP_EXECUTING_HOST_CHECKS

Deaktiviert aktive Host-Prüfungen auf programmweiter Ebene.

STOP_EXECUTING_SVC_CHECKS

STOP_EXECUTING_SVC_CHECKS

Deaktiviert aktive Service-Prüfungen auf programmweiter Ebene.

STOP_OBSESSING_OVER_HOST

STOP_OBSESSING_OVER_HOST ; <host_name>

Deaktiviert die Verarbeitung von Host-Prüfungen mit Hilfe des OCHP-Befehls für den angegebenen Host.

STOP_OBSESSING_OVER_HOST_CHECKS

STOP_OBSESSING_OVER_HOST_CHECKS

Deaktiviert die Verarbeitung von Host-Prüfungen mit Hilfe des OCHP-Befehls auf programmweiter Ebene.

STOP_OBSESSING_OVER_SVC

STOP_OBSESSING_OVER_SVC ; <host_name> ; <service_description>

Deaktiviert die Verarbeitung von Service-Prüfungen mit Hilfe des OCSP-Befehls für den angegebenen Service.

STOP_OBSESSING_OVER_SVC_CHECKS

STOP_OBSESSING_OVER_SVC_CHECKS

Deaktiviert die Verarbeitung von Service-Prüfungen mit Hilfe des OCSP-Befehls auf programmweiter Ebene.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Entwickeln von Plugins für die
Nutzung mit Embedded Perl

[Zum Anfang](#)

Installation und Benutzung der
Icinga-API

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Installation und Benutzung der Icinga-API

[Zurück](#)
[Kapitel 11. Entwicklung](#)
[Weiter](#)

Installation und Benutzung der Icinga-API

Voraussetzungen

Sie benötigen installierte und lauffähige Versionen von Icinga und IDOUtils oder MKLiveStatus, um die Icinga-API nutzen zu können.



Anmerkung

Falls Sie Icinga noch nicht installiert haben, folgen Sie den Anweisungen in der [quickstart-idoutils](#)-Dokumentation.

Wenn Sie die IDOUtils-Datenbank als Datenquelle verwenden, installieren Sie bitte PHP-PDO.

- **RHEL/Fedora/CentOS**

Stellen Sie sicher, dass Sie ein Repository/Packages für PHP 5.2.x zur Verfügung haben - RHEL/CentOS (CentOS <= 5.4) unterstützen lediglich 5.1.6.

```
# yum install php-pdo php-mysql|pgsql
```

- **Debian/Ubuntu**

```
# apt-get install php5 php5-mysql|pgsql
```

- **openSuSE;**

Bitte benutzen Sie yast zur Installation der Pakete php5, php5-pdo und php5-mysql bzw. php5-pgsql.

Installation und Konfiguration

1. Software

Klonen Sie von icinga-api.git, um einen neuen Branch zu bekommen:

```
# git clone git://git.icinga.org/icinga-api.git
```

oder laden Sie die Software von

<https://git.icinga.org/index?p=icinga-api.git;a=snapshot;h=refs/heads/master;sf=tgz>.

2. Installation



Anmerkung

Die Icinga-API ist innerhalb des Pakets von Icinga Core, IDOUtils und Docs enthalten und wird während 'make install' mit installiert. Sofern Sie das schon durchgeführt haben, befindet sich die API standardmäßig in /usr/local/icinga/share/icinga-api/ und Sie können diesen Abschnitt überspringen.



Anmerkung

Wenn Sie die Icinga-API für das neue Icinga-Web benötigen und schon Icinga Core mit den IDOUtils installiert haben, können Sie diesen Guide verlassen und direkt [Icinga-Web](#) installieren.

Download

Sie können die Icinga-API direkt aus dem GIT Repository beziehen, für einen frischen Clone führen Sie folgenden Befehl aus:

```
# git clone git://git.icinga.org/icinga-api.git
```

Sofern Sie lediglich ein Update benötigen:

```
# cd icinga-api && git pull origin master
```

Oder laden Sie einen Snapshot direkt über das Gitweb:

<https://git.icinga.org/index?p=icinga-api.git;a=snapshot;h=refs/heads/master;sf=tgz>.

Installation

Entpacken Sie die Icinga-API, führen Sie configure aus und installieren Sie die Icinga-API

```
# tar xzvf icinga-api-(version).tar.gz
# ./configure
```

Sie können den Präfix definieren, wohin die Icinga-API installiert wird, sowie den Ort der Systemkonfiguration für Icinga Core und IDOUtils und die ausführenden Benutzer. All diese Informationen werden bei einer Installation durch das Core Paket direkt gesetzt.

```
# ./configure --datarootdir=/usr/local/icinga/share \
--sysconfdir=/usr/local/icinga/etc \
--with-command-user=icinga-cmd \
--with-command-group=icinga-cmd \
--with-icinga-user=icinga \
--with-icinga-group=icinga \
--with-web-user=www-data \
--with-web-group=www-data
```



Anmerkung

Die --with-web...-Direktiven müssen gesetzt sein. Andernfalls werden die Web-Logs nicht korrekt erstellt. Außerdem kann dies zu einem leeren Haupt-Cronk führen. Bitte beachten Sie, dass die Werte von 'user' und 'group' abhängig von der Distribution sind.

```
# make install
```

Konfiguration

Wenn Sie Ihr eigenes Addon auf Basis der Icinga-API entwickeln möchten, benötigen Sie das folgende assoziative Array:

```
$idoConfig = array (
    'type'          => '<Type of database>',
    'host'          => '<Database hostname>',
    'database'      => '<Databasename>',
    'user'          => '<Username>',
    'password'      => '<password>',
    'persistent'    => <true | false>,
    'table_prefix'  => '<table prefix>',
);
```

Beispiel:

```
$idoConfig = array (
    'type'          => 'mysql',
    'host'          => 'localhost',
    'database'      => 'ido',
    'user'          => 'idouser',
    'password'      => 'idopassword',
    'persistent'    => true,
    'table_prefix'  => 'icinga_',
);
```

Unterstützte Backends

Aktuell sind folgende Backend Typen verfügbar. Mehr Information finden Sie unter doc/icinga-api-types.txt.

- IDOUtils DB - OK
- Livestatus Modul - experimentell, noch nicht produktiv einsetzbar.
- Dateibasierend, status.dat - experimentell, noch nicht produktiv einsetzbar.

Benutzung

1. Datenermittlung

Host-Namen und zugehörige Zustände

Erzeugen Sie eine Instant der Klasse IcingaApi:

```
$api = IcingaApi::getConnection(IcingaApi::CONNECTION_IDO, $idoConfig);
```

Erzeugen Sie die Suchkriterien:

```
$apiRes = $api->createSearch()
->setSearchTarget(IcingaApi::TARGET_HOST)
->setResultColumns(array('HOST_NAME', 'HOST_CURRENT_STATE'))
->fetch();
```

Mit Hilfe von setSearchFilter() können Sie Filter benutzen, um die Suche einzuschränken:

```
$apiRes = $api->createSearch()
->setSearchTarget(IcingaApi::TARGET_HOST)
->setResultColumns(array('HOST_NAME', 'HOST_CURRENT_STATE'))
->setSearchFilter(HOST_NAME, 'Switch%', IcingaApi::MATCH_LIKE)
->fetch();
```

2. Verarbeiten der Ergebnisse

```
foreach($apiRes as $apiHandle){
    echo 'Host '.$apiHandle->HOST_NAME.' has state '.$apiHandle->HOST_CURRENT_STATE.'  
>';
}
```

Ausgabe ohne Filter:

```
Host localhost has state 0
Host MySql has state 0
Host router-01 has state 0
Host windows100 has state 0
Host Apache_01 has state 0
```

Ausgabe mit Filter:

```
Host switch70 has the current state 0
Host switch71 has the current state 0
Host switch72 has the current state 0
Host switch73 has the current state 0
Host switch74 has the current state 0
Host switch75 has the current state 0
Host switch76 has the current state 0
Host switch77 has the current state 0
```

3. Kompletter Code ohne die Nutzung von Filtern

```
<?
// Path to icinga api file
$apiFile = 'icinga-api/IcingaApi.php';

// Database connection
$idoConfig = array (
    'type'      => 'mysql',
    'host'       => 'localhost',
    'database'   => 'ido',
    'user'       => 'idouser',
    'password'   => 'idopassword',
    'persistent' => true,
    'table_prefix' => 'icinga_',
);

// Include required files
require_once($apiFile);

// Instance the class
$api = IcingaApi::getConnection(IcingaApi::CONNECTION_IDO, $idoConfig);

// Create search
$apiRes = $api->createSearch()
->setSearchTarget(IcingaApi::TARGET_HOST)
->setResultColumns(array('HOST_NAME', 'HOST_CURRENT_STATE'))
->fetch();

// Create output
foreach($apiRes as $apiHandle){
    echo 'Host '.$apiHandle->HOST_NAME.' has the current state '.$apiHandle->HOST_CURRENT_STATE.'  
>';
}
?>
```

Für nähere Informationen werfen Sie bitte einen Blick in das [git repository](#) oder die Beispiele im doc/examples-Verzeichnis.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Liste der externen Befehle](#)

[Zum Anfang](#)

[Die Icinga-Web REST API](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Die Icinga-Web REST API

[Zurück](#)

[Kapitel 11. Entwicklung](#)

[Weiter](#)

Die Icinga-Web REST API

In dieser Anleitung beschreiben wir Ihnen die Icinga-Web REST API. Sie erlaubt Ihnen Ihre Überwachungsinformationen via GET oder POST abzufragen (in Zukunft (>1.2) auch via PUT).

Warum sollten Sie die API benutzen?

Den meisten Anwendern genügt der Einsatz von Icinga und Icinga-Web. Sie können den Status Ihres Monitoring sehen, auf aktuelle Probleme reagieren und Icinga-Web um gewünschte Module und Cronks erweitern.

Wenn Sie allerdings eine zusätzliche Software einsetzen möchten, die Ihre Monitoring-Daten abfragen soll (zum Beispiel: [Icinga-Chromed-Status](#)), kann Ihnen die Icinga-Web API sehr dienlich sein.

Sie können natürlich die Ausgabe des Icinga Classic UI analysieren (parsen) (so verfahren zur Zeit viele Programme, wie zum Beispiel nagstamon oder das Firefox Plugin Nagios Checker), aber das ist keine wirklich performante Lösung- und außerdem keine Freude für den Entwickler :-).

Die Icinga-Web REST API stellt Ihnen die Daten zur Verfügung, die Sie benötigen (und auch nur diese). Die Daten werden in einem standardisierten, maschinenlesbaren Format wie JSON oder XML zur Verfügung gestellt.

Features der Icinga-Web REST API

Derzeit unterstützt (v1.2):

- Verfügbarkeit von nahezu allen Überwachungsdaten via GET oder POST.
- Rückgabe der Daten als xml oder json.
- AND & OR- Suche über Filtergruppen mit unbegrenzten Verschachtelungsebenen (AND)).
- Sie wählen, welche Spalten zurückgegeben werden sollen, nicht die API (weniger Overhead).
- Unterstützung von "of limit, offset, order oder group by".
- Rückgabe eines zusätzlichen Gesamtwert-Feldes.

- Autorisierung über auth_key in Requests oder Cookies.
- Respektiert Icinga-Web Prinzipien (z.B. die Begrenzung auf bestimmte Hostgruppen).

Zukünftig unterstützt(> 1.2):

- Senden von Kommandos via PUT

Was ist der Unterschied zwischen der Icinga-API und der Icinga-Web REST API?

Die Icinga-API kann als ein internes Toolkit für den Zugriff auf die Datenbankinformationen angesehen werden. In der Tat wirkt die REST-API im oberen Teil der API und bedient sich des HTTP-Protokolls. In Zukunft wird die Icinga-API mit Icinga-Web zusammengeführt werden.

Voraussetzungen

Um die API verwenden zu können, müssen Sie zunächst den Auth-Provider aktivieren icinga-web/app/modules/AppKit/config/auth.xml .

Ändern von "auth_enabled" zu 'true':

```
># vi icinga-web/app/modules/AppKit/config/auth.xml

<ae:parameter name="auth_key">
  <ae:parameter name="auth_module">AppKit</ae:parameter>
  <ae:parameter name="auth_provider">Auth.Provider.AuthKey</ae:parameter>
  <ae:parameter name="auth_enable">true</ae:parameter>
  <ae:parameter name="auth_authoritative">true</ae:parameter>
</ae:parameter>
```



Anmerkung

Wenn Sie ein *.xml-File editieren, müssen Sie anschließend den Cache leeren!

```
rm -f app/cache/config/*.php
```

oder

```
icinga-web/bin/clearcache.sh
```

Nun brauchen Sie in Icinga-Web einen Benutzer mit API-Zugriffsberechtigung, bitte anlegen in Ihrem Icinga-Web :

- Anlegen einer neuen Benutzers
- Auswählen von auth_key in dem Auth_via Feld
- Einfügen des zu nutzenden API-Schlüssels
- Unter principals, hinzufügen von appkit.api.access principal

Das war es, nun können Sie starten.

Referenzen

In den nächsten Sätzen erläutern wir, wie auf die API zugegriffen werden kann:

GET

Vorteile:

- Leicht zu nutzen, es ist eine URL!
- Sie können immer nachvollziehen, welche Parameter angefragt wurden.

Nachteile:

- Wenn Sie in einem Browser Ihre Anfrage absetzen, könnte Ihr API-Key in der Browser-History gespeichert werden.
- In einem Browser können Sie keine URLs mit unbegrenzter Größe ansprechen (2.083 Zeichen beim Internet Explorer).

Die Struktur der URL:

Um die API anzusprechen, sollte die URL folgendermaßen aufgebaut sein (Fett markierte Werte sind erforderlich) host.com/icinga-web/web/api/ **TARGET** / **COLUMNS** / **FILTER** / **ORDER** / **GROUPING** / **LIMIT** / **COUNTFIELD** / **OUTPUT_TYPE**

Die Parameter en Detail:

- TARGET: Welches Feld wird angesprochen, es ist ein einfacher String wie host.
- COLUMNS: Liste der angefragten Spalten, die Syntax muss wie folgt aussehen: columns[COL1|COL2|COL3|...]
- FILTER: Definiert welche Filter im Request verwendet werden. Gültig sind AND oder OR Gruppen.

Der Filter sieht wie folgt aus:

```
filters[AND/OR(COLUMN|OPERATOR|VALUE;COLUMN2|OPERATOR2|VALUE2;OR(...),AND]
```

Beispiel: Select auf alle Services, deren Name "snmp" enthält, wenn sie sich im Status ok oder unknown befinden

Falsch:

```
filters[SERVICE_NAME|like|*smtp*;OR(SERVICE_CURRENT_STATE|=|0;SERVICE_CURRENT_STATE|=|3)]
```

Sie benötigen immer eine Schachtelungsebene am Beginn:

Korrekt:

```
filters[AND( SERVICE_NAME|like|*smtp*;OR( SERVICE_CURRENT_STATE|=|0;SERVICE_CURRENT_STATE|=|3) ) ]
```

- ORDER: Definiert welches Feld für die Ordnung verwendet wird und ob eine aufsteigende oder absteigende Sortierung verwendet wird. Beispiel: order[COLUMN|ASC or DESC]
- GROUPING: Definiert ein Gruppierungsfeld: group[COL]
- LIMIT: Definiert eine Start-Offset und / oder eine Begrenzung: limit[START;END (if needed)]

- COUNTFIELD: Fügt ein INSGESAMT-Feld dem Ergebnis hinzu, welches hochgezählt wird (in den meisten Fällen, die id): countColumn=COL
- OUTPUT: zur Zeit json order xml

Beispiele für GET

GET alle Dienste die kritisch oder warning sind und deren Host im Status ok ist. Absteigend sortiert nach Dienststatus und Hochzählen der Services. Authentifikation via authkey (hier: APITEST123456). Für die bessere Lesbarkeit ist die Anfrage in mehrere Zeilen aufgeteilt, XML:

```
http://localhost/icinga-web/web/api/service/filter[AND(HOST_CURRENT_STATE |= 0;OR(SERVICE_CURRENT_STATE |=n|1;SERVICE_CURRENT_STATE |=| 2))]/
columns(SERVICE_NAME|HOST_NAME|SERVICE_CURRENT_STATE|HOST_NAME|HOST_CURRENT_STATE|HOSTGROUP_NAME)/
order(SERVICE_CURRENT_STATE;DESC)/countColumn=SERVICE_ID/authkey=APITEST123456/xml
```

So sieht die Rückgabe aus:

```
<results>
  <result>
    <column name="SERVICE_ID">295</column>
    <column name="SERVICE_OBJECT_ID">139</column>
    <column name="SERVICE_IS_ACTIVE">1</column>
    <column name="SERVICE_INSTANCE_ID">1</column>
    <column name="SERVICE_NAME">MailQ</column>
    <column name="SERVICE_DISPLAY_NAME">MailQ</column>
    <column name="SERVICE_OUTPUT">Error occured:error=1:0:0</column>
    <column name="SERVICE_PERFDATA"></column>
  </result>
  <result>
    <column name="SERVICE_ID">311</column>
    <column name="SERVICE_OBJECT_ID">155</column>
    <column name="SERVICE_IS_ACTIVE">1</column>
    <column name="SERVICE_INSTANCE_ID">1</column>
    <column name="SERVICE_NAME">POP3</column>
    <column name="SERVICE_DISPLAY_NAME">POP3</column>
    <column name="SERVICE_OUTPUT">Verbindungsauftbau abgelehnt</column>
    <column name="SERVICE_PERFDATA"></column>
  </result>
  <total>2</total>
</results>
```

Wenn Sie das Format von xml zu json ändern, bekommen Sie die gleichen Informationen (plus zusätzliche Informationen für ExtJS, falls Sie sie nicht benötigen, können Sie diese ignorieren) im json Format:

```
{ "metaData": {
  "paramNames": { "start": "limit_start", "limit": "limit" },
  "totalProperty": "total",
  "root": "result",
  "fields": null },
  "result": [
    {
      "SERVICE_ID": "295",
      "SERVICE_OBJECT_ID": "139",
      "SERVICE_IS_ACTIVE": "1",
      "SERVICE_INSTANCE_ID": "1",
      "SERVICE_NAME": "MailQ",
      "SERVICE_DISPLAY_NAME": "MailQ",
      "SERVICE_OUTPUT": "Error occured:error=1:0:0",
      "SERVICE_PERFDATA": ""
    },
    {
      "SERVICE_ID": "311",
      "SERVICE_OBJECT_ID": "155",
      "SERVICE_IS_ACTIVE": "1",
      "SERVICE_INSTANCE_ID": "1",
      "SERVICE_NAME": "POP3",
      "SERVICE_DISPLAY_NAME": "POP3",
      "SERVICE_OUTPUT": "Verbindungsauftbau abgelehnt"
    }
  ]
}
```

```

        "SERVICE_OUTPUT": "Verbindungsauftbau abgelehnt",
        "SERVICE_PERFDATA": ""
    ],
    "success": "true",
    "total": "2"
}

```



Anmerkung

Wenn Sie den countField-Parameter nicht verwenden, bekommen Sie eine "flat" json-Datei mit diesem Ergebnis.

POST

Vorteile:

- Unbegrenzte Parametergröße für große Anfragen.
- Ihre verwendeten Parameter erscheinen nicht in der Browser-Historie, lediglich die Basis-URL.
- Leichter in Applikationen zu integrieren

Nachteile:

- Der POST wird über den Header gesendet, deshalb können Sie den Request nicht einfach über das Addressfeld des Browsers absetzen.

Die Parameter en Detail

Der Link entspricht dem GET-Basislink, allerdings mit der Angabe des Ausgabeformates: Zum Beispiel: host.com/icinga-web/web/api/json. Folgende Parameter werden unterstützt:

- 'target': Welches Feld wird angesprochen, es ist ein einfacher String wie "host"
- 'columns[]': Ein Array von Spalten

Example:

columns [0] = SERVICE_NAME

columns [1] = SERVICE_ID

- 'groups[]': nach diesem Feld gruppieren
- 'filters_json': Ein json beschreibender "how to" Filter

Example:

```
[
  {
    "type": "AND",
    "field": [
      {
        "type": 'atom',
        "field": ['SERVICE_NAME'],
        "method": ['like'],
        "value": ['*pop*']
      },
      {
        "type": 'OR',
        "field": [
          {
            "type": 'atom',
            "field": ['SERVICE_CURRENT_STATE'],
            "method": ['eq'],
            "value": ['down']
          }
        ]
      }
    ]
}
```

```

        "method": [ '>' ],
        "value": [ 0 ]
    }, {
        "type": 'atom',
        "field": [ 'SERVICE_IS_FLAPPING' ],
        "method": [ '=' ],
        "value": [ 1 ]
    }
}
}
}
]

```

- 'order_col' : Spalte nach der sortiert wird
- 'order_dir' : Sortierungsreihenfolge (asc oder desc)
- 'limit_start' : Start-Offset des Records
- 'limit' : Begrenzt das Ergebnis auf x Antworten
- 'countColumn' : Fügt ein INSGESAMT-Feld dem Ergebnis hinzu

Beispiele für POST

Nehmen wir das Beispiel aus "Beispiel für GET" und benutzen nun eine POST-Anfrage. Wir werden curl verwenden, so dass das Beispiel auf der Konsole wiederholt werden kann:

```

curl
-d 'target=service'
-d 'filters[0][type]:"AND", "field": [{"type": "atom", "field": ["HOST_CURRENT_STATE"], "method": ["*="], "value": [0]}, {"type": "OR", "field": [{"type": "atom", "field": ["SERVICE_CURRENT_STATE"], "method": ["*="], "value": [1]}, {"type": "atom", "field": ["SERVICE_CURRENT_STATE"], "method": ["*="], "value": [2]}]}]
-d columns[0]=SERVICE_NAME
-d columns[1]=HOSTNAME
-d columns[2]=HOSTGROUP_NAME
-d columns[3]=HOSTCURRENTSTATE
-d columns[4]=HOSTCURRENTSTATE
-d columns[5]=HOSTGROUP_NAME
-d columns[6]=HOSTCURRENTSTATE/DESC
-d countColumn=SERVICE_ID
-d 'authkey=API123456'
http://localhost:5665/icinga-web/web/api/xml

```

Dies gibt uns das gleiche Ergebnis zurück, wie zuvor in der GET-Anfrage.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

Installation und Benutzung der
Icinga-API

[Zum Anfang](#)

Kapitel 12. IDOUtils



Kapitel 12. IDOUtils

[Zurück](#)[Weiter](#)

Kapitel 12. IDOUtils

Inhaltsverzeichnis

[Einleitung](#)[Zweck](#)[Design-Überblick](#)[Instanzen](#)[Installation](#)[Komponenten](#)[Überblick](#)[IDOMOD](#)[LOG2IDO](#)[FILE2SOCK](#)[IDO2DB, IDO2DB](#)[Konfiguration der IDOUtils](#)[IDOMOD Konfigurationsoptionen](#)[IDO2DB Konfigurationsoptionen](#)[Beispielkonfigurationen](#)[Einzelner Server, einzelne Instanz](#)[Einzelner Server, mehrere Instanzen](#)[Einzelner Server, einzelne Instanz, Log-Datei-Import](#)[IDOUtils Database Model](#)[Central Tables](#)[Debugging Tables](#)[Historical Tables](#)[Current Status Tables](#)[Configuration Tables](#)[Datenbank-Anpassungen/Änderungen](#)[Zurück](#)[Weiter](#)[Die Icinga-Web REST API](#)[Zum Anfang](#)[Einleitung](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Einleitung

[Zurück](#)

Kapitel 12. IDOUtils

[Weiter](#)

Einleitung

Das IDOUtils-Addon basiert auf den NDOUtils, die ursprünglich vom Nagios (R)-Entwickler Ethan Galstad erstellt wurden, so dass die Grundlagen sowohl auf Nagios als auch auf Icinga zutreffen.

Zweck

Das IDOUtils-Addon dient dazu, alle Konfigurations- und Ereignisdaten von Icinga in einer relationalen Datenbank zu abzulegen. Das Speichern der Informationen von Icinga in einem RDBMS erlaubt die schnellere Abfrage und Verarbeitung der Daten. Die Icinga-API nutzt diese Daten.

Bisher werden MySQL, Oracle und PostgreSQL von diesem Addon unterstützt. Andere Datenbanksysteme werden ggf. unterstützt, wenn sich genügend interessierte Benutzer und vor allem Benutzer finden, die Tests durchführen.

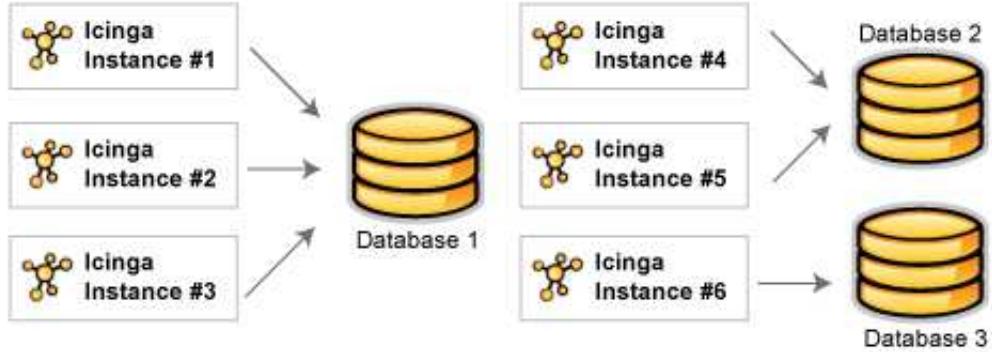
Design-Überblick

Das IDOUtils-Addon wurde entwickelt für Benutzer mit:

- einer Icinga-Installation
- mehrere einzelne oder "Vanilla"-Icinga-Installationen
- mehrere Icinga-Installationen in verteilten, redundanten und/oder Umgebungen mit Ausfallsicherung

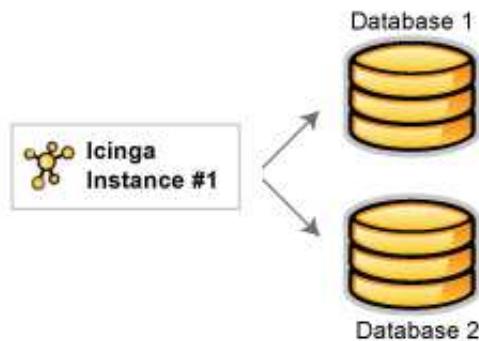
Daten eines Icinga-Prozesses (im weiteren als "Instanz" bezeichnet) können entweder in der gleichen oder in verschiedenen Datenbanken wie die Daten von anderen Icinga-Instanzen gespeichert werden.

Abbildung 12.1. Mögliche Anordnungen



Obwohl es bisher nicht unterstützt wird, könnten in der Zukunft die Daten einer beliebigen Instanz in mehrere Datenbanken gespeichert werden, falls das gewünscht wird..

Abbildung 12.2. zukünftige Entwicklung: Eine Instanz, mehrere Datenbanken

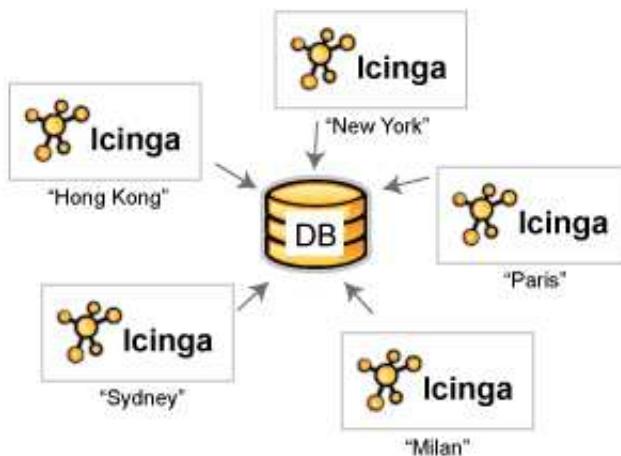


Instanzen

Jeder Icinga-Prozess, egal ob es ein einzelner Überwachungsserver oder Teil eines verteilten, redundanten Setups ist, ggf. mit Ausfallsicherung, wird als "Instanz" bezeichnet. Um die Integrität der gespeicherten Daten zu gewährleisten muss jede Icinga-Instanz mit einem eindeutigen Bezeichner oder Namen gekennzeichnet werden.

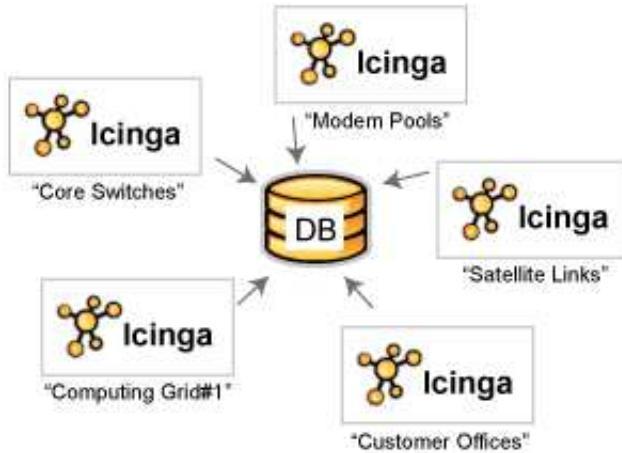
Sie können den Namen jeder Icinga-Instanz Ihren Bedürfnissen anpassen. So können Sie z.B. die Icinga-Instanzen aufgrund des geografischen Standorts bezeichnen....

Abbildung 12.3. Instanznamen basierend auf dem geografischen Standorts



Oder Sie können die Icinga-Instanzen nach dem Zweck benennen...

Abbildung 12.4. Instanznamen basierend auf dem Zweck



Wie Sie die Icinga-Instanzen nennen, bestimmen Sie. Wichtig ist dabei lediglich, dass jeder einzelne Icinga-Prozess einen eindeutigen Instanznamen erhält.

Mehr Informationen dazu, welche Rolle die Instanznamen spielen, gibt es in den nächsten Abschnitten.

Installation

Die Installation der IDOUtils wird im [Quickstart IDOUtils](#) beschrieben.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Kapitel 12. IDOUtils](#)

[Zum Anfang](#)

[Komponenten](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Komponenten

[Zurück](#)

Kapitel 12. IDOUtils

[Weiter](#)

Komponenten

Überblick

Es gibt vier Hauptkomponenten, aus denen die IDO-Utilities bestehen:

1. IDOMOD-Event-Broker-Modul
2. LOG2IDO-Utility
3. FILE2SOCK-Utility
4. IDO2DB-Daemon

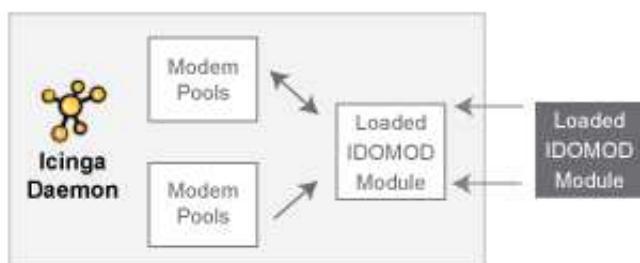
Jede Komponente wird auf den folgenden Seiten genauer beschrieben.

IDOMOD

Die IDO-Utilities enthalten ein Icinga-Event-Broker-Modul (IDOMOD.O), das die Daten des Icinga-Daemon exportiert.

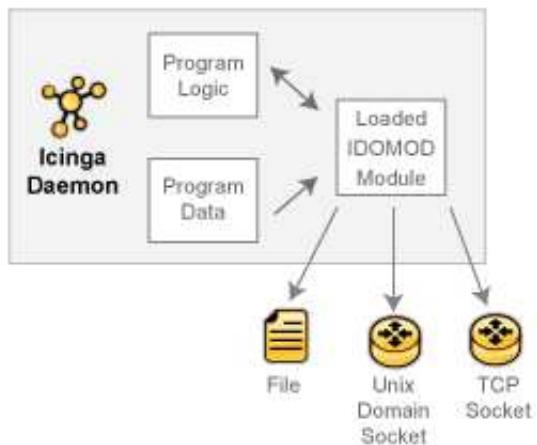
Wenn wir annehmen, dass Icinga mit aktivierten Event-Broker-Modul kompiliert wurde (das ist der Default), dann können Sie Icinga konfigurieren, dass das IDOMOD-Modul während der Laufzeit geladen wird. Sobald das Modul vom Icinga-Daemon geladen wird, kann es auf die Daten und die Logik des laufenden Icinga-Prozesses zugreifen.

Abbildung 12.5. Geladenes IDOMOD-Event-Broker-Modul



Das IDOMOD-Modul wurde konzipiert, um sowohl Konfigurationsdaten als auch Informationen über verschiedene Laufzeiteignisse, die während des Überwachungsprozesses auftreten, aus dem Icinga-Daemon zu exportieren. Das Modul kann diese Daten an eine normale Datei, einen Unix-Domain- oder einen TCP-Socket senden.

Abbildung 12.6. IDOMOD-Möglichkeiten



Das IDOMOD-Modul schreibt die Daten in einem Format, dass der IDO2DB-Daemon (wird später beschrieben) verstehen kann.

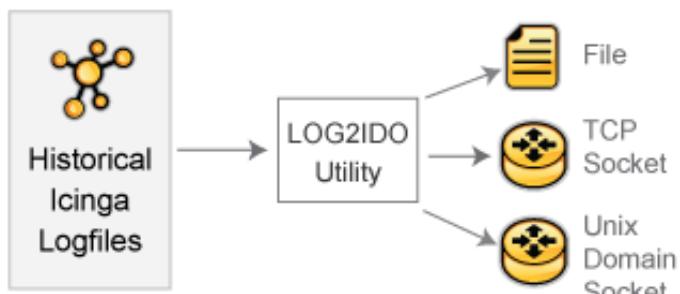
Falls das IDOMOD-Modul die Daten in eine Datei schreibt, dann können Sie es so konfigurieren, die Ausgabedatei regelmäßig rotiert und/oder mit einem vordefinierten Icinga-Befehl verarbeitet wird. Dies kann nützlich sein, wenn Sie die Ausgabedatei zu einer anderen Maschine übertragen möchten (mit SSH, etc.) oder den Inhalt mit dem FILE2SOCK-Utility (wird später beschrieben) an den IDO2DB-Daemon senden möchten.

Falls das IDOMOD-Modul die Daten an einen TCP- oder Unix-Domain-Socket schickt, dann gibt es ein wenig Schutz gegen Verbindungsabbrüche. Das Modul wird versuchen, die Ausgaben zwischenzuspeichern, bis es sich (erneut) mit dem Socket verbinden kann. Dies ist hilfreich, wenn der Prozess, der den Socket anlegt bzw. darauf lauscht, (erneut) gestartet werden muss.

LOG2IDO

Das LOG2IDO-Utility wurde entwickelt, damit Sie über den IDO2DB-Daemon ([wird später beschrieben](#)) historische Datei aus Icinga-, Nagios- und NetSaint-Log-Dateien in eine Datenbank importieren können. Das Utility funktioniert, indem historische Log-Dateidaten in einem Format, das der IDO2DB-Daemon versteht, an eine normale Datei, ein Unix-Domain- oder ein TCP-Socket geschickt werden. Der IDO2DB-Daemon kann dann genutzt werden, um diese Ausgaben zu verarbeiten und die Informationen in einer Datenbank zu speichern.

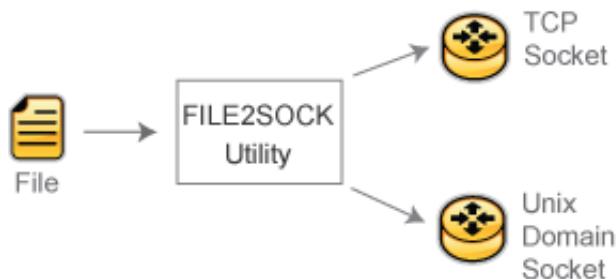
Abbildung 12.7. LOG2IDO-Utility



FILE2SOCK

Das FILE2SOCK-Utility ist ziemlich simpel. Es liest Eingaben von einer normalen Daten (oder STDIN) und schickt diese Daten entweder an einen Unix-Domain- oder einen TCP-Socket. Die gelesenen Datei werden nicht bearbeitet, bevor sie an den Socket geschickt werden.

Abbildung 12.8. FILE2SOCK-Utility



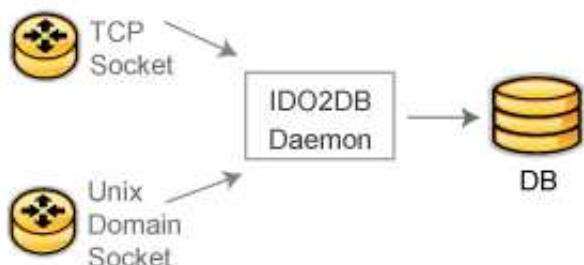
Dieses Utility ist nützlich, wenn Sie die Ausgaben des IDOMOD-Event-Broker-Moduls und/oder des LOG2IDO-Utilitys in eine normale Datei umleiten. Sobald diese Komponenten ihre Ausgaben in eine Datei geschrieben haben, können Sie das FILE2SOCK-Utility nutzen, um den Inhalt der Datei an den TCP- oder den Unix-Domain-Socket des IDO2DB-Daemons zu schicken.

IDO2DB

Das IDO2DB-Utility ist gedacht, um die Datenausgaben der IDOMOD- und LOG2IDO-Komponenten zu nehmen und in einer MySQL-, Oracle- oder PostgreSQL-Datenbank zu speichern.

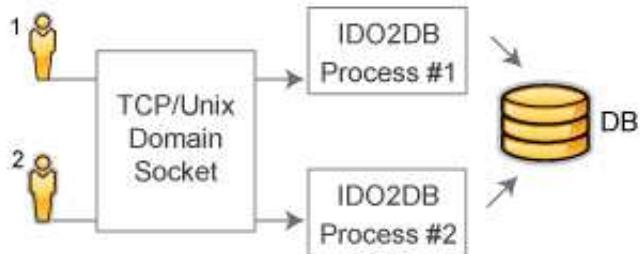
Beim Start des IDO2DB-Daemons legt dieser entweder einen TCP- oder einen Unix-Domain-Socket an und wartet auf Client, die sich verbinden. IDO2DB kann entweder als einzelner, als Multiprozess-Daemon oder unter INETD laufen (wenn ein TCP-Socket verwendet wird).

Abbildung 12.9. IDO2DB-Daemon



Mehrere Clients können sich mit dem Socket des IDO2DB-Daemons verbinden und gleichzeitig Daten übertragen. Für jeden neuen Client, der sich verbindet, wird ein separater IDO2DB-Prozess erzeugt. Die Daten jedes Clients werden gelesen und in einer benutzerdefinierten Datenbank für spätere Abfragen und Verarbeitung gespeichert.

Abbildung 12.10. IDO2DB mit mehreren Clients



Der IDO2DB-Daemon unterstützt im Moment MySQL-, Oracle- und PostgreSQL-Datenbanken.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Einleitung](#)

[Zum Anfang](#)

[Konfiguration der IDOUtils](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Konfiguration der IDOUtils

[Zurück](#)
[Kapitel 12. IDOUtils](#)
[Weiter](#)

Konfiguration der IDOUtils



Anmerkung

Dies sollte als "in Bearbeitung" angesehen werden.

[Änderungen an den Core-Dateien](#)

[Änderungen in idomod.cfg](#)

[Hinweise zu Performance und Fehlersuche](#)

[Änderungen in ido2db.cfg](#)

[idomod.cfg Konfigurationsoptionen](#)

[ido2db.cfg Konfigurationsoptionen](#)

Core

Die Konfiguration beginnt mit dem Setzen der Direktive "broker_options" in `icinga.cfg`. In den meisten Fällen ist dieser Wert in der Datei bereits vorhanden, muss aber ggf. aktiviert werden (durch das Entfernen des führenden Hash-Zeichens).



Wichtig

Bitte beachten Sie, dass diese Einstellung **alle** Event-Broker-Module beeinflusst!
(Details siehe

<http://www.mail-archive.com/nagios-users@lists.sourceforge.net/msg24002.html>.

Aktivieren Sie das idomod-Event-Broker-Modul. Bitte beachten Sie, dass die folgende Modul-Definition unter normalen Umständen bereits im modules-Unterverzeichnis existiert, so dass keine Notwendigkeit besteht, die Hauptkonfigurationsdatei zu ändern.

Falls nicht, dann kann die Definition des Broker-Moduls mit Hilfe einer [module-Definition](#) erfolgen, die der folgenden ähnelt:

```
define module {
    module_name  ido_mod
    path         /usr/local/icinga/bin/idomod.o
    module_type  neb
    args        config_file=/usr/local/icinga/etc/idomod.cfg
}
```



Anmerkung

Das bedeutet auch, dass ein broker_module-Eintrag für idomod in `icinga.cfg` nicht gleichzeitig aktiviert werden darf! "define module" ist verfügbar ab Icinga 1.4.

idomod.cfg

Bitte überprüfen Sie die Einstellungen, bevor Sie Icinga (neu)starten.

Die Direktive "output_type" sollte auf "unixsocket" (default) oder "tcpsocket" gesetzt werden, abhängig davon, ob der IDO2DB-Daemon auf einen lokalen oder entfernten Rechner läuft. Bitte setzen Sie den Wert von "output" entsprechend (und außerdem "tcp_port", falls zutreffend). Die anderen Optionen sind auf Werte gesetzt, die für einen ersten Betrieb funktionieren. Falls es während der Laufzeit Schwierigkeiten gibt, dann sollten Sie einen Blick auf die folgenden Zeilen werfen:

Hinweise zu Performance und Fehlersuche

- *Puffer*

Das Erhöhen der Puffergröße wird in Fällen helfen, wo es "Aussetzer" durch einen langsamen ido2db-Daemon oder eine Datenbank auf der anderen Seite gibt. Bedenken Sie aber, dass ein zu hoher Wert für mehr Core-Last durch die Bearbeitung von aktuellen und die Synchronisierung von zwischengespeicherten Daten sorgt. Aus Performance-Gründen ist es eine gute Idee, den Puffer auf eine RAM-Disk zulegen, um File-I/O zu reduzieren. Bei einer virtuellen Umgebung ist es nicht ratsam, den Puffer und die Prüfergebnisse (checkresults-Verzeichnis) / Protokolldateien auf das gleiche Volume zu legen, zusammen mit der `status.dat` wird ggf. funktionieren.

```
output_buffer_items=
buffer_file=
```

- *Reconnect*

Es kann passieren, dass ein langsamer ido2db-Daemon oder ein RDBMS die Verbindung zum Client wegen Timeout abbricht. Der Output-Puffer wird dann gefüllt, während auf ein Reconnect gewartet wird. Um einen schnelleren Reconnect zu erreichen, können Sie den Default-Wert des Reconnect verringern. Bitte beachten Sie dabei, dass ein zu geringer bzw. ein zu großer Wert zu Performance-Beeinträchtigungen führen kann. Dies erfordert ein wenig Ausprobieren.

```
reconnect_interval=15
reconnect_warning_interval=15
```

Datenverarbeitungsoptionen

Entscheiden Sie, was Sie benötigen. Möchten Sie historische Daten behalten oder ist es lediglich eine Datenbank, die aktuelle Werte enthält?

- Normaler Betrieb
 - auf *TIMED_EVENT_DATA verzichten (Default)
- Icinga-Web-Erfordernisse

- auf *TIMED_EVENT_DATA verzichten
- auf *CHECK_DATA verzichten
- *LOG_DATA behalten
- *STATUS_DATA behalten
- *STATE_HISTORY behalten
- Config tables
- Nur Status-Daten
 - auf *CHECK_DATA verzichten
 - auf *TIMED_EVENT_DATA verzichten
 - auf *LOG_DATA verzichten
- Historische Daten
 - *LOG_DATA betrachten
 - *CHECK_DATA betrachten
 - ältere, statische Daten in separate Tabellen auslagern
 - die "Housekeeping"-Zyklen in `ido2db.cfg` verwenden
- Nur Performance-Daten
 - auf *LOG_DATA verzichten
 - keep *CHECK_DATA

Benutzen Sie den [Rechner von Consol](#), um den richtigen Wert auf einfache Weise zu berechnen.

Für Einzelheiten werfen Sie einen Blick auf die ["data_processing_options"-Direktive](#).

- *Konfigurationsausgabeoptionen*

Diese Optionen legen fest, in welchem Umfang die Konfiguration bei einem Start/Restart/Reload des Core ausgegeben wird (Haupt- und Objektkonfiguration, abhängig von den "data processing options"). Bitte bedenken Sie, dass ein "reload" lediglich ein "SIGHUP restart" ist, bei dem weder Speicher noch Werte verworfen (bzw. neu geladen) werden. Intern wird das IDOMOD auf die gleiche Weise handeln, indem die Daten durch die NEB-API geholt werden.

Wenn Sie Icinga-Web benutzen, dann ist der Standard momentan "2" ("aufbewahrte Konfigurationsdaten"), so dass Datenaufbewahrung (retention data) aktiviert sein muss (Standard) und die Werte mindestens einmal gelesen werden mussten (also starten Sie und führen Sie "reload" durch, um die Konfigurationsdaten zu diesem Zeitpunkt auszugeben). Dies wird nicht so lange dauern wie die ursprüngliche Ausgabe der Konfiguration, weil die Konfiguration bereits kompiliert/zusammengestellt wurde in die resultierenden Objektdaten.

Ausgehend von der originalen Konfiguration wird dies einen "Hook" an den Core-Config-Parser anhängen und jede einzelne Direktive auch zu IDOMOD hinzufügen und dadurch zusätzliche CPU-Zyklen benötigen und viele Daten während des (Neu-)Starts erzeugen. Icinga-Web kann angewiesen werden, diese Daten ebenfalls zu nutzen.

Innerhalb der Datenbank gibt es in den Tabellen icinga_hosts und icinga_services die Spalte config_type, wobei

- config_type=1 <==> config_output_options=2 (retained)
- config_type=0 <==> config_output_options=1 (original)

config_type selbst kann für die Icinga-Web-Konfiguration als XML-Attribut genutzt werden.

Für Einzelheiten werfen Sie einen Blick auf die "["config_output_options"-Direktive](#)".

- *Debug*

Bitte bedenken Sie, dass das Schreiben einer Debug-Datei das Modul verlangsamen kann. Allerdings wird es helfen, um zu analysieren, welche Daten vom Core durch idomod über den Socket zu ido2db laufen.

ido2db.cfg

- *Tabellenbereinigungsoptionen*

Per Default werden "timed events" nicht verarbeitet, weil die verbleibenden Daten ziemlich nutzlos wären. Der Rest kann bis auf aktuelle Daten bereinigt werden, wobei die Aktualität durch das maximale Alter festgelegt wird.

Es scheint eine gute Idee zu sein, dies insbesondere für die Tabelle logentries zu tun, weil diese schnell sehr groß werden kann.

Bitte bedenken Sie: Wenn Sie den Wert von instance_name in idomod.cfg ändern, verbleiben die historischen weiterhin in der Datenbank. Der Aufräumprozess wird lediglich Spalten mit der gerade aktiven instance_id bearbeiten (die von IDOMOD und DB während des ido2db-Starts gelesen wurde).

```
## TABLE TRIMMING OPTIONS
# Several database tables containing Icinga event data can become quite large
# over time. Most admins will want to trim these tables and keep only a
# certain amount of data in them. The options below are used to specify the
# age (in MINUTES) that data should be allowed to remain in various tables
# before it is deleted. Using a value of zero (0) for any value means that
# that particular table should NOT be automatically trimmed.
#
# Remember: There are no optimized settings, it depends on your rdbm install,
# number/checkinterval of host/service-checks and your desired time of data
# savings - historical vs live-data. Please keep in mind that low delete
# intervals may interfere with insert/update data from Icinga.

# ***DEFAULT***

# Keep timed events for 1 hour
max_timedeevents_age=60

# Keep system commands for 1 day
max_systemcommands_age=1440

# Keep service checks for 1 day
```

```

max_servicechecks_age=1440

# Keep host checks for 1 day
max_hostchecks_age=1440

# Keep event handlers for 1 week
max_eventhandlers_age=10080

# Keep external commands for 1 week
max_externalcommands_age=10080

# Keep logentries for 31 days
max_logentries_age=44640

# Keep acknowledgements for 31 days
max_acknowledgements_age=44640

```

Beginnend mit Icinga 1.6 wurde die Bereinigung um die Benachrichtigungstabellen erweitert, so dass die folgenden Direktiven verfügbar sind.

```

# Keep notifications for 31 days
max_notifications_age=44640

# Keep contactnotifications for 31 days
max_contactnotifications_age=44640

# Keep contactnotificationmethods for 31 days
max_contactnotificationmethods_age=44640

```

- *Datenbankaufräumintervall*

Diese Option kann genutzt werden, um das allgemeine Aufräumintervall anzupassen (die Zeit zwischen den Zeitpunkten, zu denen die Löschbefehle tatsächlich gegen die Datenbank laufen). Dieser Wert wurde mit Icinga 1.4 aus Performance-Gründen von 60 auf 3600 Sekunden erhöht (also nicht mehr minütlich, sondern stündig).

Bitte beachten Sie, dass die Tabellenbereinigungsoptionen Hand in Hand mit diesem Wert gehen. Das bedeutet z.B., dass durch das Setzen von max_logentries_age auf 30 Minute auch das Aufräumintervall auf einen Wert von weniger als 30*60 Sekunden gesetzt werden muss.

```
trim_db_interval=3600
```

- *Tabellen beim Start löschen*

Per Default sind Host/Service-Zustände und Ausfallzeiten (downtimes) nicht von diesen Werten betroffen, da sie in der Datenbank bleiben. Der Rest wird in jedem Fall bereinigt. Falls Sie verschiedene Daten über einen Core-Restart/Reload benötigen, dann deaktivieren Sie diese Einstellungen:

```

clean_realtime_tables_on_core_startup=1
clean_config_tables_on_core_startup=1

```

IDOMOD Konfigurationsoptionen

Instanzname

Format: **instance_name=<name>**

Example: **instance_name=default**

Diese Option legt den "Namen" fest, der mit dieser Icinga-Instanz verbunden ist und wird benutzt, um Daten von mehreren Instanzen zu unterscheiden. Standard ist 'default' (ohne Anführungszeichen).

Ausgabetyp

Format: **output_type=<file> | <tcpsocket> | <unixsocket>**

Beispiel: **output_type=unixsocket**

Diese Option legt fest, welchen Typ von "Abfluss" das IDO-NEB-Modul für die Datenausgabe nutzen soll.

Mögliche Werte sind:

- file = Standardtextdatei
- tcpsocket = TCP-Socket
- unixsocket = UNIX-Domain-Socket (Default)

Ausgabe

Format: **output=<ip address> | <file name>**

Beispiel: **output=/usr/local/icinga/var/ido.sock**

Diese Option legt Namen und Pfad der Datei oder des UNIX-Domain-Socket fest, wohin die Daten geschickt werden sollen, wenn als Ausgabetyp "file" oder "unixsocket" angegeben wurde. Wenn der Ausgabetyp "tcpsocket" ist, dann wird diese Option dazu genutzt, die IP-Adresse oder den FQDN des Hosts anzugeben, mit dem sich das Modul verbinden soll, um die Daten zu senden.

TCP-port

Format: **tcp_port=<n>**

Beispiel: **tcp_port=5668**

Diese Option legt fest, mit welchem Port sich das Modul verbinden soll, um die Daten zu senden. Diese Option ist nur gültig, wenn als Ausgabetype "tcpsocket" angegeben wurde.

Verschlüsselung benutzen

Format: **use_ssl=**

Beispiel: **use_ssl=0**

Diese Option legt fest, ob das Modul SSL benutzen wird, um den Netzwerkverkehr zwischen Modul und ido2db-Daemon zu verschlüsseln. Beide Seiten müssen dieses Feature aktivieren, das von SSL-Bibliotheken wie openssl oder Kerberos abhängt.

Diese Option ist nur gültig, wenn als Ausgabetype "tcpsocket" angegeben wurde.

Größe des Ausgabepuffers

Format: **output_buffer_items=<n>**

Beispiel: **output_buffer_items=5000**

Diese Option legt die Größe des Ausgabepuffers fest, der verhindern soll, dass Daten bei einem vorübergehenden Verbindungsabbruch zum "Abfluss" verlorengehen. Die angegebene Zahl gibt die Zahl von Zeilen (jeweils mit variabler Länge) an, die gepuffert werden.

Pufferdatei

Format: **buffer_file=<file name>**

Beispiel: **buffer_file=/usr/local/icinga/var/idomod.tmp**

Diese Option legt den Namen einer Datei fest, die zur Speicherung von gepufferten Daten benutzt wird, die nicht an den IDO2DB-Daemon gesendet werden konnten, während Icinga heruntergefahren wird. Vor dem Ende wird das IDO-NEB-Modul alle gepufferten Daten in diese Datei schreiben, damit sie später verarbeitet werden können. Wenn Icinga (neu)startet, wird das IDO-NEB-Modul den Inhalt dieser Datei lesen und ihn an den IDO2DB-Daemon zur weiteren Verarbeitung senden.

Datenrotationsintervall

Format: **file_rotation_interval=<seconds>**

Beispiel: **file_rotation_interval=14400**

Diese Option legt fest, wie oft (in Sekunden) die Datei von Icinga rotiert wird. Dateirotation wird von Icinga erledigt, indem der durch die "file_rotation_command"-Option angegebene Befehl ausgeführt wird. Diese Option hat keine Auswirkung, wenn als Ausgabetyp ein Socket angegeben wurde.

Dateirotationsbefehl

Format: **file_rotation_command=<command>**

Beispiel: **file_rotation_command=rotate_ido_log**

Diese Option gibt den Befehl an (wie in Icinga definiert), der zur Rotation der Ausgabedatei benutzt werden soll im Intervall, das durch die Option "file_rotation_interval" angegeben wurde. Diese Option hat keine Auswirkung, wenn als Ausgabetyp ein Socket angegeben wurde.

Die Datei 'misccommands.cfg' enthält eine Beispiel-command-Definition, die Sie zur Rotation der Log-Datei nutzen können.

Dateirotations-Timeout

Format: **file_rotation_timeout=<seconds>**

Beispiel: **file_rotation_timeout=60**

Diese Option legt die maximale Anzahl von Sekunden fest, die der Rotationsbefehl laufen darf, bevor er abgebrochen wird.

Reconnect-Intervall

Format: **reconnect_interval=<seconds>**

Beispiel: **reconnect_interval=15**

Diese Option legt fest, in welchem Intervall (in Sekunden) das IDO-NEB-Modul versuchen wird, einen Verbindungsaufbau durchzuführen, wenn die Verbindung zur Ausgabedatei oder zum Socket verloren gegangen ist.

Reconnect-Warnintervall

Format: **reconnect_warning_interval=<seconds>**

Beispiel: **reconnect_warning_interval=15**

Diese Option legt fest, in welchem Intervall (in Sekunden) eine Warnmeldung in die Icinga-Protokolldatei geschrieben wird, wenn die Verbindung zur Ausgabedatei oder zum Socket nicht wieder aufgebaut werden kann.

Datenverarbeitungsoptionen

Format: **data_processing_options=<n>**

Beispiel: **data_processing_options=-1**



Achtung

Ändern Sie nicht den Wert dieser Option, wenn Sie die Auswirkungen nicht kennen!!!

Diese Option legt fest, welche Daten vom IDO-NEB-Modul verarbeitet werden.

Lesen Sie den Source-Code (`module/idoutils/include/idomod.h`) und suchen Sie nach `"IDOMOD_PROCESS_"`, um festzulegen, welche Werte hier zu benutzen sind.

Die Werte aus dem Source-Code sollten addiert werden, um den benötigten Wert zu ermitteln. Der Wert "-1" bewirkt, dass alle Daten verarbeitet werden.

Generell stehen die folgenden Werte zur Berechnung zur Verfügung:

<code>#define IDOMOD_PROCESS_PROCESS_DATA</code>	1
<code>#define IDOMOD_PROCESS_TIMED_EVENT_DATA</code>	2
<code>#define IDOMOD_PROCESS_LOG_DATA</code>	4
<code>#define IDOMOD_PROCESS_SYSTEM_COMMAND_DATA</code>	8
<code>#define IDOMOD_PROCESS_EVENT_HANDLER_DATA</code>	16
<code>#define IDOMOD_PROCESS_NOTIFICATION_DATA</code>	32
<code>#define IDOMOD_PROCESS_SERVICE_CHECK_DATA</code>	64
<code>#define IDOMOD_PROCESS_HOST_CHECK_DATA</code>	128
<code>#define IDOMOD_PROCESS_COMMENT_DATA</code>	256
<code>#define IDOMOD_PROCESS_DOWNTIME_DATA</code>	512
<code>#define IDOMOD_PROCESS_FLAPPING_DATA</code>	1024
<code>#define IDOMOD_PROCESS_PROGRAM_STATUS_DATA</code>	2048
<code>#define IDOMOD_PROCESS_HOST_STATUS_DATA</code>	4096

```

#define IDOMOD_PROCESS_SERVICE_STATUS_DATA          8192
#define IDOMOD_PROCESS_ADAPTIVE_PROGRAM_DATA       16384
#define IDOMOD_PROCESS_ADAPTIVE_HOST_DATA          32768
#define IDOMOD_PROCESS_ADAPTIVE_SERVICE_DATA        65536
#define IDOMOD_PROCESS_EXTERNAL_COMMAND_DATA        131072
#define IDOMOD_PROCESS_OBJECT_CONFIG_DATA           262144
#define IDOMOD_PROCESS_MAIN_CONFIG_DATA             524288
#define IDOMOD_PROCESS_AGGREGATED_STATUS_DATA       1048576
#define IDOMOD_PROCESS_RETENTION_DATA               2097152
#define IDOMOD_PROCESS_ACKNOWLEDGEMENT_DATA         4194304
#define IDOMOD_PROCESS_STATECHANGE_DATA              8388608
#define IDOMOD_PROCESS_CONTACT_STATUS_DATA           16777216
#define IDOMOD_PROCESS_ADAPTIVE_CONTACT_DATA         33554432
#
#define IDOMOD_PROCESS_EVERYTHING                   67108863
#
# You may use the Online Calculator by Gerhard Lausser:
# http://labs.consol.de/nagios/ndo-data-processing-options/
# (please note that there is a checkbox for everything!)
#
# The default setting will remove everything not used by default.
#      TIMED_EVENT_DATA      (-2)
#      SERVICE_CHECK_DATA     (-64)
#      HOST_CHECK_DATA        (-128)
#
# 67108863-(2+64+128) = 67108863-194 = 67108669

data_processing_options=67108669

# If you are planning to use NagVis you may want to use the following setting:
#
#data_processing_options=4061953
#
# You may have to experiment in your environment and find the best value yourself!

```

Konfigurationsausgabeoptionen

Format: **config_output_options=<0|1|2|3>**

Beispiel: **config_output_options=2**

Diese Option legt fest, welche Arten der Icinga-Konfigurationsdaten vom IDO-NEB-Modul ausgegeben werden. Die Werte werden addiert. Hinweis: "2" ist der bevorzugte Wert.

Mögliche Werte sind:

- 0 = Keine Konfigurationsinformationen ausgeben
- 1 = Nur die originale Konfiguration ausgeben (aus den Konfigurationsdateien)
- 2 = Nur die Konfiguration nach dem Wiederherstellen der aufbewahrten Informationen ausgeben
- 3 = Originale und wiederhergestellte Konfiguration ausgeben

Debug-Level

Format: **debug_level=<-1|0|1|2>**

Beispiel: **debug_level=0**

Diese Option legt fest, wieviel (falls überhaupt) Debugging-Informationen in die Debug-Datei geschrieben werden. Addieren Sie die Werte, um mehrere Arten von Informationen zu erhalten.

Mögliche Werte sind:

- -1 = Alles
- 0 = Nichts
- 1 = Prozessinformationen
- 2 = SQL-Abfragen

Debug-Ausführlichkeit

Format: **debug_verbosity=<0|1|2>**

Beispiel: **debug_verbosity=1**

Diese Option legt fest, wie ausführlich die Debug-Protokollausgaben sein werden.

Mögliche Werte sind:

- 0 = Kurze Ausgaben
- 1 = Detailliertere Ausgaben
- 2 = Sehr detailliert

Name der Debug-Datei

Format: **debug_file=<file name>**

Beispiel: **debug_file=/usr/local/icinga/var/idomod.debug**

Diese Option legt fest, an welche Stelle der Daemon die Debugging-Informationen schreiben soll.

Maximale Größe der Debug-Datei

Format: **max_debug_file_size=<bytes>**

Beispiel: **max_debug_file_size=100000000**

Diese Option legt die maximale Größe (in Bytes) der Debug-Protokolldatei fest. Falls die Datei diese Größe überschreitet, wird sie in eine Datei mit der Endung .old umbenannt. Eine bereits bestehende Datei mit dieser Endung wird automatisch gelöscht. Dies soll dazu beitragen, dass die Dateisystemauslastung während des Debugging nicht außer Kontrolle gerät.

IDO2DB Konfigurationsoptionen

Sperrdatei

Format: **lock_file=<file name>**

Beispiel: **lock_file=id02db.lock**

Dies ist die Sperrdatei, die IDO2DB benutzen wird, um die PID-Nummer zu speichern, wenn es im Daemon-Modus läuft.

IDO2DB-Benutzer/Gruppe

Format: **ido2db_user=<user name>**
ido2db_group=<group name>

Beispiel: **ido2db_user=icinga ido2db_group=icinga-cmd**

Diese Optionen legen Benutzer/Gruppe fest, mit denen der Daemon läuft. Sie können für jede Option jeweils eine Zahl (uid/gid) oder einen Namen angeben.

Socket-Typ

Format: **socket_type=<unix | tcp>**
Beispiel: **socket_type=unix**

Diese Option legt fest, welchen Socket-Typ der Daemon anlegt, um Verbindungen zu akzeptieren.

Socket-Name

Format: **socket_name=<file name>**
Beispiel: **socket_name=/usr/local/icinga/rw/ido.sock**

Diese Option legt den Namen und Pfad des UNIX-Domain-Socket fest, den der Daemon anlegt, um Verbindungen zu akzeptieren. Diese Option ist nur gültig, wenn der Socket-Typ den Wert "unix" hat.

TCP-Port

Format: **tcp_port=<n>**
Beispiel: **tcp_port=5668**

Diese Option legt fest, mit welchem Port sich das Modul verbindet, um Ausgaben zu senden. Diese Option ist nur gültig, wenn der Socket-Typ den Wert "tcp" hat.

Verschlüsselung benutzen

Format: **use_ssl=**
Beispiel: **use_ssl=0**

Diese Option legt fest, ob das Modul SSL benutzen wird, um den Netzwerkverkehr zwischen Modul und ido2db-Daemon zu verschlüsseln. Beide Seiten müssen dieses Feature aktivieren, das von SSL-Bibliotheken wie openssl oder Kerberos abhängt.

Diese Option ist nur gültig, wenn als Ausgabetype "tcp" angegeben wurde.

Datenbank-Server-Typ

Format: **db_servletype=<type>**

Beispiel: **db_servletype=mysql**

Diese Option legt fest, mit welchem Typ von DB-Server sich der Daemon verbinden soll. Unterstützte Datenbanken sind MySQL, PostgreSQL und Oracle

Mögliche Werte sind:

- mysql (benutzt libdbi)
- pqsql (benutzt libdbi)
- oracle (benutzt ocilib)

Datenbank-Host

Format: **db_host=<host name | ip address>**

Beispiel: **db_host=localhost**

Diese Option legt fest, auf welchem Host der DB-Server läuft. **Note:** Oracle wird diese Einstellung ignorieren.

Datenbank-Port

Format: **db_port=<n>**

Beispiel: **db_port=3306**

Diese Option gibt den Port an, auf dem der DB-Server läuft.

Übliche Werte sind:

- 3306 (Default MySQL-Port)
- 5432 (Default PostgreSQL-Port)
- 1521 (Default Oracle-Port)



Anmerkung

Die ocilib wird diesen Wert ignorieren, Sie müssen daher die `tnsnames.ora` anpassen.

Datenbank-Socket

Format: **db_socket=<file name>**

Beispiel: **db_socket=/var/lib/mysql/mysql.sock**

Die Direktive "db_socket" erlaubt es, eine abweichende Socket-Position anzugeben. Diese wird an libdbi-MySQL als mysql_unix_socket weitergegeben, während PostgreSQL den Port überschreibt und ocilib-Oracle diese Einstellung ignoriert.



Anmerkung

Diese Einstellung überschreibt db_port und macht den Wert dadurch nutzlos!

Datenbankname

Format: **db_name=<name>**

Beispiel: **db_name=icinga**

Diese Option gibt den Namen der Datenbank an, die benutzt werden soll.



Anmerkung

Oracle mit ocilib setzt voraus, dass tnsnames.ora mit Host, Port und Datenbankinformation gefüllt ist. Dann können Sie eins der folgenden benutzen:

- //DBSERVER/SID
- SID

Datenbank-Präfix

Format: **db_prefix=<name>**

Beispiel: **db_prefix=icinga_**

Legt den Präfix fest (falls überhaupt), der Tabellennamen vorangestellt werden soll. Wenn Sie den Tabellenpräfix ändert, dann müssen Sie auch das SQL-Skript zur Erzeugung der Datenbank anpassen!



Anmerkung

Oracle wird diesen Präfix ignorieren, weil die Länge von Tabellennamen auf 30 Stellen begrenzt ist.

Datenbank-Benutzer/Passwort

db_user=<user name>

Format:

db_pass=<password>

Beispiel: **db_user=icinga db_pass=icinga**

Dies sind Benutzername/Passwort, die für die Authentifizierung an der DB verwendet werden. Der Benutzer benötigt mindestens SELECT-, INSERT-, UPDATE- und DELETE-Privilegien auf der Datenbank.

Debug level

Format: **debug_level=<-1|0|1|2>**

Beispiel: **debug_level=0**

Diese Option legt fest, wieviel (falls überhaupt) Debugging-Informatioen in die Debug-Datei geschrieben werden. Addieren Sie die Werte, um mehrere Arten von Informationen zu erhalten.

Mögliche Werte sind:

- -1 = Alles
- 0 = Nichts
- 1 = Prozessinformationen
- 2 = SQL-Abfragen

Debug verbosity

Format: **debug_verbosity=<0|1|2>**

Beispiel: **debug_verbosity=1**

Diese Option legt fest, wie ausführlich die Debug-Protokollausgaben sein werden.

Mögliche Werte sind:

- 0 = Kurze Ausgaben
- 1 = Detailliertere Ausgaben
- 2 = Sehr detailliert

Debug file name

Format: **debug_file=<file name>**

Beispiel: **debug_file=/usr/local/icinga/var/idomod.debug**

Diese Option legt fest, an welche Stelle der Daemon die Debugging-Informationen schreiben soll.

Maximum debug file size

Format: **max_debug_file_size=<bytes>**

Beispiel: **max_debug_file_size=100000000**

Diese Option legt die maximale Größe (in Bytes) der Debug-Protokolldatei fest. Falls die Datei diese Größe überschreitet, wird sie in eine Datei mit der Endung .old umbenannt. Eine bereits bestehende Datei mit dieser Endung wird automatisch gelöscht. Dies soll dazu beitragen, dass die Dateisystemauslastung während des Debugging nicht außer Kontrolle gerät.

Lesbarer Debug-Zeitstempel

Format: **debug_readable_timestamp=<0|1>**

Beispiel: **debug_readable_timestamp=0**

Diese Option erlaubt Ihnen, einen (menschlich) lesbaren statt des Standard-Unix-Zeitstempels auszugeben.

Mögliche Werte sind:

- 0 = deaktiviert (Unix-Zeitstempel)
- 1 = aktiviert (menschlich lesbarer Zeitstempel)

OCI-Fehler nach Syslog

Format: **oci_errors_to_syslog=<0|1>**

Beispiel: **oci_errors_to_syslog=0**

IDO2DB registriert eine Fehlerbehandlungsroutine in ocilib, die per Default alle Meldungen in die Debug-Protokolldatei und ins Syslog schreibt. Beim Wert 0 wird die Ausgabe ins Syslog deaktiviert, so dass die Meldungen nur in die Debug-Protokolldatei ausgegeben werden (falls debug_level entsprechend gesetzt ist).

Oracle-Trace-Level

Format: **oracle_trace_level=<0|1|4|8|12>**

Beispiel: **oracle_trace_level=0**

Diese Einstellung aktiviert einen Oracle-Session-Trace für jede ido2db-Verbindung mit Hilfe von trace event. Der Wert muss einen der momenten unterstützten Werte (1,4,8,12) oder 0 für ausgeschaltet. Dies erfordert explizit das "alter session"-Privileg, Select-Rechte auf v\$session und v\$process werden empfohlen.

Mögliche Werte sind:

- 0 = Pseudo level TRACE OFF
- 1 = Standard SQL-Trace, keine Wait-Events oder Bind-Variablen
- 4 = Nur Bind-Variablen
- 8 = Nur Wait-Events
- 12 = Bind-Variables und Wait-Events

SLA-Datensammlung aktivieren

Format: **enable_sla=<0|1>**

Beispiel: **enable_sla=0**

Diese Einstellung steuert das Sammeln von SLA-Daten in der slahistory-Tabelle.

Mögliche Werte sind:

- 0 = deaktiviert (keine Sammlung)
- 1 = aktiviert



Anmerkung

Diese Option ist verfügbar ab Icinga 1.6

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Komponenten](#)

[Zum Anfang](#)

[Beispielkonfigurationen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Beispielkonfigurationen

[Zurück](#)
[Kapitel 12. IDOUtils](#)
[Weiter](#)

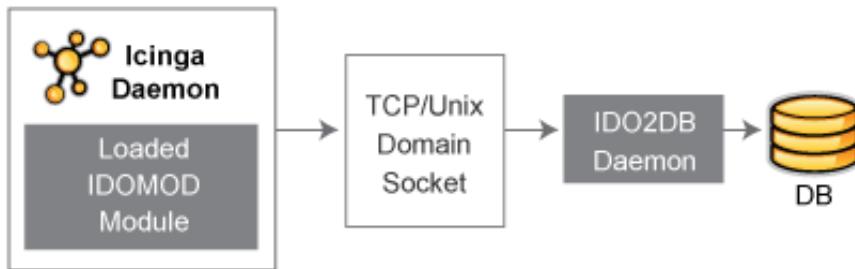
Beispielkonfigurationen

Einzelner Server, einzelne Instanz

Die einfachste Konfiguration tritt auf, wenn in Ihrem Netzwerk eine einzelne Icinga-Instanz läuft. In diesem Fall ist die Installation und Konfiguration der einzelnen Komponenten ziemlich geradeaus.

Das folgende Schaubild zeigt, wie die einzelnen Komponenten in diesem Einzelserver, Einzel-Icinga-Instanz-Aufbau zusammenspielen...

Abbildung 12.11. Einzelserver, Einzelinstanz



Hier eine Beschreibung, was an jedem Punkt des Schaubilds passiert:

1. Das IDOMOD-Modul wird mit einem Instanznamen "default" konfiguriert, weil es lediglich eine Icinga-Instanz im Netzwerk gibt.
2. Während der Icinga-Daemon läuft und die üblichen Aufgaben zur Überwachung des Netzwerks ausführt, sendet das IDOMOD-Modul Konfigurationsdaten und Ereignisinformationen an den TCP- oder Unix-Domain-Socket, der durch den IDO2DB-Daemon angelegt wurde.
3. Der IDO2DB-Daemon liest die Daten, die durch den Socket vom IDOMOD-Modul kommen.
4. Der IDO2DB-Daemon verarbeitet und überträgt die Daten, die vom IDOMOD-Modul empfanden wurden.
5. Die verarbeiteten Daten werden für spätere Abfragen und Verarbeitung in einer Datenbank gespeichert.

Dieses Beispiel setzt voraus, dass:

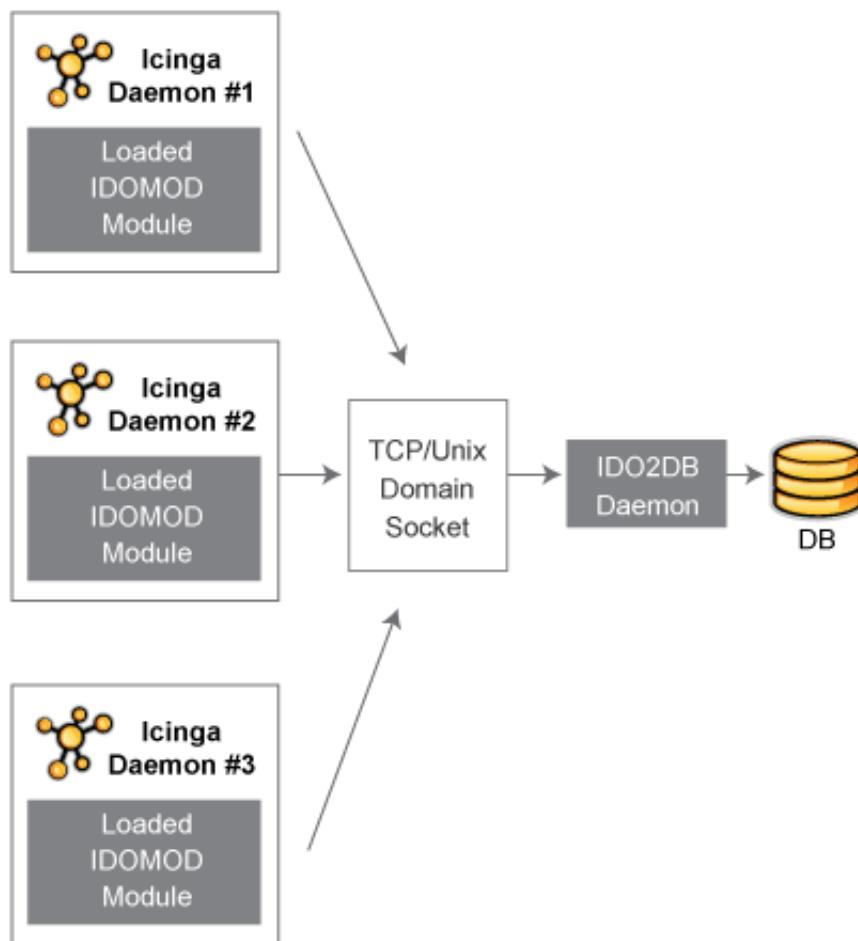
1. Icinga konfiguriert wurde, um das IDOMOD-Modul beim Start zu laden.
2. Der IDO2DB-Daemon läuft (der ein separater, vom Icinga-Daemon unabhängiger, Prozess ist).

Einzelner Server, mehrere Instanzen

Eine weitere einfache Konfiguration kann genutzt werden, wenn Sie mehrere Icinga-Instanzen haben, die auf einem einzigen Server laufen. Installation und Konfiguration der verschiedenen Komponenten des IDOUtils-Addons ist ähnlich zum vorigen Beispiel.

Das folgende Schaubild zeigt, wie die verschiedenen Komponenten in diesem "einzelner Server, mehrere Icinga-Instanzen"-Aufbau zusammenspielen...

Abbildung 12.12. Einzelner Server, mehrere Instanzen



Sie werden bemerken, dass das obige Schaubild ähnlich zu dem "einzelner Server, einzelne Instanz"-Aufbau ist. Der Hauptunterschied besteht darin, dass es nun drei (3) verschiedene Icinga-Daemons statt eines einzelnen gibt.

1. Jeder Icinga-Daemon lädt das IDOMOD-Modul beim Start mit einem eindeutigen Instanznamen. In diesem Beispiel werden die Instanzen einfach "Icinga1", "Icinga2" und "Icinga3" benannt.

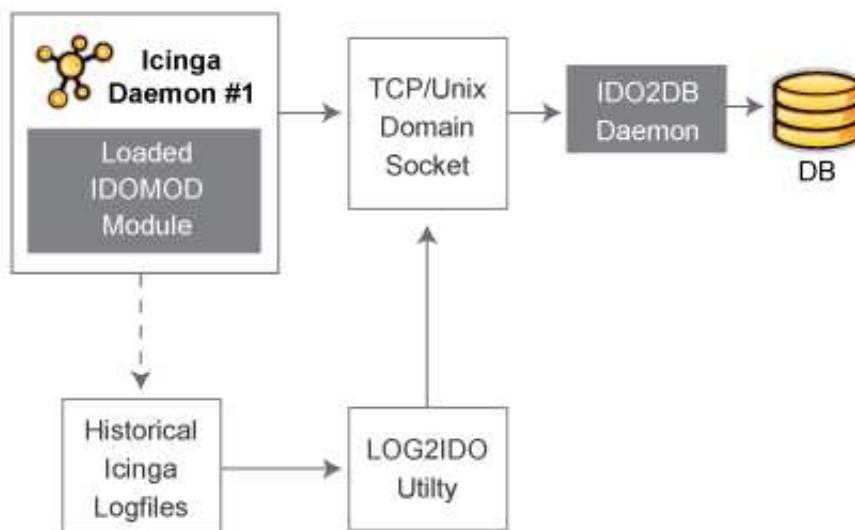
2. Jedes IDOMOD-Modul sendet Konfigurationsdaten und Ereignisinformationen seiner Instanz des Icinga-Daemons an den TCP- oder Unix-Domain-Sockets, der vom IDO2DB-Daemon angelegt wurde.
3. Der IDO2DB-Daemon liest die Daten, die durch den Socket von den drei IDOMOD-Modulen kommen.
4. Der IDO2DB-Daemon verarbeitet und überträgt die Daten, die von den IDOMOD-Modulen empfangen wurden.
5. Die verarbeiteten Daten werden für spätere Abfragen und Verarbeitung in einer Datenbank gespeichert. Die Daten jeder Icinga-Instanz werden (mit Hilfe des eindeutigen Instanznamens) in der Datenbank getrennt voneinander gehalten.

Einzelner Server, einzelne Instanz, Log-Datei-Import

Es gibt zwei Gründe, warum Sie vielleicht Ihre Icinga-Log-Dateien in die gleiche Datenbank importieren möchte, die Icinga-Konfigurations- und Ereignisdaten enthält:

1. Historische Log-Datei-Daten werden nicht automatisch in die Datenbank importiert und möglicherweise ist es wünschenswert, Einträge von Ereignissen zu haben, die vor der Implementierung des IDOUtils-Addon eintraten.
2. Das IDOMOD-Modul ist nicht in der Lage, Echtzeit-Log-Einträge von direkt nach dem Start des Icinga-Daemon bis zum Zeitpunkt des Ladens des IDOMOD-Moduls durch den Icinga-Daemon zu verarbeiten. Diese "Blackout-Periode" ist unvermeidbar und führt zu Log-Einträgen wie "Icinga 1.0 starting...", die das IDOMOD-Modul nicht mitbekommt. Daher wird das Importieren der Logdateien des vorangegangenen Tages auf täglicher Basis (über einen cron-Job) empfohlen.

Abbildung 12.13. Einzelner Server, einzelne Instanz, Log-Datei-Import



Hier eine Beschreibung, was an jedem Punkt des Schaubilds passiert:

1. Historische Icinga-Log-Dateien werden vom LOG2IDO-Utility gelesen.
2. Das LOG2IDO-Utility verarbeitet den Inhalt der Log-Dateien und versieht sie mit dem Instanznamen "default". Dieser Instanzname muss mit dem Instanznamen übereinstimmen, der vom IDOMOD-Modul im Icinga-Daemon verwendet wird.

3. Historische Log-Datei-Daten werden in einem Format an den TCP- oder Unix-Domain-Socket geschickt, das der IDO2DB-Daemon verstehen kann.
4. Der IDO2DB-Daemon liest die Log-Datei-Daten vom TCP- oder Unix-Domain-Socket.
5. Der IDO2DB-Daemon verarbeitet die Log-Datei-Daten.
6. Historische Log-Datei-Daten werden für spätere Abfragen und Verarbeitung in einer Datenbank gespeichert. Der IDO2DB-Daemon wird einige Prüfungen durchführen, um das mehrfache Importieren der gleichen historischen Log-Einträge zu verhindern, so dass der Aufruf des LOG2IDO-Utilitys mit den gleichen Dateien keine negativen Auswirkungen haben sollte.

Das war's! Ziemlich einfach.

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Konfiguration der IDOUtils](#)

[Zum Anfang](#)

[IDOUtils Database Model](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



IDOUtils Database Model

[Zurück](#)

[Kapitel 12. IDOUtils](#)

[Weiter](#)

IDOUtils Database Model

This documentation is based on the NDOUtils database model documentation by Ethan Galstad and altered to fit the changes necessary in IDOUtils.

Introduction

This documentation is still in flux, and there are undoubtably errors present, so take everything you find here with a grain of salt. If you have suggestions, changes, etc. for the documentation, please let us know.



Wichtig

Although we took care to be accurate there may be alterations in the current code which aren't reflected here so if in doubt please take a look at the actual code.

Table Names

The IDOUtils addon allows users to specify a custom prefix to each table name in the database. By default, this prefix is set to "icinga_" in ido2db.cfg. The tables documented here are listed without any prefix.



Anmerkung

Due to limitations in Oracle the length of table names cannot exceed 30 characters so

- The name of one table has been shortened: serviceescalation_contactgroups -> serviceescalationcontactgroups
- The table prefix is ignored

Keys

Every table has a primary key (designated as "PK"). Most tables have a unique key consisting of one ("UK") or more columns ("UKn" whereas n shows the position in the key). Some tables have a non-unique key ("NK") which may be composed of several columns as well ("NKn").

There are a lot of tables containing different information so the description is divided into five parts:

- Central Tables
- Debugging Tables
- Historical Tables
- Current Status Tables
- Configuration Tables

Central Tables

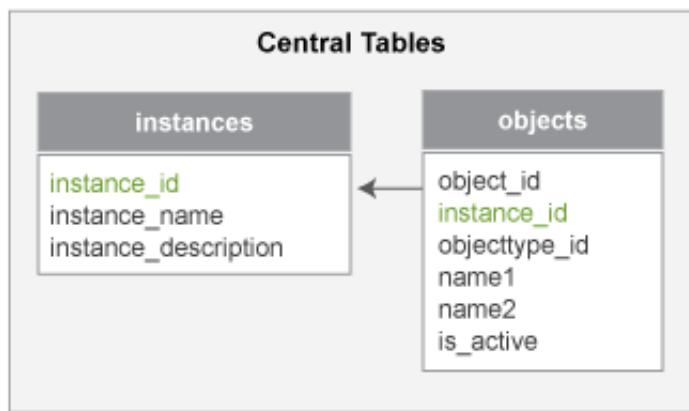
There are two "core" or "central" tables, described below, that are referenced by nearly every table in the database. Read below for more information.

Table List

- [instances](#)
- [objects](#)

Relationship Diagram

Abbildung 12.14. Relationship of Central Tables



Instance Table

Description: This table is needed to ensure that multiple instances of Icinga can store their configuration and status information in the same database. Each instance represents a different Icinga installation/process. A new instance will automatically be created when the user specifies a new instance name (when running one of the IDOUtils components) that does not already exist in the database.

Structure:

Field	Type	Notes	Key
instance_id	SERIAL	Unique number identifying a distinct instance of Icinga	PK
instance_name	VARCHAR(64)	Instance name, as passed to and used by IDOUtils components	
instance_description	VARCHAR(128)	Optional text describing the instance in more detail	

Objects Table

Description: This table is used to store all current (and past) objects that are (and have been) defined in your Icinga configuration files. Why are the names of the objects stored in this table and not elsewhere? Well, when you delete an object definition from your Icinga configuration, that object will no longer appear in the object tables of the database. Since you're still going to want to be able to run reports for old hosts, service, etc., we store the name of the object here so you're not completely baffled by the reports you get. :-)

Structure:

Field	Type	Notes	Values	Key
object_id	SERIAL	A unique number identifying the object		PK
instance_id	BIGINT unsigned	A number indicating the instance of Icinga to which the object belongs		
objecttype_id	BIGINT unsigned	A number indicating what type of object this is	1 = Host; 2 = Service; 3 = Host group; 4 = Service group; 5 = Host escalation; 6 = Service escalation; 7 = Host dependency; 8 = Service dependency; 9 = Timeperiod; 10 = Contact; 11 = Contact group; 12 = Command; 13 = Extended host info (deprecated); 14 = Extended service info (deprecated)	NK1
name1	VARCHAR(128)	The first name associated with the object definition, as used in your Icinga configuration files		NK2
name2	VARCHAR(128)	The second name (if any) associated with the object definition, as used in your Icinga configuration files. This field is only used for service definitions which have a host name (name1 field) and service description (name2 field)		NK3
is_active	SMALLINT	A number indicating whether or not the object is currently defined in your Icinga configuration files. If an object definition is removed from your Icinga configuration files, it will remain in this table, but will be marked as inactive	0 = Inactive; 1 = Active	

Relationships:

Field	Foreign Key
instance_id	instances.instance_id

Debugging Tables

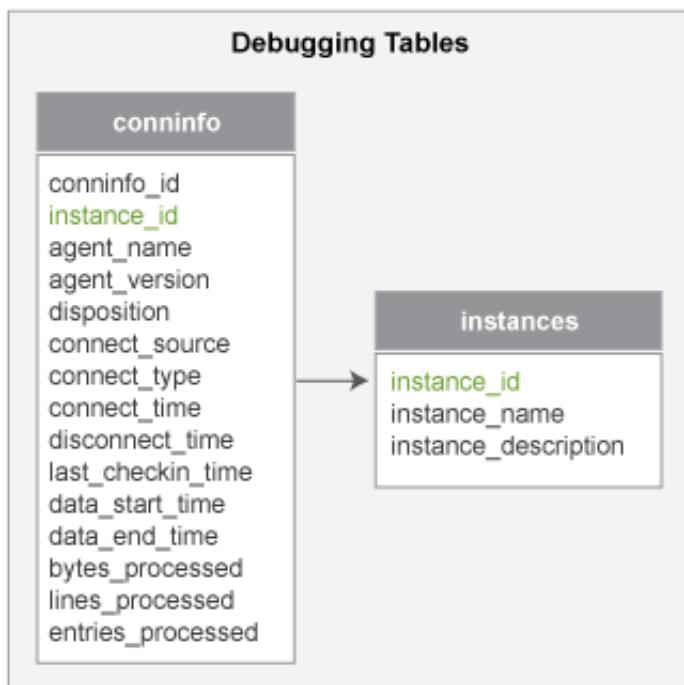
There is currently only one table in the database that is used to hold information that might be useful for debugging purposes. Read below for more information.

Table List

- [conninfo](#)

Relationship Diagram

Abbildung 12.15. Relationship of Debugging Tables



Conninfo Table

Description: This table is used to store debugging information regarding the IDO2DB daemon and the user agents (e.g. LOG2DB, IDOMOD NEB module, etc.) that connect to it. This information is probably only interesting if you are attempting to debug connection problems.

Structure:

Field	Type	Notes	Values	Key
conninfo_id	SERIAL	Unique number identifying the connection info record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga for which data is being transmitted/processed		

agent_name	VARCHAR(32)	Text string identifying the user agent that is sending data to the IDO2DB daemon	Typically "IDOMOD" or "LOG2IDO"	
agent_version	VARCHAR(8)	Text string identifying the version of the user agent that is sending data		
disposition	VARCHAR(16)	Text string identifying the disposition or type of data that is being sent to the IDO2DB daemon	"REALTIME" if being sent directly from a running Icinga process or "ARCHIVED" if being sent from a flat file	
connect_source	VARCHAR(16)	Text string identifying the method that the user agent is using to connect to the IDO2DB daemon	"TCP SOCKET" or "UNIXSOCKET"	
connect_type	VARCHAR(16)	Text string identifying whether this connect was a new connection, or if it was a reconnect due to an earlier communications failure between the user agent and the IDO2DB daemon	"INITIAL" or "RECONNECT"	
connect_time	TIMESTAMP	The initial time the user agent connected to the daemon		
disconnect_time	TIMESTAMP	The time (if any) the user agent disconnect from the daemon		
last_checkin_time	TIMESTAMP	The time that the user agent last checked in with the daemon to indicate that it was still alive and sending data		
data_start_time	TIMESTAMP	The timestamp of the first data that the user agent sent to the daemon		
data_end_time	TIMESTAMP	The timestamp of the last (or latest) data that the user agent sent to the daemon		
bytes_processed	BIGINT unsigned	The number of bytes of data that have been sent by the user agent and processed by the daemon		

lines_processed	BIGINT unsigned	The number of lines of data that have been sent by the user agent and processed by the daemon		
entries_processed	BIGINT unsigned	The number of data entries that have been sent by the user agent and processed by the daemon		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id

Historical Tables

Historical Data Tables

There are several tables in the database which are used to hold "historical" information about Icinga and the hosts/services it is monitoring or was monitoring at some point in the past. Keep in mind that historical items may not necessarily be "old" - they could have occurred 5 seconds ago, so the information used within these tables could/should be used when reporting current status information. Links to hosts/services which no longer exist in the Icinga configuration are maintained due to references for these previous objects existing in the objects table - this is by design.

Table List

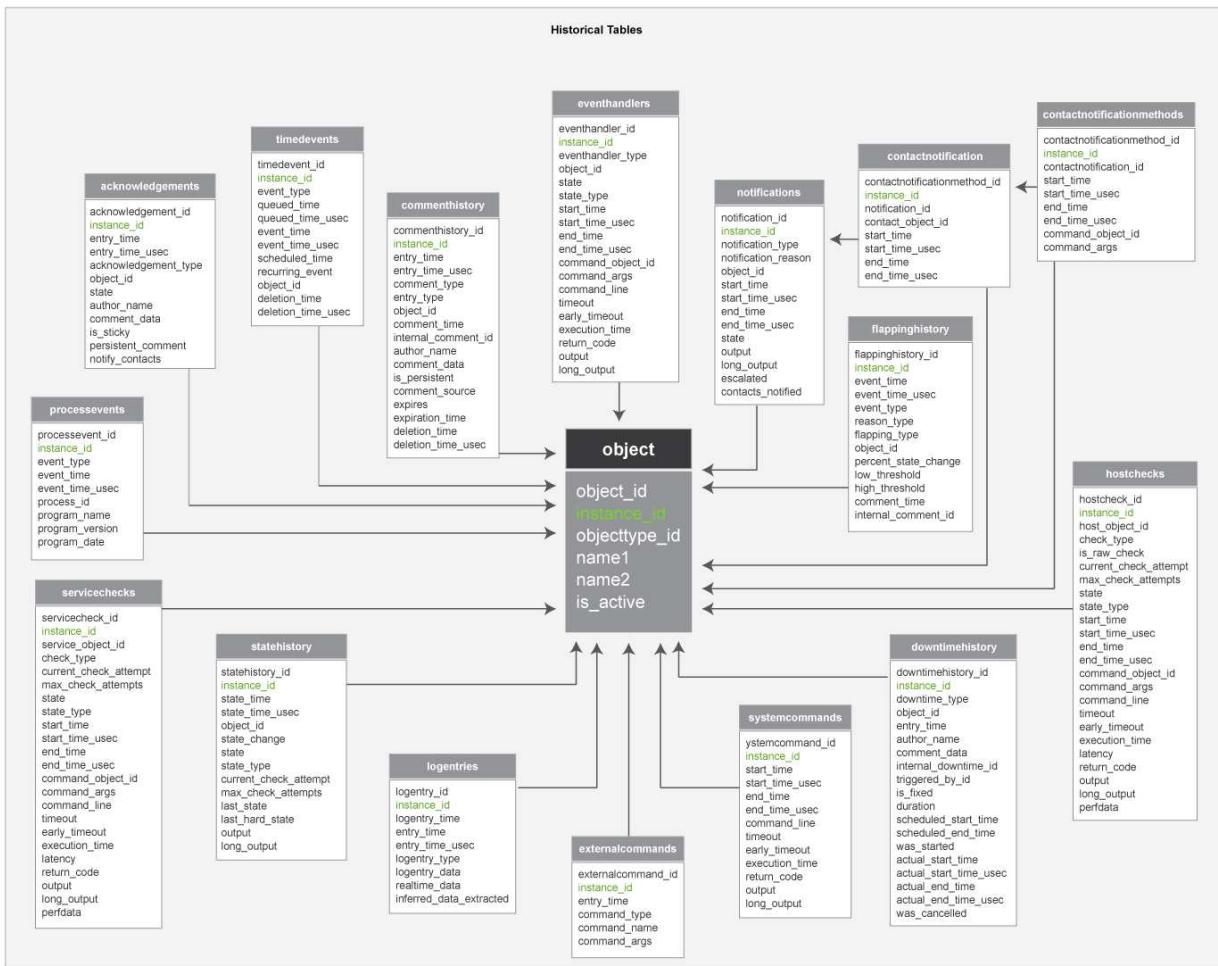
- [acknowledgements](#)
- [commenthistory](#)
- [contactnotifications](#)
- [dbversion](#)
- [downtimehistory](#)
- [eventhandlers](#)
- [externalcommands](#)
- [flappinghistory](#)
- [hostchecks](#)
- [logentries](#)
- [notifications](#)
- [processevents](#)

- [servicechecks](#)
- [statehistory](#)
- [systemcommands](#)
- [timedevents](#)
- [slahistory](#)

Relationship Diagram

Notes: For clarity, the instances table (to which all these tables are related) is not shown. There are 17 historical tables, so please excuse the mess. :-)

Abbildung 12.16. Relationship of Historical Tables



Acknowledgements Table

Table Description: This table is used to store host and service acknowledgements for historical purposes.

Structure:

Field	Type	Notes	Values	Key
acknowledgement_id	SERIAL	Unique number identifying the acknowledgement record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
entry_time	TIMESTAMP	Date and time the acknowledgement was entered		

entry_time_usec	INT	Microsecond portion of acknowledgement entry time		
acknowledgement_type	SMALLINT	Indicates whether this is a host or service acknowledgement	0 = Host ack; 1 = Service ack	
object_id	BIGINT unsigned	The object id of the host or service this acknowledgement applies to		
state	SMALLINT	Integer indicating the state the host or service was in when the acknowledgement was made	Host acks: 0 = UP; 1 = DOWN; 2 = UNREACHABLE; Service acks: 0 = OK; 1 = WARNING; 2 = CRITICAL; 3 = UNKNOWN	
author_name	VARCHAR(64)	Text field containing the name of the person who made the acknowledgement		
comment_data	VARCHAR(255)	Text field containing notes on the acknowledgement		
is_sticky	SMALLINT	Indicates whether or not the acknowledgement is considered "sticky"	0 = Not sticky; 1 = Sticky	
persistent_comment	SMALLINT	Indicates whether or not the comment associated with the acknowledgement is persistent	0 = Not persistent; 1 = Persistent	
notify_contacts	SMALLINT	Indicates whether or not contacts are to be notified of the acknowledgement	0 = Don't notify; 1 = Notify	
end_time	TIMESTAMP	Date and time the acknowledgement was deleted		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

Commenthistory Table

Table Description: This table is used to store historical host and service comments. Current comments will also appear in this table, but it is recommended to use the comments table to retrieve a list of current host and service comments.

Structure:

Field	Type	Notes	Values	Key
commenthistory_id	SERIAL	Unique number identifying the comment record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
entry_time	TIMESTAMP	Date and time the comment was entered		
entry_time_usec	INT	Microsecond portion of comment entry time		
comment_type	SMALLINT	Indicates whether this is a host or service comment	1 = Host comment; 2 = Service comment	
entry_type	SMALLINT	Indicates how this comment came to be entered	1 = User; 2 = Scheduled downtime; 3 = Flapping; 4 = Acknowledgement	
object_id	BIGINT unsigned	The object id of the host or service this acknowledgement applies to		
comment_time	TIMESTAMP	Date and time associated with the comment		UK2
Internal_comment_id	BIGINT unsigned	The comment ID internal to the Icinga daemon, which may no longer be valid or present		UK3

author_name	VARCHAR(64)	Text field containing the name of the person who made the comment		
comment_data	VARCHAR(255)	Text field containing the comment		
is_persistent	SMALLINT	Indicates whether or not the comment is persistent	0 = Not persistent; 1 = Persistent	
comment_source	SMALLINT	Indicates the source of the comment	0 = Internal (Icinga); 1 = External (user)	
expires	SMALLINT	Indicates whether or not the comment expires	0 = Doesn't expire; 1 = Expires	
expiration_time	TIMESTAMP	Date and time at which the comment expires		
deletion_time	TIMESTAMP	Date and time (if any) when the comment was deleted		
deletion_time_usec	INT	Microsecond time (if any) when the comment was deleted		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

Contactnotifications Table

Description: This table is used to store a historical record of host and service notifications that have been sent out to individual contacts.

Structure:

Field	Type	Notes	Values	Key
contactnotification_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
notification_id	BIGINT unsigned	The id of the notification this record is associated with		
contact_object_id	BIGINT unsigned	The object id of the contact this notification was send to		UK2
start_time	TIMESTAMP	The date/time the notification to this contact was started		UK3
start_time_usec	INT	The microsecond portion of the time the notification started		UK4
end_time	TIMESTAMP	The date/time the notification to this contact ended		
end_time_usec	INT	The microsecond portion of the time the notification ended		

Dbversion Table

Description: This table is used to store version information about the DB.

Structure:

Field	Type	Notes	Values	Key
dbversion_id	SERIAL	Unique number identifying the record		PK
name	VARCHAR(10)	Name of the database		UK
version	VARCHAR(10)	Version number of the database		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
notification_id	notifications.notification_id
contact_object_id	objects.object_id

Contactnotificationmethods Table

Description: This table is used to store a historical record of commands (methods) that were used to contact individuals about host and service problems and recoveries.

Structure:

Field	Type	Notes	Values	Key
contactnotificationmethod_id	INT	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
contactnotification_id	INT	The id of the contact notification this record is associated with		UK2
start_time	TIMESTAMP	The date/time the notification command started		UK3
start_time_usec	INT	The microsecond portion of the time the notification command started		UK4
end_time	TIMESTAMP	The date/time the notification command ended		
end_time_usec	INT	The microsecond portion of the time the notification command ended		
command_object_id	INT	The id of the command that was used for the notification command		
command_args	VARCHAR	The arguments that were passed to the notification command		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
contactnotification_id	contactnotifications.contactnotification_id
command_object_id	objects.object_id

Downtimehistory Table

Description: This table is used to store a historical record of scheduled host and service downtime

Structure:

Field	Type	Notes	Values	Key
downtimehistory_id	SERIAL	Unique number identifying the record		PK

instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
downtime_type	SMALLINT	A number identifying what type of scheduled downtime this is	1 = Service downtime; 2 = Host downtime	
object_id	BIGINT unsigned	The object id of the host or service this scheduled downtime is associated with		UK2
entry_time	TIMESTAMP	The date/time the scheduled downtime was entered/submitted		UK3
author_name	VARCHAR(64)	The name of the person who scheduled this downtime		
comment_data	VARCHAR(255)	A comment, as entered by the author, associated with the scheduled downtime		
internal_downtime_id	BIGINT unsigned	A number (internal to the Icinga daemon) associated with the scheduled downtime		UK4
triggered_by_id	BIGINT unsigned	The id of another scheduled downtime entry that scheduled downtime is optionally triggered by. Non-triggered downtimes will have a value of 0 in this field		
is_fixed	SMALLINT	A number indicating whether or not this scheduled downtime is fixed (i.e. its start and end times are exactly what they are listed below as) or if it is flexible	0 = Flexible (Not fixed); 1 = Fixed	
duration	SMALLINT	The number of seconds that the scheduled downtime should last. This is only used by Icinga if the downtime is flexible. If the downtime is fixed, this value should reflect the difference between the start and end times		
scheduled_start_time	TIMESTAMP	The date/time the scheduled downtime is supposed to start. If this is a flexible (non-fixed) downtime, this refers to the earliest possible time that the downtime can start		

scheduled_end_time	TIMESTAMP	The date/time the scheduled downtime is supposed to end. If this is a flexible (non-fixed) downtime, this refers to the last possible time that the downtime can start		
was_started	SMALLINT	Number indicated whether or not the scheduled downtime was started. Some flexible downtimes may never actually start if the host/service they are associated with never enter a problem state	0 = Was not started; 1 = Was started	
actual_start_time	TIMESTAMP	The date/time the scheduled downtime was actually started (if applicable)		
actual_start_time_usec	INT	Microsecond portion of the actual start time		
actual_end_time	TIMESTAMP	The date/time the scheduled downtime actually ended		
actual_end_time_usec	INT	Microsecond portion of the actual end time		
was_cancelled	SMALLINT	Number indicating whether or not the scheduled downtime was cancelled before it ended normally	0 = Not cancelled; 1 = Cancelled early	

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id
triggered_by_id	[downtimehistory.]downtimehistory_id

Eventhandlers Table

Description: This table is used to store a historical record of host and service event handlers that have been run. NOTE: This table is usually trimmed periodically by the IDO2DB daemon, as it would otherwise grow to an enormous size.

Structure:

Field	Type	Notes	Values	Key
eventhandler_id	SERIAL	Unique number identifying the record		PK

instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
eventhandler_type	SMALLINT	A number indicating what type of event handler this is	0 = Host event handler; 1 = Service event handler; 2 = Global host event handler; 3 = Global service event handler	
object_id	BIGINT unsigned	The object id of the host or service associated with this event handler		UK2
state	SMALLINT	Number indicating the state of host or service when the event handler was run.	For host event handlers: 0 = UP; 1 = DOWN; 2 = UNREACHABLE; For service event handlers: 0 = OK; 1 = WARNING; 2 = CRITICAL; 3 = UNKNOWN	
state_type	SMALLINT	Number indicating the state type of the host or service when the event handler was run	0 = SOFT state; 1 = HARD state	
start_time	TIMESTAMP	The date/time the event handler started		UK3
start_time_usec	INT	The microsecond portion of the time the event handler started		UK4
end_time	TIMESTAMP	The date/time the event handler ended		
end_time_usec	INT	The microsecond portion of the time the event handler ended		
command_object_id	BIGINT unsigned	The id of the command that was run		

command_args	VARCHAR(255)	Arguments to the event handler command that was run		
command_line	VARCHAR(1024)	Fully expanded command line of the event handler that was run		
timeout	SMALLINT	Timeout value in seconds for the event handler		
early_timeout	SMALLINT	Number indicating whether or not the event handler command timed out	0 = Did NOT time out. 1 = Timed out	
execution_time	DOUBLE	Time in seconds that the event handler command was running		
return_code	SMALLINT	The return code value from the event handler command		
output	VARCHAR(255)	The first line of text output (if any) from the event handler command		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id
command_object_id	objects.object_id

Externalcommands Table

Description: This table is used to store a historical record of external commands that have been processed by the Icinga daemon.

Structure:

Field	Type	Notes	Values	Key
externalcommand_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
entry_time	TIMESTAMP	The date/time the external command was processed		
command_type	SMALLINT	A number indicating what type of external command this is. Each external command has its own type or "id"	See Icinga source code	
command_name	VARCHAR(128)	The name of the command that was processed		
command_args	VARCHAR(255)	Optional arguments that were specified with the command.		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id

Flappinghistory Table

Table Description: This table is used to store a historical record of host and service flapping events.

Structure:

Field	Type	Notes	Values	Key
flappinghistory_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
event_time	TIMESTAMP	The date/time of the flapping event		
event_time_usecs	INT	The microsecond portion of the time of the flapping event		
event_type	SMALLINT	The type of flapping event indicated by this record	1000 = Flapping started; 1001 = Flapping stopped	
reason	SMALLINT	Number indicating the reason (if any) that the host or service stopped flapping. This is only valid if this record is a flapping stopped event (see event_type field)	1 = Flapping stopped normally 2 = Flapping was disabled	
flapping_type	SMALLINT	Number indicating whether this flapping event relates to a host or service	0 = Host 1 = Service	
object_id	BIGINT unsigned	The id of the host or service associated with the flapping event		
percent_state_change	DOUBLE	The percent state change of the host or service at the time of the event		
low_threshold	DOUBLE	The low flapping percent state change threshold (as configured in Icinga) of the host or service		
high_threshold	DOUBLE	The high flapping percent state change threshold (as configured in Icinga) of the host or service		
comment_time	TIMESTAMP	The date/time of the comment associated with the flapping event		
internal_comment_id	BIGINT unsigned	The number (internal to the Icinga daemon) of the comment associated with the flapping event.		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

Hostchecks Table

Description: This table is used to store a historical record of "raw" and "processed" host checks. What's the difference between raw and processed host checks? Raw checks are the raw results from a host check command that gets executed. Icinga must do some processing on the raw host check results before it can determine the real state of the host. Host checks (plugins) cannot directly determine whether a host is DOWN or UNREACHABLE - only Icinga can do that. In fact, host checks return the same status codes as service checks (OK, WARNING, UNKNOWN, or CRITICAL). Icinga processes the raw host check result to determine the true state of the host (UP, DOWN, or UNREACHABLE). These "processed" checks contain the the true state of the host. NOTE: This table is usually trimmed periodically by the IDO2DB daemon, as it would otherwise grow to an enormous size.

Structure:

Field	Type	Notes	Values	Key
hostcheck_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
host_object_id	BIGINT unsigned	The id of the host this check applies to		UK2
check_type	SMALLINT	Number indicating whether this is an active or passive check	0 = Active check 1 = Passive check	
is_raw_check	SMALLINT	Number indicating whether this is a "raw" or "processed" host check	0 = Processed check; 1 = Raw check	

current_check_attempt	SMALLINT	Current check attempt of the host		
max_check_attempts	SMALLINT	Max check attempts (as defined in Icinga) for the host		
state	SMALLINT	Current state of the host	For raw checks: 0 = UP 1 = DOWN/UNREACHABLE; For processed checks: 0 = UP 1 = DOWN 2 = UNREACHABLE	
state_type	SMALLINT	Number indicating whether the host is in a soft or hard state	0 = SOFT state 1 = HARD state	
start_time	TIMESTAMP	The date/time the host check was started		UK3
start_time_usec	INT	Microsecond portion of the time the host check was started		UK4
end_time	TIMESTAMP	The date/time the host check was completed		
end_time_usec	INT	Microsecond portion of the time the host check was completed		
command_object_id	BIGINT unsigned	The id of the command that was used to perform the host check		
command_args	VARCHAR(255)	The arguments that were passed to the host check command		

command_line	VARCHAR(1024)	The fully expanded command line that was used to check the host		
timeout	SMALLINT	Number of seconds before the host check command would time out		
early_timeout	SMALLINT	Number indicating whether or not the host check timed out early	0 = Did NOT timeout 1 = Timed out	
execution_time	DOUBLE	Number of seconds it took to execute the host check		

latency	DOUBLE	<p>Number of seconds the host check was "late" in being executed. Scheduled host checks can have a latency, but on-demand checks will have a latency of 0. Latency is the difference between the time the check was scheduled to be executed and the time it was actually executed. For passive checks it is the difference between the timestamp on the passive host check result (submitted through the external command file) and the time the passive check result was processed by Icinga</p>		
return_code	SMALLINT	The return code from the host check command		
output	VARCHAR(255)	Status text output from the host check command (first line)		

long_output	TEXT	Status text output from the host check command (consecutive lines)		
perfdata	TEXT	Optional performance data returned from the host check command.		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
host_object_id	objects.object_id
command_object_id	objects.object_id

Logentries Table

Description: This table is used to store a historical record of entries from the Icinga log.

Structure:

Field	Type	Notes	Values	Key
logentry_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
logentry_time	TIMESTAMP	The date/time associated with the log entry. This is NOT necessarily the same as the date/time that Icinga wrote the log entry to the log file (see below)		
entry_time	TIMESTAMP	The date/time that Icinga wrote this log entry to the log file		
entry_time_usec	INT	The microsecond portion of the time that Icinga wrote this log entry		
logentry_type	INT	A number indicating what general type of log entry this is	See Icinga source code	
logentry_data	VARCHAR(255)	The log entry that was written out to the log file		
realtime_data	SMALLINT	A number used internally by the IDO2DB daemon		
inferred_data_extracted	SMALLINT	A number used internally by the IDO2DB daemon.		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id

Notifications Table

Description: This table is used to store a historical record of host and service notifications that have been sent out. For each notification, one or more contacts receive notification messages. These contact notifications are stored in the contactnotifications table.

Structure:

Field	Type	Notes	Values	Key
notification_id	SERIAL	Unique number identifying the record		PK

instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
notification_type	SMALLINT	Number indicating whether this is a host or service notification	0 = Host notification 1 = Service notification	
notification_reason	SMALLINT	Number indicating the type of or reason for the notification	0 = Normal notification; 1 = Problem acknowledgement; 2 = Flapping started; 3 = Flapping stopped; 4 = Flapping was disabled; 5 = Downtime started; 6 = Downtime ended; 7 = Downtime was cancelled; 99 = Custom notification	
object_id	BIGINT unsigned	The id of the host or service this notification applies to		UK2
start_time	TIMESTAMP	The date/time the notification was started		UK3
start_time_usec	INT	Microsecond portion of the time the notification was started		UK4
end_time	TIMESTAMP	The date/time the notification ended		
end_time_usec	INT	Microsecond portion of the time the notification ended		
state	SMALLINT	Number indicating the state of the host or service when the notification was sent out.	For Host Notifications: 0 = UP; 1 = DOWN; 2 = CRITICAL; For Service Notifications: 0 = OK; 1 = WARNING; 2 = CRITICAL; 3 = UNKNOWN	

output	VARCHAR(255)	The current plugin (text) output of the host or service when the notification was sent out (first line)		
long_output	TEXT	The current plugin (text) output of the host or service when the notification was sent out (consecutive lines)		
escalated	SMALLINT	Number indicating whether or not this notification was escalated or not	0 = NOT escalated; 1 = Escalated	
contacts_notified	SMALLINT	Number of contacts that were notified about the host or service as part of this notification.		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

Processevents Table

Description: This table is used to store a historical record of Icinga process events (program starts, restarts, shutdowns, etc.).

Structure:

Field	Type	Notes	Values	Key
processevent_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
event_type	SMALLINT	Number indicating the type of process event that occurred.	100 = Process start; 101 = Process daemonized; 102 = Process restart; 103 = Process shutdown; 104 = Prelaunch; 105 = Event loop start; 106 = Event loop end	
event_time	TIMESTAMP	The date/time that the event occurred		
event_time_usecs	INT	The microsecond portion of the time the event occurred		
process_id	INT	The current process ID (PID) of the Icinga daemon		
program_name	VARCHAR(16)	"Icinga"		
program_version	VARCHAR(20)	Version of Icinga that is running (e.g. "1.0")		
program_date	VARCHAR(10)	Release date of Icinga		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id

Servicechecks Table

Description: This table is used to store a historical record of service checks that have been performed. NOTE: This table is usually trimmed periodically by the IDO2DB daemon, as it would otherwise grow to an enormous size.

Structure:

Field	Type	Notes	Values	Key
servicecheck_id	SERIAL	Unique number identifying the record		PK

instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
service_object_id	BIGINT unsigned	The id of the service this record refers to		UK2
check_type	SMALLINT	Number indicating whether this was an active or a passive service check	0 = Active check; 1 = Passive check	
current_check_attempt	SMALLINT	Number indicating the current check attempt for the service		
max_check_attempts	SMALLINT	Number indicating the max number of check attempts for the service		
state	SMALLINT	Number indicating the current state of the service	0 = OK 1 = WARNING; 2 = CRITICAL; 3 = UNKNOWN	
state_type	SMALLINT	Number indicating the current state type of the service	0 = SOFT state; 1 = HARD state	
start_time	TIMESTAMP	The date/time the service check was started		UK3
start_time_usec	INT	Microsecond portion of the time the service check was started		UK4
end_time	TIMESTAMP	The date/time the service check ended		
end_time_usec	INT	Microsecond portion of the time the service check ended		
command_object_id	BIGINT unsigned	The id of the command that was run to perform the service check		
command_args	VARCHAR(255)	The arguments passed to the command that was run to perform the service check		
command_line	VARCHAR(1024)	The fully expanded command line that was executed to perform the service check		

timeout	SMALLINT	Number of seconds before the service check command was scheduled to timeout		
early_timeout	SMALLINT	Number indicating whether or not the service check timed out	0 = Did NOT timeout 1 = Timed out	
execution_time	DOUBLE	Number of seconds it took to execute the service check command		
latency	DOUBLE	Number of seconds the service check was "late" in being executed. For active checks this is the difference between the scheduled service check time and the time the check actually occurred. For passive checks this is the difference between the timestamp on the passive check result (submitted through the external command file) and the time the passive check result was picked up by the Icinga daemon for processing		
return_code	SMALLINT	The return code from the service check command		
output	VARCHAR(255)	The status output that was returned from the service check command (first line)		
long_output	TEXT	The status output that was returned from the service check command (consecutive lines)		
perfdata	TEXT	Optional performance data that was returned from the service check command		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
service_object_id	objects.object_id
command_object_id	objects.object_id

Statehistory Table

Description: This table is used to store a historical record of host and service state changes.

Structure:

Field	Type	Notes	Values	Key
statehistory_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
state_time	TIMESTAMP	The date/time that the state change occurred		
state_time_usec	INT	The microsecond portion of the time the state change occurred		
object_id	BIGINT unsigned	The id of the host or service object this state change applies to		
state_change	SMALLINT	Number indicating whether or not a state change occurred for the host or service	0 = No state change; 1 = State change	
state	SMALLINT	Number indicating the current state of the host or service	For Hosts: 0 = UP; 1 = DOWN; 2 = UNREACHABLE; For Services: 0 = OK; 1 = WARNING; 2 = CRITICAL; 3 = UNKNOWN	
state_type	SMALLINT	Number indicating whether the service is in a soft or hard state	0 = SOFT state; 1 = HARD state	

current_check_attempt	SMALLINT	Number indicating the current check attempt for the host or service		
max_check_attempts	SMALLINT	Number indicating the max check attempts (as configured in Icinga) for the host or service		
last_state	SMALLINT	Number indicating the last state (whether hard or soft) of the host or service (if available)	For Hosts: -1 = unavailable; 0 = UP; 1 = DOWN; 2 = UNREACHABLE; For Services: -1 = unavailable; 0 = OK; 1 = WARNING; 2 = CRITICAL; 3 = UNKNOWN	
last_hard_state	SMALLINT	Number indicating the last hard state of the host or service (if available)	For Hosts: -1 = unavailable; 0 = UP; 1 = DOWN; 2 = UNREACHABLE; For Services: -1 unavailable; 0 = OK; 1 = WARNING; 2 = CRITICAL	
output	VARCHAR(255)	The current plugin/status output of the host or service (first line)		
long_output	TEXT	The current plugin/status output of the host or service (consecutive lines)		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

Systemcommands Table

Description: This table is used to store a historical record of system commands that are run by the Icinga daemon. Note that each event handler, notification, OCSP command, etc. requires that Icinga execute a system command. NOTE: This table is usually trimmed periodically by the IDO2DB daemon, as it would otherwise grow to an enormous size.

Structure:

Field	Type	Notes	Values	Key
systemcommand_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
start_time	TIMESTAMP	The date/time the command was executed		UK2
start_time_usec	INT	The microsecond portion of the time the command was executed		UK3
end_time	TIMESTAMP	The date/time the command finished executing		
end_time_usec	INT	The microsecond portion of the time the command finished executing		
command_line	VARCHAR(1024)	Fully expanded command line that was executed		
timeout	SMALLINT	Number of seconds before the command should timeout		
early_timeout	SMALLINT	Number indicating whether or not the command timed out early	0 = Did NOT timeout; 1 = Timed out	
execution_time	DOUBLE	Number of seconds it took to execute the command		
return_code	SMALLINT	Return code of the command		
output	VARCHAR(255)	First line of text output (if available) that was returned from the command		
long_output	TEXT	Consecutive lines of text output (if available) that were returned from the command		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id

Timedevents Table

Description: This table is used to store a historical record of timed events that the Icinga process handled. Timed events are internal to the Icinga daemon and used to initiate service checks, host checks, status file updates, etc. They are at the heart of what Icinga does and how it operates. NOTE: This table is usually trimmed periodically by the IDO2DB daemon, as it would otherwise grow to an enormous size.

Structure:

Field	Type	Notes	Values	Key
systemcommand_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
event_type	SMALLINT	Number indicating the type of event that was run	See Icinga source code	UK2
queued_time	TIMESTAMP	The date/time the event was added to the event queue		
queued_time_usec	INT	Microsecond portion of the time the event was added to the event queue		
event_time	TIMESTAMP	The date/time the event was handled		
event_time_usec	INT	Microsecond portion of the time the event was handled		
scheduled_time	TIMESTAMP	The date/time the event was scheduled to be handled/run		UK3
recurring_event	SMALLINT	Number indicating whether or not the event is a recurring one or a one-time event	0 = One-time event; 1 = Recurring event	
object_id	BIGINT unsigned	The id of the host or service that the event applies to. Not all events apply to hosts or services - in these cases the value of this field will be 0.		UK4
deletion_time	TIMESTAMP	The date/time the event was deleted/removed from the event queue		
deletion_time_usec	INT	Microsecond portion of the time the event was removed from the event queue		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

SLA-History Table

Description: This table is used to keep track of host/service availability statistics in a separate history table. A configuration flag (enable_sla) is used to enable/disable SLA data collection; it defaults to 0 (disabled).



Anmerkung

This table is introduced with Icinga 1.6.

Structure:

Field	Type	Notes	Values	Key
slahistory_id	SERIAL	unique ID for a SLA history entry		PK
instance_id	BIGINT unsigned	Icinga instance ID		
start_time	TIMESTAMP	Time interval for the history entry (default '0000-00-00 00:00:00')		
end_time	TIMESTAMP	Time interval for the history entry (default '0000-00-00 00:00:00')		
acknowledgement_time	TIMESTAMP	Time the state was acknowledged (default NULL)		
object_id	BIGINT unsigned	The host/service ID this history entry is related to (default 0)		
state	SMALLINT	Number indicating the state of host or service	For hosts: 0 = UP; 1 = DOWN; 2 = UNREACHABLE; For services: 0 = OK; 1 = WARNING; 2 = CRITICAL; 3 = UNKNOWN	
state_type	SMALLINT	Number indicating the state type of the host or service	0 = SOFT state; 1 = HARD state	
scheduled_downtime	TINYINT(1)	whether the whole time interval was in scheduled downtime (a history entry are split into several entries if only a fraction of the time interval was in scheduled downtime)		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

Current Status Tables

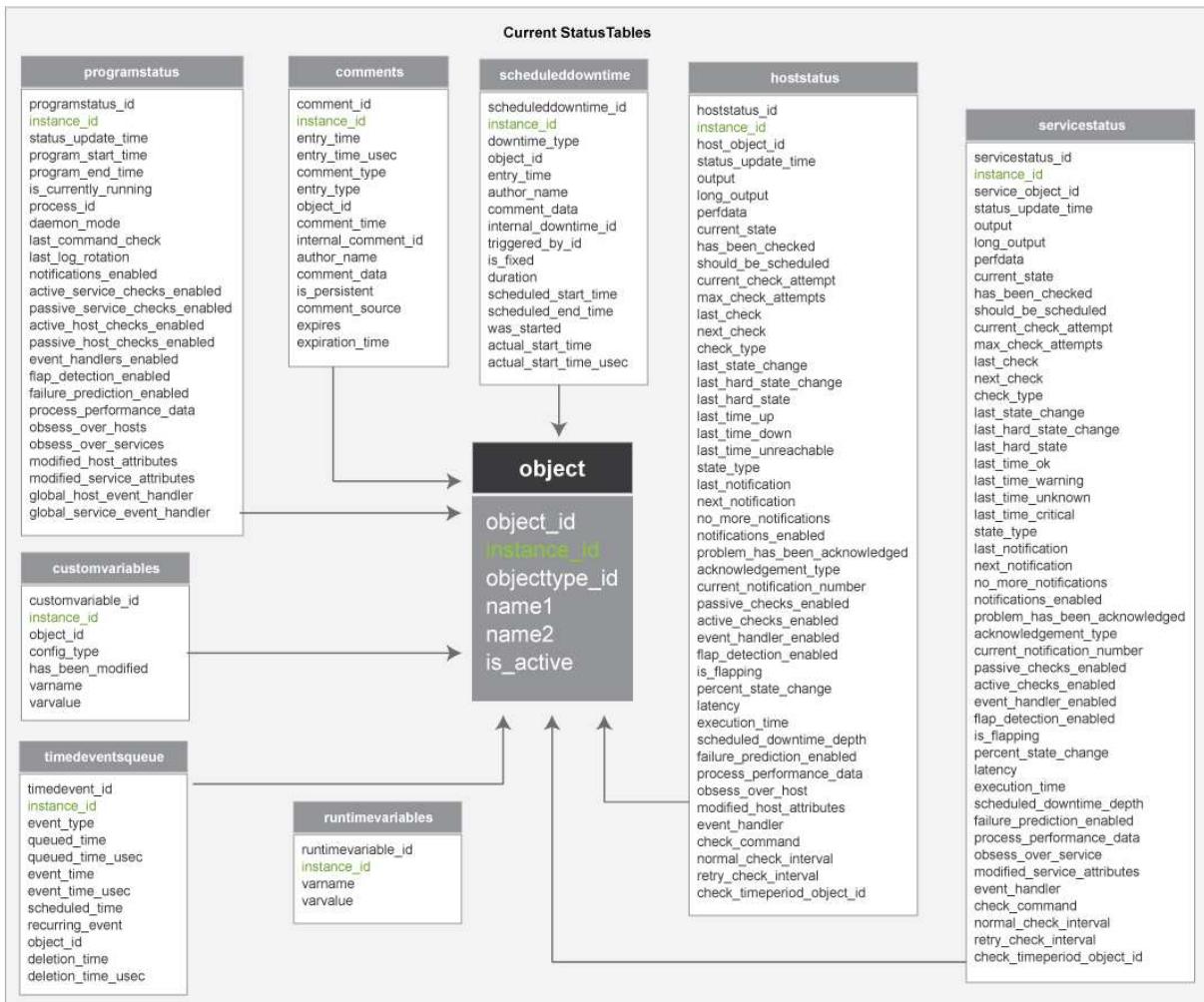
There are several tables in the database which are used to hold current status information on the Icinga process and all hosts and services that it is monitoring. Entries in these tables are cleared whenever the Icinga daemon (belonging to the same instance) (re)starts

Table List

- [comments](#)
- [customvariablestatus](#)
- [hoststatus](#)
- [programstatus](#)
- [runtimevariables](#)
- [scheduleddowntime](#)
- [servicestatus](#)
- [contactstatus](#)
- [timedeventqueue](#)

Relationship Diagram Notes: To reduce clutter, the links to the instances table (to which all these tables are related) is not shown.

Abbildung 12.17. Relationship of Current Status Tables



Comments Table

Description: This table is used to store current host and service comments. Historical comments can be found in the commenthistory table.

Structure:

Field	Type	Notes	Values	Key
comment_id	SERIAL	Unique number identifying the comment record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
entry_time	TIMESTAMP	Date and time the comment was entered		
entry_time_usec	INT	Microsecond portion of comment entry time		

comment_type	SMALLINT	Indicates whether this is a host or service comment	1 = Host comment; 2 = Service comment	
entry_type	SMALLINT	Indicates how this comment came to be entered	1 = User; 2 = Scheduled downtime; 3 = Flapping; 4 = Acknowledgement	
object_id	BIGINT unsigned	The object id of the host or service this acknowledgement applies to		
comment_time	TIMESTAMP	Date and time associated with the comment		UK2
internal_comment_id	INT	The comment ID internal to the Icinga daemon		UK3
author_name	VARCHAR(64)	Text field containing the name of the person who made the comment		
comment_data	VARCHAR(255)	Text field containing the comment		
is_persistent	SMALLINT	Indicates whether or not the comment is persistent	0 = Not persistent; 1 = Persistent	
comment_source	SMALLINT	Indicates the source of the comment	0 = internal (Icinga); 1 = External (user)	
expires	SMALLINT	Indicates whether or not the comment expires	0 = Doesn't expire; 1 = Expires	
expiration_time	TIMESTAMP	Date and time at which the comment expires.		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

Customvariablestatus Table

Description: This table is used to store the current state/values of all custom host, service, and contact variables. Custom variables are only support in Icinga or Nagios 3.x and higher, so this table will be empty for Nagios 2.x.

Structure:

Field	Type	Notes	Values	Key
customvariablestatus_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
object_id	BIGINT unsigned	The object id of the host or service this acknowledgement applies to		UK1
status_update_time	TIMESTAMP	Date and time the status of the custom variable was last updated		
has_been_modified	SMALLINT	Indicates whether the value of the custom variable has been modified (during runtime) from its original value in the config files	0 = Has not been modified; 1 = Has been modified	
varname	VARCHAR(255)	Text field containing the name of the custom variable		UK2, NK
varvalue	VARCHAR(255)	Text field containing the value of the custom variable		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

Hoststatus Table

Description: This table is used to store the current status of hosts that are being monitored.

Structure:

Field	Type	Notes	Values	Key
hoststatus_id	SERIAL	Unique number identifying the record		PK

instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
host_object_id	BIGINT unsigned	The object id of the host this status entry is associated with		U1
status_update_time	TIMESTAMP	Date and time the status data was updated		
output	VARCHAR(255)	Plugin output from the latest host check (first line)		
long_output	TEXT	Plugin output from the latest host check (consecutive lines)		
perfdata	TEXT	Performance data from the latest host check		
current_state	SMALLINT	Number indicating the current state of the host	0 = UP; 1 = DOWN; 2 = UNREACHABLE	
has Been checked	SMALLINT	Number indicating whether or not the host has been checked yet	0 = Not checked; 1 = Checked	
should_be_scheduled	SMALLINT	Number indicating whether or not checks should be regularly scheduled for this host	0 = Not scheduled; 1 = Scheduled	
current_check_attempt	SMALLINT	Number indicating the current check attempt of the host. This is only interesting during soft host states		
max_check_attempts	SMALLINT	Number indicating how many maximum check attempts will be made to determine the hard state of the host		

last_check	TIMESTAMP	Time the host was last checked		
next_check	TIMESTAMP	The host is scheduled to be checked next. Will be set to the epoch if the host is not scheduled for another check		
check_type	SMALLINT	Number indicating if the last host check was an active or passive check	0 = Active; 1 = Passive	
last_state_change	TIMESTAMP	Time the host last had a hard or soft state change. Will be set to the epoch if the host has not changed state		
last_hard_state_change	TIMESTAMP	The host last had a hard state change. Will be setup to the epoch if the host has not changed state		
last_time_up	TIMESTAMP	Time the host was last in an UP state (if ever)		
last_time_down	TIMESTAMP	Time the host was last in a DOWN state (if ever)		
last_time_unreachable	TIMESTAMP	Time the host was last in an UNREACHABLE state (if ever)		
state_type	SMALLINT	Number indicating the type of state the host is in	0 = SOFT state; 1 = HARD state	
last_notification	TIMESTAMP	Time a notification was last sent out for the host (if ever)		
next_notification	TIMESTAMP	Next possible time that a notification can be sent out for the host		

no_more_notifications	SMALLINT	Number indicating whether or not more notifications can be sent out about the current host problem	0 = Send notifications; 1 = Do not send notifications	
notifications_enabled	SMALLINT	Number indicating whether or not notifications are enabled for this host	0 = Notifications disabled; 1 = Notifications enabled	
problem_has_been_acknowledged	SMALLINT	Number indicating whether or not the current host problem has been acknowledged	0 = Not acknowledged; 1 = Acknowledged	
acknowledgement_type	SMALLINT	Number indicating the type of acknowledgement associated with the host	0 = None; 1 = Normal; 2 = Sticky	
current_notification_number	SMALLINT	Number indicating the current notification number for the current host problem. This number gets reset to 0 when the host recovers		
passive_checks_enabled	SMALLINT	Number indicating whether or not passive checks are enabled for this host	0 = Disabled; 1 = Enabled	
active_checks_enabled	SMALLINT	Number indicating whether or not active checks are enabled for this host	0 = Disabled; 1 = Enabled	
event_handler_enabled	SMALLINT	Number indicating whether or not the host's event handler is enabled	0 = Disabled; 1 = Enabled	
flap_detection_enabled	SMALLINT	Number indicating whether or not flap detection is enabled for this host	0 = Disabled; 1 = Enabled	

is_flapping	SMALLINT	Number indicating whether or not the host is currently flapping	0 = Not flapping; 1 = Flapping	
percent_state_change	DOUBLE	Number indicating the current percent state change (a measure of stability/volatility) for the host		
latency	DOUBLE	Number of seconds that the host check was "late" in being executed. The difference between the checks scheduled time and the time it was actually checked		
execution_time	DOUBLE	Number of seconds it took to perform the last check of the host		
scheduled_downtime_depth	SMALLINT	Number indicating how many periods of scheduled downtime are currently active for this host; >0 = In scheduled downtime	0 = Not in scheduled downtime downtime are currently active for this host; >0 = In scheduled downtime	
failure_prediction_enabled	SMALLINT	Number indicating whether or not failure prediction (not yet implemented) is enabled for this host	0 = Disabled; 1 = Enabled	
process_performance_data	SMALLINT	Number indicating whether or not performance data should be processed for this host	0 = Disabled; 1 = Enabled	
obsess_over_host	SMALLINT	Number indicating whether or not this host should be obsessed over	0 = Do not obsess; 1 = Obsess	

modified_host_attributes	INT	Number indicating which attributes of the host have been modified during runtime. Used by the retention data routines		
event_handler	VARCHAR	The current event handler command associated with the host		
check_command	VARCHAR	The current check command associated with the host		
check_interval	DOUBLE	Number of seconds between normal checks of the host		
retry_interval	DOUBLE	Number of seconds between retry checks of the host		
check_timeperiod_object_id	BIGINT unsigned	Unique number of the timeperiod object currently used for determining times the host can be checked		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
host_object_id	objects.object_id
timeperiod_object_id	objects.object_id

Programstatus Table

Description: This table stored status information on the currently (or previously) running Icinga process/daemon.

Structure:

Field	Type	Notes	Values	Key
programstatus_id	SERIAL	Unique number identifying the record		PK

instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		U1
status_update_time	TIMESTAMP	Date and time the status of the process was last updated		
program_start_time	TIMESTAMP	Date and time the Icinga process was started		
program_end_time	TIMESTAMP	Date and time the Icinga process was stopped (if currently not running)		
is_currently_running	SMALLINT	Indicates whether or not the Icinga process is currently running	0 = Process is not running; 1 = Process is running	
process_id	BIGINT unsigned	The processed ID (PID) of the Icinga process		
daemon_mode	SMALLINT	Indicates whether Icinga is running as a foreground process or a daemon	0 = Foreground process; 1 = Daemon	
last_command_check	TIMESTAMP	Date and time the Icinga process last checked external commands		
last_log_rotation	TIMESTAMP	Date and time the log file was last rotated (if at all)		
notifications_enabled	SMALLINT	Indicates whether or not notifications are enabled	0 = Disabled; 1 = Enabled	
active_service_checks_enabled	SMALLINT	Indicates whether or not active service checks are enabled	0 = Disabled; 1 = Enabled	
passive_service_checks_enabled	SMALLINT	Indicates whether or not passive service checks are enabled	0 = Disabled; 1 = Enabled	

active_host_checks_enabled	SMALLINT	Indicates whether or not active host checks are enabled	0 = Disabled; 1 = Enabled	
passive_host_checks_enabled	SMALLINT	Indicates whether or not passive host checks are enabled	0 = Disabled; 1 = Enabled	
event_handlers_enabled	SMALLINT	Indicates whether or not event handlers are enabled	0 = Disabled; 1 = Enabled	
flap_detection_enabled	SMALLINT	Indicates whether or not flap detection is enabled	0 = Disabled; 1 = Enabled	
failure_prediction_enabled	SMALLINT	Indicates whether or not failure prediction is enabled	0 = Disabled; 1 = Enabled	
process_performance_data	SMALLINT	Indicates whether or not performance data is enabled/being processed	0 = Disabled; 1 = Enabled	
obsess_over_hosts	SMALLINT	Indicates whether or not hosts are being obsessed over	0 = Disabled; 1 = Enabled	
obsess_over_services	SMALLINT	Indicates whether or not services are being obsessed over	0 = Disabled; 1 = Enabled	
modified_host_attributes	INT	Indicates what (if any) host-related program status variables have been modified during runtime	See Icinga source code for values	
modified_service_attributes	INT	Indicates what (if any) service-related program status variables have been modified during runtime	See Icinga source code for values	
global_host_event_handler	VARCHAR(255)	Text field indicating the current global host event handler command that is being used.		

global_service_event_handlers	VARCHAR(255)	Text field indicating the current global service event handler command that is being used		
-------------------------------	--------------	---	--	--

Relationships:

Field	Foreign Key
instance_id	instances.instance_id

Runtimevariables Table

Table Description: This table is used to store some runtime variables from the Icinga process that may be useful to you. The only variables currently stored in this table are some initial variables calculated at startup, but more variables may be stored here in future versions.

Structure:

Field	Type	Notes	Values	Key
runtimevariable_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
varname	VARCHAR(64)	Text field containing the name of the variable		UK2
varvalue	VARCHAR(255)	Text field containing the value of the variable		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id

Scheduledowntime Table

Description: This table is used to store current host and service downtime, which may either be current in effect or scheduled to begin at a future time. Historical scheduled downtime information can be found in the downtimehistory table.

Structure:

Field	Type	Notes	Values	Key
scheduledowntime_id	SERIAL	Unique number identifying the record		PK

instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
downtime_type	SMALLINT	Indicates whether this is a host or service downtime entry	1 = Service downtime; 2 = Host downtime	
object_id	BIGINT unsigned	The object id of the host or service this downtime applies to		UK2
entry_time	TIMESTAMP	Date and time this downtime was entered		UK3
author_name	VARCHAR(64)	Text field containing the name of the person who created this scheduled downtime		
comment_data	VARCHAR(255)	Text field containing information about this scheduled downtime (as entered by the user)		
internal_downtime_id	BIGINT unsigned	The ID number (internal to the Icinga daemon) associated with this scheduled downtime entry		UK4
triggered_by_id	BIGINT unsigned	The internal Icinga ID number (if any) of another scheduled downtime entry that this downtime is "triggered" (started) by. If this field is nonzero, this is a triggered downtime entry, otherwise it is not		
is_fixed	SMALLINT	Indicates whether this is a "fixed" scheduled downtime entry (that should start and end at the start and end times indicated) or a "flexible" entry that can start at a variable time	0 = Flexible (not fixed) 1 = Fixed	
duration	SMALLINT	Indicates the number of seconds that the scheduled downtime should last. This is usually only needed if this is "flexible" downtime, which can start at a variable time, but lasts for the specified duration		

scheduled_start_time	TIMESTAMP	Date and time that the downtime is scheduled to start if it is "fixed" downtime. If this is a "flexible" downtime entry, this is the first possible time the downtime can start		
scheduled_end_time	TIMESTAMP	Date and time the downtime is scheduled to end if it is "fixed" downtime. If this is a "flexible" downtime entry, this is the last possible time the downtime can start		
was_started	SMALLINT	Indicates whether or not the downtime was started (is currently #FIXME)	0 = Not started (inactive) 1 = Started (active)	
actual_start_time	TIMESTAMP	Date and time the scheduled downtime was actually started		
actual_start_time_usec	INT	Microsecond portion of time the scheduled downtime was actually started		

Relationships:

Field	Foreign Key	
instance_id	instances.instance_id	
object_id	objects.object_id	

Servicestatus Table

Description: This table is used to store current status information for all services that are being monitored.

Structure:

Field	Type	Notes	Values	Key
servicestatus_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		

service_object_id	BIGINT unsigned	The id of the service this record is associated with		U1
status_update_time	TIMESTAMP	The date/time the status record was updated		
output	VARCHAR(255)	The text output from the most current service check (first line)		
long_output	TEXT	The text output from the most current service check (consecutive lines)		
perfdata	TEXT	Optional performance data from the most current service check		
current_state	SMALLINT	Number indicating the current state of the service	0 = OK; 1 = WARNING; 2 = CRITICAL; 3 = UNKNOWN	
has Been checked	SMALLINT	Number indicating whether or not the service has been checked yet	0 = Has NOT been checked; 1 = Has been checked	
should_be_scheduled	SMALLINT	Number indicating whether or not the service should be scheduled for periodic checks on a regular basis	0 = Not scheduled; 1 = Scheduled	
current_check_attempt	SMALLINT	The current check attempt for the service		
max_check_attempts	SMALLINT	The max check attempts (as configured in Icinga) for the service		

last_check	TIMESTAMP	The date/time the service was last checked. Set to the epoch if the service has not been checked yet		
next_check	TIMESTAMP	The date/time the service is scheduled to be checked next		
check_type	SMALLINT	Number indicating whether or not the last service check was active or passive	0 = Active; 1 = Passive	
last_state_change	TIMESTAMP	The date/time the service last changed state (if at all). This gets updated for both HARD and SOFT state changes		
last_hard_state_change	TIMESTAMP	The date/time the service last changed HARD states (if at all)		
last_hard_state	SMALLINT	0 = OK 1 = WARNING; 2 = CRITICAL; 3 = UNKNOWN		
last_time_ok	TIMESTAMP	The date/time the service was last in an OK state (if at all)		
last_time_warning	TIMESTAMP	The date/time the service was last in a WARNING state (if at all)		
last_time_unknown	TIMESTAMP	The date/time the service was last in an UNKNOWN state (if at all)		
last_time_critical	TIMESTAMP	The date/time the service was last in a CRITICAL state (if at all).		

state_type	SMALLINT	Number indicating whether the service is in a hard or soft state	0 = SOFT state; 1 = HARD	
state_last_notification	TIMESTAMP	The date/time that a notification was last sent out for the current service problem (if applicable)		
next_notification	TIMESTAMP	The earliest date/time that the next notification can be sent out for the current service problem (if applicable)		
no_more_notifications	SMALLINT	Number indicating whether or not future notifications can be sent out for the current service problem	0 = Do not send more notifications; 1 = Keep sending notifications	
notifications_enabled	SMALLINT	Number indicating whether notifications are enabled for the service	0 = Disabled; 1 = Enabled	
problem_has_been_acknowledged	SMALLINT	Number indicating whether or not the current status problem has been acknowledged	0 = Not acknowledged; 1 = Acknowledged	
acknowledgement_type	SMALLINT	Number indicating the type of acknowledgement (if any)	0 = No acknowledgement; 1 = Normal acknowledgement; 2 = Sticky acknowledgement	
current_notification_number	SMALLINT	Number indicating how many notifications have been sent out about the current service problem (if applicable)		

passive_checks_enabled	SMALLINT	Number indicating whether or not passive checks are enabled for the service	0 = Disabled; 1 = Enabled	
active_checks_enabled	SMALLINT	Number indicating whether or not active checks are enabled for the service	0 = Disabled; 1 = Enabled	
event_handler_enabled	SMALLINT	Number indicating whether or not the service event handler is enabled	0 = Disabled; 1 = Enabled	
flap_detection_enabled	SMALLINT	Number indicating whether or not flap detection is enabled for the service	0 = Disabled; 1 = Enabled	
is_flapping	SMALLINT	Number indicating whether or not the service is currently flapping	0 = Not flapping; 1 = Flapping	
percent_state_change	DOUBLE	Number indicating the current percent state change (a measure of volatility) for the service		

latency	DOUBLE	<p>Number indicating how "late" the last service check was in being run. For active checks, this is the difference between the time the service was scheduled to be checked and the time it was actually checked. For passive checks, this is the difference between the timestamp on the passive check (submitted via an external command) and the time Icinga processed the check result.</p> <p>execution_time</p> <p>DOUBLE</p> <p>Number of seconds it took to run the last service check</p>		
execution_time	DOUBLE	<p>Number of seconds it took to execute the service check</p>		
scheduled_downtime_depth	SMALLINT	<p>Number indicating how many periods of scheduled downtime are currently in effect for the service. A value of 0 indicates the service is not in a period of downtime</p>		

failure_prediction_enabled	SMALLINT	Number indicating whether or not failure prediction is enabled for the service. This feature has not yet been implemented	0 = Disabled; 1 = Enabled	
process_performance_data	SMALLINT	Number indicating whether or not performance data should be processed for the service	0 = Do NOT process perfdas; 1 = Process perfdas	
obsess_over_service	SMALLINT	Number indicating whether or not Icinga should obsess of check results of the service	0 = Do NOT obsess; 1 = Obsess	
modified_service_attributes	INT	Number indicating what service attributes have been modified during runtime	See Icinga source code	
event_handler	VARCHAR(255)	The current event handler command that is associated with the service		
check_command	VARCHAR(255)	The current check command that is used to check the status of the service		
normal_check_interval	DOUBLE	The current normal check interval for the service (in seconds)		
retry_check_interval	DOUBLE	The current retry check interval for the service (in seconds)		
check_timeperiod_object_id	BIGINT unsigned	The currently timeperiod that is used to determine when the service can be checked.		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
service_object_id	objects.object_id
check_timeperiod_object_id	objects.object_id

Contactstatus Table

Description: This table is used to store current status information for all contacts that are being used.

Structure:

Field	Type	Notes	Values	Key
contactstatus_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
contact_object_id	BIGINT unsigned	The id of the service this record is associated with		U1
status_update_time	TIMESTAMP	The date/time the status record was updated		
hostnotifications_enabled	SMALLINT	Number indicating whether or not host notifications are enabled for this contact	0 = Notifications disabled; 1 = Notifications enabled	
servicenotifications_enabled	SMALLINT	Number indicating whether or not service notifications are enabled for this contact	0 = Notifications disabled; 1 = Notifications enabled	
last_hostnotification	TIMESTAMP	Time the last host notification was sent		
last_servicenotification	TIMESTAMP	Time the last service notification was sent		
modified_attributes	INT	Default = 0		
modified_host_attributes	INT	Default = 0		
modified_service_attributes	INT	Default = 0		

Relationships:

Field	Foreign Key
instance_id	instances.instance_id

Timedequeue Table

Description: This table is used to store all timed events that are in the Icinga event queue, scheduled to be executed at a future time. Historical timed events can be found in the timedevents table.

Structure:

Field	Type	Notes	Values	Key
timedequeue_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
event_type	SMALLINT	Value indicating the type of event		UK2
queued_time	TIMESTAMP	Date and time the event was originally placed into the timed event queue		
queued_time_usec	INT	Microsecond portion of time the event was queued		
scheduled_time	TIMESTAMP	Date and time the event is scheduled to be executed		UK3
recurring_event	SMALLINT	Indicates whether or not this is a recurring event	0 = Not recurring; 1 = Recurring	
object_id	BIGINT unsigned	The object id of the host, service, contact, etc. that this scheduled event applies to (if applicable). If the event is not associated with any particular object, this field will have a value of zero (0)		UK4

Relationships:

Field	Foreign Key
instance_id	instances.instance_id
object_id	objects.object_id

Configuration Tables



Anmerkung

The tables that contain configuration data have not yet been fully documented.

There are many tables in the database that are used to store Icinga configuration. Note that the data in these tables represents a read-only output view of the configuration that Icinga was using during its last (or current) run. Configuration information from these tables is NOT read by the Icinga daemon in any way, and thus cannot be used to configure Icinga.

Table List

- [commands](#)
- [configfiles](#)
- [configfilevariables](#)
- [contact_addresses](#)
- [contact_notificationcommands](#)
- [contactgroup_members](#)
- [contactgroups](#)
- [contactnotificationmethods](#)
- [contacts](#)
- [customvariables](#)
- [host_contactgroups](#)
- [host_contactgroups](#)
- [host_parenthosts](#)
- [hostdependencies](#)
- [hostescalation_contactgroups](#)
- [hostescalation_contacts](#)
- [hostescalations](#)
- [hostgroup_members](#)
- [hostgroups](#)
- [hosts](#)
- [service_contactgroups](#)
- [service_contacts](#)

- [servicedependencies](#)
- [serviceescalation_contactgroups](#)
- [serviceescalation_contacts](#)
- [serviceescalations](#)
- [servicegroup_members](#)
- [servicegroups](#)
- [services](#)
- [timeperiod_timeranges](#)
- [timeperiods](#)

Abbildung 12.18. Relationship of Configuration Tables

Historical Tables	
hosts	hostdependency_id instance_id config_type host_object_id alias display_name address check_command_object_id check_command_args eventhandler_command_object_id eventhandler_command_args notification_timeperiod_object_id check_timeperiod_object_id failure_prediction_options check_interval retry_interval max_check_attempts first_notification_delay notification_interval notify_on_down notify_on_unreachable notify_on_recovery notify_on_flapping notify_on_downtime stalk_on_up stalk_on_down stalk_on_unreachable flap_detection_enabled flap_detection_on_up flap_detection_on_down flap_detection_on_unreachable low_flap_threshold high_flap_threshold process_performance_data freshness_checks_enabled freshness_threshold passive_checks_enabled event_handler_enabled active_checks_enabled retain_status_information retain_nonstatus_information notifications_enabled obsess_over_host failure_prediction_enabled notes notes_url action_url icon_image icon_image_alt vrm_image statusmap_image have_2d_coords x_2d y_2d have_3d_coords x_3d y_3d z_3d
hostgroup	hostgroup_id instance_id config_type hostgroup_object_id alias
hostgroup_member	contactgroup_member_id instance_id contactgroup_id contact_object_id
host_contactgroup	host_contactgroup_id instance_id host_id contactgroup_object_id
hostdependencies	hostdependency_id instance_id config_type host_object_id dependent_host_object_id dependency_type inherits_parent timeperiod_object_id fail_on_up fail_on_down fail_on_unreachable
hostescalations	hostescalation_id instance_id config_type host_object_id timeperiod_object_id first_notification last_notification notification_interval escalate_on_recovery escalate_on_down escalate_on_unreachable
hostescalation_contactgroups	hostescalation_contactgroup_id instance_id hostescalation_id contactgroup_object_id
host_parenthosts	host_parenthost_id instance_id host_id parent_host_object_id
configfilevariables	configfilevariable_id instance_id configfile_id varname varvalue
contactgroups	contactgroup_id instance_id config_type contactgroup_object_id alias
contactgroup_members	contactgroup_member_id instance_id contactgroup_id contact_object_id
contact_addresses	contact_address_id instance_id contact_id address_number address
timeperiod	timeperiod_id instance_id config_type timeperiod_object_id alias
timeperiod_timeranges	timeperiod_timerange_id instance_id timeperiod_id day start_sec end_sec
object	object_id instance_id objecttype_id name1 name2 is_active
configfiles	configfile_id instance_id configfile_type configfile_path
contacts	contact_id instance_id config_type contact_object_id alias email_address pager_address host_timeperiod_object_id service_timeperiod_object_id host_notifications_enabled service_notifications_enabled can_submit_commands notify_service_recovery notify_service_warning notify_service_unknown notify_service_critical notify_service_flapping notify_service_downtime notify_host_recovery notify_host_down notify_host_unreachable notify_host_flapping notify_host_downtime
customvariables	customvariable_id instance_id object_id config_type has_been_modified varname varvalue
contact_notificationcommands	contact_notificationcommand_id instance_id contact_id notification_type command_object_id command_args
contactnotificationmethods	contactnotificationmethod_id instance_id contactnotification_id start_time start_time_usec end_time end_time_usec command_object_id command_args
services	service_id instance_id config_type host_object_id service_object_id display_name check_command_object_id check_command_args eventhandler_command_object_id eventhandler_command_args notification_timeperiod_object_id check_timeperiod_object_id failure_prediction_options check_interval retry_interval max_check_attempts first_notification_delay notification_interval notify_on_warning notify_on_unknown fail_on_critical
serviceescalations	serviceescalation_id instance_id config_type service_object_id timeperiod_object_id first_notification last_notification notification_interval escalate_on_recovery escalate_on_warning escalate_on_unknown escalate_on_critical
serviceescalation_contactgroups	serviceescalation_contactgroup_id instance_id serviceescalation_id contactgroup_object_id
servicegroups	servicegroup_id instance_id config_type servicegroup_object_id alias
servicegroup_members	servicegroup_member_id instance_id servicegroup_id service_object_id
serviceescalation_contactgroups	serviceescalation_contactgroup_id instance_id serviceescalation_id contactgroup_object_id

Commands Table

Description: .

Structure:

Field	Type	Notes	Values	Key
command_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK3
object_id	BIGINT unsigned			UK2
command_line	VARCHAR(1024)			

Configfiles Table

Description: .

Structure:

Field	Type	Notes	Values	Key
configfile_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
configfile_type	SMALLINT			UK2
configfile_path	VARCHAR(255)			UK3

Configfilevariables Table

Description: .

Structure:

Field	Type	Notes	Values	Key
configfilevariable_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
configfile_id	BIGINT unsigned			
varname	VARCHAR(64)			
varvalue	VARCHAR(1024)			

Contact_addresses Table

Description: .

Structure:

Field	Type	Notes	Values	Key
contact_address_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
contact_id	BIGINT unsigned			UK1
address_number	SMALLINT			UK2
address	VARCHAR(255)			

Contact_notificationcommands Table

Description: .

Structure:

Field	Type	Notes	Values	Key
contact_notificationcommand_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
contact_id	BIGINT unsigned			UK1
notification_type	SMALLINT			UK2
command_object_id	BIGINT unsigned			UK3
command_args	VARCHAR(255)			UK4

Contactgroup_members Table

Description: .

Structure:

Field	Type	Notes	Values	Key
contactgroup_member_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
contactgroup_id	BIGINT unsigned			UK1
contact_object_id	BIGINT unsigned			UK2

Contactgroups Table

Description: .

Structure:

Field	Type	Notes	Values	Key
contactgroup_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
contactgroup_object_id	BIGINT unsigned			UK3
alias	VARCHAR(255)			

Contactnotificationmethods Table

Description: .

Structure:

Field	Type	Notes	Values	Key
contactnotificationmethod_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
contactnotification_id	BIGINT unsigned			UK2
start_time	TIMESTAMP			UK3
start_time_usec	INT			UK4
end_time	TIMESTAMP			
end_time_usec	INT			
command_object_id	BIGINT unsigned			
command_args	VARCHAR(255)			

Contacts Table

Description: .

Structure:

Field	Type	Notes	Values	Key
contact_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
contact_object_id	BIGINT unsigned			UK3
alias	VARCHAR(64)	String describing the contact		
email_address	VARCHAR(255)	String containing the e-mail address of the contact		
pager_address	VARCHAR(64)	String containing the pager address of the contact		

host_timeperiod_object_id	BIGINT unsigned			
service_timeperiod_object_id	BIGINT unsigned			
host_notifications_enabled	SMALLINT	Indicates whether or not the contact will receive host notifications	0 = Disabled; 1 = Enabled	
service_notifications_enabled	SMALLINT	Indicates whether or not the contact will receive service notifications	0 = Disabled; 1 = Enabled	
can_submit_commands	SMALLINT	Indicates whether or not the contact can submit external commands via the web interface	0 = Disabled; 1 = Enabled	
notify_service_recovery	SMALLINT	Indicates whether or not the contact will receive notifications if a service enters the state "RECOVERY"	0 = Disabled; 1 = Enabled	
notify_service_warning	SMALLINT	Indicates whether or not the contact will receive notifications if a service enters the state "WARNING"	0 = Disabled; 1 = Enabled	
notify_service_unknown	SMALLINT	Indicates whether or not the contact will receive notifications if a service enters the state "UNKNOWN"	0 = Disabled; 1 = Enabled	
notify_service_critical	SMALLINT	Indicates whether or not the contact will receive notifications if a service enters the state "CRITICAL"	0 = Disabled; 1 = Enabled	
notify_service_flapping	SMALLINT	Indicates whether or not the contact will receive notifications if a service enters the state "FLAPPING"	0 = Disabled; 1 = Enabled	
notify_service_downtime	SMALLINT	Indicates whether or not the contact will receive notifications if a service enters the state "DOWNTIME"	0 = Disabled; 1 = Enabled	

notify_host_recovery	SMALLINT	Indicates whether or not the contact will receive notifications if a host enters the state "RECOVERY"	0 = Disabled; 1 = Enabled	
notify_host_down	SMALLINT	Indicates whether or not the contact will receive notifications if a host enters the state "DOWN"	0 = Disabled; 1 = Enabled	
notify_host_unreachable	SMALLINT	Indicates whether or not the contact will receive notifications if a host enters the state "UNREACHABLE"	0 = Disabled; 1 = Enabled	
notify_host_flapping	SMALLINT	Indicates whether or not the contact will receive notifications if a host enters the state "FLAPPING"	0 = Disabled; 1 = Enabled	
notify_host_downtime	SMALLINT	Indicates whether or not the contact will receive notifications if a host enters the state "DOWNTIME"	0 = Disabled; 1 = Enabled	

customvariables Table

Description: .

Structure:

Field	Type	Notes	Values	Key
customvariable_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
object_id	BIGINT unsigned			UK1
config_type	SMALLINT			UK2
has Been Modified	SMALLINT			
varname	VARCHAR(255)	String containing the name of the custom variable		UK3,NK
varvalue	VARCHAR(255)	String containing the value of the custom variable		

Host_contactgroups Table

Description: .

Structure:

Field	Type	Notes	Values	Key
host_contactgroup_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
host_id	BIGINT unsigned			UK1
contactgroup_object_id	BIGINT unsigned			UK2

Host_contacts Table

Description: .

Structure:

Field	Type	Notes	Values	Key
host_contact_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
host_id	BIGINT unsigned			UK1
contact_object_id	BIGINT unsigned			UK2

Hostescalation_contactgroups Table

Description: .

Structure:

Field	Type	Notes	Values	Key
hostescalation_contactgroup_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
hostescalation_id	BIGINT unsigned			UK1
contactgroup_object_id	BIGINT unsigned			UK2

Hostescalation_contacts Table

Description: .

Structure:

Field	Type	Notes	Values	Key
hostescalation_contact_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
hostescalation_id	BIGINT unsigned			UK1
contact_object_id	BIGINT unsigned			UK2

Host_parenthosts Table

Description: .

Structure:

Field	Type	Notes	Values	Key
host_parenthost_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
host_id	BIGINT unsigned			UK1
parent_host_object_id	BIGINT unsigned			UK2

Hostdependencies Table

Description: .

Structure:

Field	Type	Notes	Values	Key
hostdependency_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
host_object_id	BIGINT unsigned			UK3
dependent_host_object_id	BIGINT unsigned			UK4
dependency_type	SMALLINT	Indicates the type of the dependency	1 = Notification dependency, 2 = Execution dependency	UK5
inherits_parent	SMALLINT	Indicates whether or not the host will inherit dependencies from parent hosts	0 = do not inherit dependencies, 1 = inherit dependencies	UK6
timeperiod_object_id	BIGINT unsigned			
fail_on_up	SMALLINT	Indicates whether or not the host will be checked if the master host is UP	0 = check host, 1 = do not check host	UK7
fail_on_down	SMALLINT	Indicates whether or not the host will be checked if the master host is DOWN	0 = check host, 1 = do not check host	UK8
fail_on_unreachable	SMALLINT	Indicates whether or not the host will be checked if the master host is UNREACHABLE	0 = check host, 1 = do not check host	UK9

Hostescalations Table

Description: .

Structure:

Field	Type	Notes	Values	Key
hostescalation_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
host_object_id	BIGINT unsigned			UK3
timeperiod_object_id	BIGINT unsigned			UK4
first_notification	SMALLINT			UK5
last_notification	SMALLINT			UK6
notification_interval	DOUBLE			
escalate_on_recovery	SMALLINT			
escalate_on_down	SMALLINT			
escalate_on_unreachable	SMALLINT			

Hostgroup_members Table

Description: .

Structure:

Field	Type	Notes	Values	Key
hostgroup_member_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
hostgroup_id	BIGINT unsigned			UK1
host_object_id	BIGINT unsigned			UK2

Hostgroups Table

Description: .

Structure:

Field	Type	Notes	Values	Key
hostgroup_id	SERIAL			PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			
hostgroup_object_id	BIGINT unsigned			UK2
alias	VARCHAR(255)	String describing the hostgroup		

Hosts Table

Description: .

Structure:

Field	Type	Notes	Values	Key
host_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
host_object_id	BIGINT unsigned			UK3
alias	VARCHAR(64)	String describing the host		
display_name	VARCHAR(255)			
address	VARCHAR(128)	IP-address		
address6	VARCHAR(128)	IPv6-address		
check_command_object_id	BIGINT unsigned			
check_command_args	VARCHAR(255)			
eventhandler_command_object_id	BIGINT unsigned			
eventhandler_command_args	VARCHAR(255)			
notification_timeperiod_object_id	BIGINT unsigned			
check_timeperiod_object_id	BIGINT unsigned			

failure_prediction_options	VARCHAR(64)			
check_interval	DOUBLE			
retry_interval	DOUBLE			
max_check_attempts	SMALLINT			
first_notification_delay	DOUBLE			
notification_interval	DOUBLE			
notify_on_down	SMALLINT			
notify_on_unreachable	SMALLINT			
notify_on_recovery	SMALLINT			
notify_on_flapping	SMALLINT			
notify_on_downtime	SMALLINT			
stalk_on_up	SMALLINT			
stalk_on_down	SMALLINT			
stalk_on_unreachable	SMALLINT			
flap_detection_enabled	SMALLINT			
flap_detection_on_up	SMALLINT			
flap_detection_on_down	SMALLINT			
flap_detection_on_unreachable	SMALLINT			
low_flap_threshold	DOUBLE			
high_flap_threshold	DOUBLE			
process_performance_data	SMALLINT			
freshness_checks_enabled	SMALLINT			
freshness_threshold	SMALLINT			
passive_checks_enabled	SMALLINT			
eventhandler_enabled	SMALLINT			
active_checks_enabled	SMALLINT			
retain_status_information	SMALLINT			
retain_nonstatus_information	SMALLINT			
notifications_enabled	SMALLINT			
obsess_over_host	SMALLINT			
failure_prediction_enabled	SMALLINT			
notes	VARCHAR(255)			
notes_url	VARCHAR(255)			

action_url	VARCHAR(255)			
icon_image	VARCHAR(255)			
icon_image_alt	VARCHAR(255)			
vrm1_image	VARCHAR(255)			
statusmap_image	VARCHAR(255)			
have_2d_coords	SMALLINT			
x_2d	SMALLINT			
y_2d	SMALLINT			
have_3d_coords	SMALLINT			
x_3d	DOUBLE			
y_3d	DOUBLE			
z_3d	DOUBLE			

Service_contactgroups Table

Description: .

Structure:

Field	Type	Notes	Values	Key
service_contactgroup_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
service_id	BIGINT unsigned			UK2
contactgroup_object_id	BIGINT unsigned			UK3

Service_contacts Table

Description: .

Structure:

Field	Type	Notes	Values	Key
service_contact_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
service_id	BIGINT unsigned			UK2
contact_object_id	BIGINT unsigned			UK3

Servicedependencies Table

Description: .

Structure:

Field	Type	Notes	Values	Key
servicedependency_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
service_object_id	BIGINT unsigned			UK3
dependent_service_object_id	BIGINT unsigned			UK4
dependency_type	SMALLINT			UK5
inherits_parent	SMALLINT			UK6
timeperiod_object_id	BIGINT unsigned			
fail_on_ok	SMALLINT			UK7
fail_on_warning	SMALLINT			UK8
fail_on_unknown	SMALLINT			UK9
fail_on_critical	SMALLINT			UK10

Serviceescalation_contactgroups Table

Description: .

Structure:

Field	Type	Notes	Values	Key
serviceescalation_contactgroup_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
serviceescalation_id	BIGINT unsigned			UK1
contactgroup_object_id	BIGINT unsigned			UK2

Serviceescalation_contacts Table

Description: .

Structure:

Field	Type	Notes	Values	Key
serviceescalation_contact_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
serviceescalation_id	BIGINT unsigned			UK1
contact_object_id	BIGINT unsigned			UK2

Serviceescalations Table

Description: .

Structure:

Field	Type	Notes	Values	Key
serviceescalation_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
service_object_id	BIGINT unsigned			UK3
timeperiod_object_id	BIGINT unsigned			UK4
first_notification	SMALLINT			UK5
last_notification	SMALLINT			UK6
notification_interval	DOUBLE			
escalate_on_recovery	SMALLINT			
escalate_on_warning	SMALLINT			
escalate_on_unknown	SMALLINT			
escalate_on_critical	SMALLINT			

Servicegroup_members Table

Description: .

Structure:

Field	Type	Notes	Values	Key
servicegroup_member_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
servicegroup_id	BIGINT unsigned			UK1
service_object_id	BIGINT unsigned			UK2

Servicegroups Table

Description: .

Structure:

Field	Type	Notes	Values	Key
servicegroup_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
servicegroup_object_id	BIGINT unsigned			UK3
alias	VARCHAR(255)	String describing the servicegroup		

Services Table

Description: .

Structure:

Field	Type	Notes	Values	Key
service_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
host_object_id	BIGINT unsigned			
service_object_id	BIGINT unsigned			UK3
display_name	VARCHAR(255)			
check_command_object_id	BIGINT unsigned			
check_command_args	VARCHAR(255)			
eventhandler_command_object_id	BIGINT unsigned			
eventhandler_command_args	VARCHAR(255)			
notification_timeperiod_object_id	BIGINT unsigned			
check_timeperiod_object_id	BIGINT unsigned			
failure_prediction_options	VARCHAR(64)			

check_interval	DOUBLE			
retry_interval	DOUBLE			
max_check_attempts	SMALLINT			
first_notification_delay	DOUBLE			
notification_interval	DOUBLE			
notify_on_warning	SMALLINT			
notify_on_unknown	SMALLINT			
notify_on_critical	SMALLINT			
notify_on_recovery	SMALLINT			
notify_on_flapping	SMALLINT			
notify_on_downtime	SMALLINT			
stalk_on_ok	SMALLINT			
stalk_on_warning	SMALLINT			
stalk_on_unknown	SMALLINT			
stalk_on_critical	SMALLINT			
is_volatile	SMALLINT			
flap_detection_enabled	SMALLINT			
flap_detection_on_ok	SMALLINT			
flap_detection_on_warning	SMALLINT			
flap_detection_on_unknown	SMALLINT			
flap_detection_on_critical	SMALLINT			
low_flap_threshold	DOUBLE			
high_flap_threshold	DOUBLE			
process_performance_data	SMALLINT			
freshness_checks_enabled	SMALLINT			
freshness_threshold	SMALLINT			
passive_checks_enabled	SMALLINT			
eventhandler_enabled	SMALLINT			
active_checks_enabled	SMALLINT			
retain_status_information	SMALLINT			
retain_nonstatus_information	SMALLINT			
notifications_enabled	SMALLINT			
obsess_over_service	SMALLINT			

failure_prediction_enabled	SMALLINT			
notes	VARCHAR(255)			
notes_url	VARCHAR(255)			
action_url	VARCHAR(255)			
icon_image	VARCHAR(255)			
icon_image_alt	VARCHAR(255)			

Timeperiod_timeranges Table

Description: .

Structure:

Field	Type	Notes	Values	Key
timeperiod_timerange_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		
timeperiod_id	BIGINT unsigned			UK1
day	SMALLINT			UK2
start_sec	INT			UK3
end_sec	INT			UK4

Timeperiods Table

Description: .

Structure:

Field	Type	Notes	Values	Key
timeperiod_id	SERIAL	Unique number identifying the record		PK
instance_id	BIGINT unsigned	Unique number identifying the distinct instance of Icinga which this entry is associated with		UK1
config_type	SMALLINT			UK2
timeperiod_object_id	BIGINT unsigned			UK3
alias	VARCHAR(255)	String describing the timeperiod		

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[Beispielkonfigurationen](#)

[Zum Anfang](#)

[Datenbank-Anpassungen/Änderungen](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>



Datenbank-Anpassungen/Änderungen

[Zurück](#)[Kapitel 12. IDOUtils](#)[Weiter](#)

Datenbank-Anpassungen/Änderungen

Ändern des Instance-Namens

Möglicherweise möchten Sie den Instance-Namen ändern. Es gibt einige Schritte, die im folgenden Abschnitt beschrieben sind. Danke an [ralfk](#), der uns diese Anleitung geliefert hat.

- Stoppen Sie Icinga und die ido2db-Daemonen (denn anderenfalls wird statt einer Änderung automatisch ein neuer Instanzname zur Datenbank hinzugefügt)

```
#> /etc/init.d/icinga stop
#> /etc/init.d/ido2db stop
```

- Ändern Sie den Instanznamen in der Datei `/usr/local/icinga/etc/idomod.cfg`

```
instance_name=newinstance
```

- Ändern Sie den Instanznamen in der Datenbanktabelle "icinga_instances" bzw. "instances"

MySQL/PostgreSQL

```
SQL> UPDATE icinga_instances SET instance_name='NEWNAME' WHERE instance_name='OLDNAME';
```

Oracle

```
SQL> UPDATE instances SET instance_name='NEWNAME' WHERE instance_name='OLDNAME';
```

- Ändern Sie den Instanznamen in der command pipe-Konfiguration in einer der folgenden Dateien (Site-Datei an erster Stelle)

- `/usr/local/icinga-web/app/modules/Web/config/icinga-io.xml`
 - `/usr/local/icinga-web/app/modules/Web/config/icinga-io.site.xml`

- Löschen Sie den Web-Cache

```
#> /usr/local/icinga-web/bin/clearcache.sh
```

- Starten Sie Icinga und die ido2db-Daemonen

```
#> /etc/init.d/ido2db start
#> /etc/init.d/icinga start
```

[Zurück](#)

[Nach oben](#)

[Weiter](#)

[IDOUtils Database Model](#)

[Zum Anfang](#)

[Stichwortverzeichnis](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>

**Stichwortverzeichnis**[Zurück](#)**Stichwortverzeichnis****A****Abhängigkeiten**

"harte" Abhängigkeiten, [Host- und Service-Abhängigkeiten](#)

gleicher-Host-Abhängigkeiten, [Zeitsparende Tricks für Objektdefinitionen](#)

gleicher-Host-Abhängigkeiten mit Servicegruppen, [Zeitsparende Tricks für Objektdefinitionen](#)

Host- und Service-Abhängigkeiten, [Host- und Service-Abhängigkeiten](#)

vorausschauende Abhängigkeitsprüfungen, [Vorausschauende Abhängigkeitsprüfungen](#)

accept_passive_host_checks=

akzeptieren von passiven Hostprüfungen, [Optionen der Hauptkonfigurationsdatei](#)

accept_passive_service_checks=

akzeptieren von passiven Service-Prüfungen, [Optionen der Hauptkonfigurationsdatei](#)

action_url_target=

Action-URL-Ziel, [Optionen CGI-Konfigurationsdatei](#)

Adaptive Überwachung, Adaptive Überwachung**additional_freshness_latency=**

zusätzliche Frische-Verschiebung, [Optionen der Hauptkonfigurationsdatei](#)

Addons

Business Process Addon, [Icinga Addons](#)

check_mk, [Icinga Addons](#)

IDOUtils, [Icinga Addons](#)

LConf, [Icinga Addons](#)

Lilac, [Icinga Addons](#)

MKLiveStatus-Integration, [MKLiveStatus-Integration](#)

MultiSite, [Icinga Addons](#)

NagiosQL, [Icinga Addons](#)

NagVis, [Icinga Addons](#)

NConf, [Icinga Addons](#)

NRPE, [Icinga Addons](#)

NSCA, [Icinga Addons](#)

NSClient++, [Icinga Addons](#)

PNP4Nagios, [Icinga Addons](#)

Thruk, [Icinga Addons](#)

add_notif_num_hard=

Service-Zustand und Benachrichtigungsnummer anzeigen (hard), [Optionen](#)

[CGI-Konfigurationsdatei](#)

add_notif_num_soft=
 Service-Zustand und Benachrichtigungsnummer anzeigen (soft), [Optionen CGI-Konfigurationsdatei](#)

admin_email=
 Administrator-e-Mail-Adresse, [Optionen der Hauptkonfigurationsdatei](#)

admin_pager=
 Administrator-Pager, [Optionen der Hauptkonfigurationsdatei](#)

Aktive Prüfungen
 in regelmäßigen Abständen, [Aktive Prüfungen \(Active Checks\)](#)
 nach Bedarf, [Aktive Prüfungen \(Active Checks\)](#)

Aktualisieren (upgrading), [Icinga aktualisieren](#)
 einer früheren Icinga-Version, [Icinga aktualisieren](#)
 von Nagios Version 2.x, [Icinga aktualisieren](#)
 von Nagios Version 3.x, [Icinga aktualisieren](#)

Aktualisierung der Icinga-Web-Datenbank, [Aktualisierung von Icinga-Web und Icinga-Web Datenbank](#)

Aktualisierung von Icinga-Web, [Aktualisierung von Icinga-Web und Icinga-Web Datenbank](#)

allow_empty_hostgroup_assignment=
 Leere Hostgruppenzuordnung erlauben, [Optionen der Hauptkonfigurationsdatei](#)

Anfänger, Hinweise für, [Hinweise für Neulinge](#)

Angepasste CGI-Kopf- und Fußzeilen, [Angepasste CGI-Kopf- und Fußzeilen](#)

API/Icinga, [Installation und Benutzung der Icinga-API](#)

audio_alerts=
 Audio-Alarne, [Optionen CGI-Konfigurationsdatei](#)

Ausfallzeit
 geplante Ausfallzeit, [Geplante Ausfallzeiten](#)

authorized_contactgroup_for_all_hosts=
 Zugang zu globalen Host-Informationen (contactgroup), [Optionen CGI-Konfigurationsdatei](#)

authorized_contactgroup_for_all_services=
 Zugang zu globalen Service-Informationen (contactgroup), [Optionen CGI-Konfigurationsdatei](#)

authorized_contactgroup_for_all_service_commands=
 Zugang zu globalen Service-Befehlen (contactgroup), [Optionen CGI-Konfigurationsdatei](#)

authorized_contactgroup_for_configuration_information=
 Zugang zu Konfigurationsinformationen (contactgroup), [Optionen CGI-Konfigurationsdatei](#)

authorized_contactgroup_for_host_commands=
 Zugang zu globalen Host-Befehlen (contactgroup), [Optionen CGI-Konfigurationsdatei](#)

authorized_contactgroup_for_read_only=
 Zugang zu Befehlen/Kommentaren sperren (contactgroup), [Optionen CGI-Konfigurationsdatei](#)

authorized_contactgroup_for_system_commands=
 Zugang zu System/Prozess-Befehlen (contactgroup), [Optionen CGI-Konfigurationsdatei](#)

authorized_contactgroup_for_system_information=
 Zugang zu System/Prozess-Informationen (contactgroup), [Optionen CGI-Konfigurationsdatei](#)

authorized_for_all_hosts=
 Zugang zu globalen Host-Informationen (contact), [Optionen CGI-Konfigurationsdatei](#)

authorized_for_all_services=
 Zugang zu globalen Service-Informationen (contact), [Optionen CGI-Konfigurationsdatei](#)

authorized_for_all_service_commands=
 Zugang zu globalen Service-Befehlen (contact), [Optionen CGI-Konfigurationsdatei](#)

authorized_for_configuration_information=
 Zugang zu Konfigurationsinformationen (contact), [Optionen CGI-Konfigurationsdatei](#)

authorized_for_full_command_resolution=
 Benutzer(n) VOLLEN Zugriff auf Kommandozeilenanzeige erlauben, [Optionen CGI-Konfigurationsdatei](#)

authorized_for_host_commands=
 Zugang zu globalen Host-Befehlen (contact), [Optionen CGI-Konfigurationsdatei](#)

authorized_for_read_only=
 Zugang zu Befehlen/Kommentaren sperren (contact), [Optionen CGI-Konfigurationsdatei](#)

authorized_for_system_commands=
 Zugang zu System/Prozess-Befehlen (contact), [Optionen CGI-Konfigurationsdatei](#)

authorized_for_system_information=
 Zugang zu System/Prozess-Informationen (contact), [Optionen CGI-Konfigurationsdatei](#)

auto_reschedule_checks=
 Auto-Rescheduling Option, [Optionen der Hauptkonfigurationsdatei](#)

auto_rescheduling_interval=
 Auto-Rescheduling, [Optionen der Hauptkonfigurationsdatei](#)

auto_rescheduling_windows=
 Auto-Rescheduling Window, [Optionen der Hauptkonfigurationsdatei](#)

B

Befehls-Definition (command-Definition), [Befehls-Definition \(command\)](#)
 Benachrichtigungen, [Benachrichtigungen](#)
 Audio-Alarme, [Benachrichtigungen](#)
 Filter, [Benachrichtigungen](#)
 Bereitschaftszeiten-Rotation, [Bereitschafts-Rotation](#)
 broker_module=
 Eventbroker-Module, [Optionen der Hauptkonfigurationsdatei](#)
 buffer_file=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
 Business Process Addon, [Icinga Addons](#)

C

Cached Checks, [Zwischengespeicherte Prüfungen](#)
 cached_host_check_horizon=
 Cached Host Check Horizon, [Optionen der Hauptkonfigurationsdatei](#)
 cached_service_check_horizon=
 Cached Service Check, [Optionen der Hauptkonfigurationsdatei](#)
 cfg_dir=
 Objektkonfigurationsverzeichnis, [Optionen der Hauptkonfigurationsdatei](#)
 cfg_file=
 Objektkonfigurationsdatei, [Optionen der Hauptkonfigurationsdatei](#)
 CGI-Authentifizierung
 Aktivieren der Authentifizierungs/Autorisierungsfunktionalität in den CGIs, [Authentifizierung und Autorisierung in den CGIs](#)
 Authentifizierter Benutzer, [Authentifizierung und Autorisierung in den CGIs](#)
 Authentifizierter Kontakt, [Authentifizierung und Autorisierung in den CGIs](#)
 Authentifizierung auf sicheren Web-Servern, [Authentifizierung und Autorisierung in den CGIs](#)
 Erstellen von authentifizierten Benutzern, [Authentifizierung und Autorisierung in den CGIs](#)

Standardberechtigungen für CGI-Informationen, [Authentifizierung](#) und [Autorsierung](#) in den CGIs

Zusätzliche Berechtigungen zu CGI-Informationen gewähren, [Authentifizierung](#) und [Autorsierung](#) in den CGIs

[CGI-Autorisierung, Authentifizierung und Autorisierung in den CGIs](#)

[CGI-Konfiguration](#)

Optionen der CGI-Konfigurationsdatei, [Optionen CGI-Konfigurationsdatei](#)

[CGI-Parameter](#)

ahas, [Informationen zu den CGI-Parametern](#)

alerttypes, [Informationen zu den CGI-Parametern](#)

archive, [Informationen zu den CGI-Parametern](#)

assumeinitialstates, [Informationen zu den CGI-Parametern](#)

assumestateretention, [Informationen zu den CGI-Parametern](#)

assumestatesduringnotrunning, [Informationen zu den CGI-Parametern](#)

backtrack, [Informationen zu den CGI-Parametern](#)

breakdown, [Informationen zu den CGI-Parametern](#)

broadcast_notification, [Informationen zu den CGI-Parametern](#)

childoptions, [Informationen zu den CGI-Parametern](#)

cmd_mod, [Informationen zu den CGI-Parametern](#)

cmd_typ, [Informationen zu den CGI-Parametern](#)

columns, [Informationen zu den CGI-Parametern](#)

com_author, [Informationen zu den CGI-Parametern](#)

com_data, [Informationen zu den CGI-Parametern](#)

com_id, [Informationen zu den CGI-Parametern](#)

contact, [Informationen zu den CGI-Parametern](#)

createimage, [Informationen zu den CGI-Parametern](#)

csvoutput, [Informationen zu den CGI-Parametern](#)

displaytype, [Informationen zu den CGI-Parametern](#)

down_id, [Informationen zu den CGI-Parametern](#)

eday, [Informationen zu den CGI-Parametern](#)

ehour, [Informationen zu den CGI-Parametern](#)

embedded, [Informationen zu den CGI-Parametern](#)

emin, [Informationen zu den CGI-Parametern](#)

emon, [Informationen zu den CGI-Parametern](#)

end_time, [Informationen zu den CGI-Parametern](#)

esec, [Informationen zu den CGI-Parametern](#)

eyear, [Informationen zu den CGI-Parametern](#)

fixed, [Informationen zu den CGI-Parametern](#)

force_check, [Informationen zu den CGI-Parametern](#)

force_notification, [Informationen zu den CGI-Parametern](#)

full_log_entries, [Informationen zu den CGI-Parametern](#)

get_date_parts, [Informationen zu den CGI-Parametern](#)

graphevents, [Informationen zu den CGI-Parametern](#)

graphstatetypes, [Informationen zu den CGI-Parametern](#)

host, [Informationen zu den CGI-Parametern](#)

hostgroup, [Informationen zu den CGI-Parametern](#)

hostprops, [Informationen zu den CGI-Parametern](#)

hoststates, [Informationen zu den CGI-Parametern](#)

hoststatusatypes, [Informationen zu den CGI-Parametern](#)

hours, [Informationen zu den CGI-Parametern](#)

includesoftstates, [Informationen zu den CGI-Parametern](#)

initialassumedhoststate, [Informationen zu den CGI-Parametern](#)

initialassumedservicestate, [Informationen zu den CGI-Parametern](#)

initialstateslogged, [Informationen zu den CGI-Parametern](#)
input, [Informationen zu den CGI-Parametern](#)
jsonoutput, [Informationen zu den CGI-Parametern](#)
limit, [Informationen zu den CGI-Parametern](#)
minutes, [Informationen zu den CGI-Parametern](#)
navbarsearch, [Informationen zu den CGI-Parametern](#)
newstatesonly, [Informationen zu den CGI-Parametern](#)
nodowntime, [Informationen zu den CGI-Parametern](#)
noflapping, [Informationen zu den CGI-Parametern](#)
nofrills, [Informationen zu den CGI-Parametern](#)
noheader, [Informationen zu den CGI-Parametern](#)
nosystem, [Informationen zu den CGI-Parametern](#)
notimebreaks, [Informationen zu den CGI-Parametern](#)
not_dly, [Informationen zu den CGI-Parametern](#)
oldestfirst, [Informationen zu den CGI-Parametern](#)
performance_data, [Informationen zu den CGI-Parametern](#)
persistent, [Informationen zu den CGI-Parametern](#)
plugin_output, [Informationen zu den CGI-Parametern](#)
plugin_state, [Informationen zu den CGI-Parametern](#)
ptc, [Informationen zu den CGI-Parametern](#)
report, [Informationen zu den CGI-Parametern](#)
report_type, [Informationen zu den CGI-Parametern](#)
rpttimeperiod, [Informationen zu den CGI-Parametern](#)
sched_dly, [Informationen zu den CGI-Parametern](#)
sday, [Informationen zu den CGI-Parametern](#)
send_notification, [Informationen zu den CGI-Parametern](#)
service, [Informationen zu den CGI-Parametern](#)
servicefilter, [Informationen zu den CGI-Parametern](#)
servicegroup, [Informationen zu den CGI-Parametern](#)
serviceprops, [Informationen zu den CGI-Parametern](#)
servicestates,
servicestatustypes, [Informationen zu den CGI-Parametern](#)
service_divisor, [Informationen zu den CGI-Parametern](#)
shour, [Informationen zu den CGI-Parametern](#)
showscheduleddowntime, [Informationen zu den CGI-Parametern](#)
show_log_entries, [Informationen zu den CGI-Parametern](#)
smin, [Informationen zu den CGI-Parametern](#)
smon, [Informationen zu den CGI-Parametern](#)
sortoption, [Informationen zu den CGI-Parametern](#)
sorttype, [Informationen zu den CGI-Parametern](#)
ssecc, [Informationen zu den CGI-Parametern](#)
standardreport, [Informationen zu den CGI-Parametern](#)
start_time, [Informationen zu den CGI-Parametern](#)
statetype, [Informationen zu den CGI-Parametern](#)
statetypes, [Informationen zu den CGI-Parametern](#)
sticky_ack, [Informationen zu den CGI-Parametern](#)
style, [Informationen zu den CGI-Parametern](#)
syear, [Informationen zu den CGI-Parametern](#)
t1, [Informationen zu den CGI-Parametern](#)
t2, [Informationen zu den CGI-Parametern](#)
timeperiod, [Informationen zu den CGI-Parametern](#)
trigger, [Informationen zu den CGI-Parametern](#)
type, [Informationen zu den CGI-Parametern](#)

CGI-Sicherheit

Implementieren der Digest Authentication, Verbesserte CGI-Sicherheit und Authentifizierung

Implementieren erzwungener TLS/SSL-Kommunikation, Verbesserte CGI-Sicherheit und Authentifizierung

Implementieren von IP-Subnetz-Beschränkung, Verbesserte CGI-Sicherheit und Authentifizierung

Verbesserte CGI-Sicherheit und Authentifizierung, Verbesserte CGI-Sicherheit und Authentifizierung

Zusätzliche Techniken, Verbesserte CGI-Sicherheit und Authentifizierung

CGIs

Alert Histogram CGI, [Icinga Classic UI](#): Informationen über die CGIs

Alert History CGI, [Icinga Classic UI](#): Informationen über die CGIs

Alert Summary CGI, [Icinga Classic UI](#): Informationen über die CGIs

Änderungen am Classic UI, [Icinga Classic UI](#): Informationen über die CGIs

Ausführen von CGIs auf der Kommandozeile, Ausführen von CGIs auf der Kommandozeile

Availability Reporting CGI, [Icinga Classic UI](#): Informationen über die CGIs

Command CGI, [Icinga Classic UI](#): Informationen über die CGIs

Configuration CGI, [Icinga Classic UI](#): Informationen über die CGIs

Event log CGI, [Icinga Classic UI](#): Informationen über die CGIs

Extended Information CGI, [Icinga Classic UI](#): Informationen über die CGIs

Filtereigenschaften (filter properties), [Informationen zu den CGI-Parametern](#)

Network Outages CGI, [Icinga Classic UI](#): Informationen über die CGIs

Notification CGI, [Icinga Classic UI](#): Informationen über die CGIs

Parameter, [Informationen zu den CGI-Parametern](#)

Status CGI, [Icinga Classic UI](#): Informationen über die CGIs

Status Map CGI, [Icinga Classic UI](#): Informationen über die CGIs

Statustypen (status types), [Informationen zu den CGI-Parametern](#)

Tactical Overview CGI, [Icinga Classic UI](#): Informationen über die CGIs

Trends CGI, [Icinga Classic UI](#): Informationen über die CGIs

WAP Interface CGI, [Icinga Classic UI](#): Informationen über die CGIs

cgi_log_archive_path=

Pfad des CGI-Protokolldatei-Archivs, [Optionen CGI-Konfigurationsdatei](#)

cgi_log_file=

Name der CGI-Protokolldatei, [Optionen CGI-Konfigurationsdatei](#)

cgi_log_rotation_method=

Rotationsmethode der CGI-Protokolldatei, [Optionen CGI-Konfigurationsdatei](#)

check_external_commands=

prüfen von externen Befehlen, [Optionen der Hauptkonfigurationsdatei](#)

check_for_orphaned_hosts=

verwaiste Host-Prüfungen, [Optionen der Hauptkonfigurationsdatei](#)

check_for_orphaned_services=

verwaiste Service-Prüfungen, [Optionen der Hauptkonfigurationsdatei](#)

check_host_freshness=

Host-Frischeprüfung, [Optionen der Hauptkonfigurationsdatei](#)

check_mk, [Icinga Addons](#)

check_result_path=

Prüfergebnispfad, [Optionen der Hauptkonfigurationsdatei](#)

check_result_reaper_frequency=

Prüfungsergebnisertreffrequenz, [Optionen der Hauptkonfigurationsdatei](#)

check_service_freshness=

Service-Frischeprüfung, [Optionen der Hauptkonfigurationsdatei](#)

child_processes_fork_twice=
 Child Processes Fork, [Optionen der Hauptkonfigurationsdatei](#)
Cluster
 Service- und Host-Gruppen überwachen, Service- und Host-Gruppen überwachen
command_check_interval=
 Prüfintervall externe Befehle, [Optionen der Hauptkonfigurationsdatei](#)
command_file=
 Datei für (externe) Befehle, [Optionen der Hauptkonfigurationsdatei](#)
config_output_options=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)

D

Danksagungen, Über Icinga
 Darstellung von Performance-Informationen mit PNP4Nagios, [Performance-Daten](#)
data_processing_options=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
date_format=
 Datumsformat, [Optionen der Hauptkonfigurationsdatei](#)
db_host=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
db_name=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
db_pass=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
db_port=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
db_prefix=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
db_servtype=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
db_socket=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
db_user=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
debug_file=
 Debug-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
debug_level=
 Debug-Level, [Optionen der Hauptkonfigurationsdatei](#)
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
debug_readable_timestamp=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
debug_verbosity=
 Debug-Ausführlichkeit, [Optionen der Hauptkonfigurationsdatei](#)
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
default_downtime_duration=
 Default-Ausfallzeit-Dauer setzen, [Optionen CGI-Konfigurationsdatei](#)
default_expiring_acknowledgement_duration=
 Setzt die Verfallszeit von Bestätigungen (acknowledgements), [Optionen CGI-Konfigurationsdatei](#)

default_statusmap_layout=
 Default-Statusmap-Layout, [Optionen CGI-Konfigurationsdatei](#)
 default_user_name=
 Standard-Benutzername, [Optionen CGI-Konfigurationsdatei](#)
 Downtime
 geplante Ausfallzeit, [Geplante Ausfallzeiten](#)

E

Embedded Perl
 entwickeln von Plugins für die Benutzung mit Embedded Perl, [Entwickeln von Plugins für die Nutzung mit Embedded Perl](#)
 Embedded Perl Interpreter
 benutzen des ..., [Benutzen des Embedded Perl Interpreters](#)
 enabled_embedded_perl=
 eingebauter Perl-Interpreter, [Optionen der Hauptkonfigurationsdatei](#)
 enable_environment_macros=
 Environment Macros Option, [Optionen der Hauptkonfigurationsdatei](#)
 enable_event_handlers=
 aktivieren von Event-Handlern, [Optionen der Hauptkonfigurationsdatei](#)
 enable_flap_detection=
 Flattererkennung, [Optionen der Hauptkonfigurationsdatei](#)
 enable_notifications=
 Benachrichtigungsoption, [Optionen der Hauptkonfigurationsdatei](#)
 enable_predictive_host_dependency_checks=
 vorausschauende Host-Abhängigkeiten, [Optionen der Hauptkonfigurationsdatei](#)
 enable_predictive_service_dependency_checks=
 vorausschauende Service-Abhängigkeiten, [Optionen der Hauptkonfigurationsdatei](#)
 enable_sla=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
 enable_splunk_integration=
 Splunk-Integrationsoption, [Optionen CGI-Konfigurationsdatei](#)
 enforce_comments_on_actions=
 Erzwingen von Kommentaren bei Aktionen, [Optionen CGI-Konfigurationsdatei](#)
 Erreichbarkeit
 Ermitteln des Zustands und der Erreichbarkeit von Netzwerk-Hosts, [Ermitteln des Zustands und der Erreichbarkeit von Netzwerk-Hosts](#)
 Erweiterte Hostinformations-Definition (extended hostinformation-Definition), [erweiterte Hostinformations-Definition \(hostextinfo\)](#)
 Erweiterte Serviceinformations-Definition (extended serviceinformation-Definition), [erweiterte Serviceinformations-Definition \(serviceextinfo\)](#)
 Escalations
 Escalation condition, [Eskalations-Bedingung](#)
 escape_html_tags=
 Maskieren von HTML-Tags, [Optionen CGI-Konfigurationsdatei](#)
 Eskalationen
 Benachrichtigungseskalationen, [Benachrichtigungseskalationen](#)
 Eventhandler, [Eventhandler](#)
 Beispiel, [Eventhandler](#)
 event_broker_options=
 Eventbroker-Optionen, [Optionen der Hauptkonfigurationsdatei](#)
 event_handler_timeout=
 Eventhandler-Zeitüberschreitung, [Optionen der Hauptkonfigurationsdatei](#)

execute_host_checks= ausführen von Host-Prüfungen, [Optionen der Hauptkonfigurationsdatei](#)

execute_service_checks= ausführen von Service-Prüfungen, [Optionen der Hauptkonfigurationsdatei](#)

external_command_buffer_slots= Buffer slots für externe Befehle, [Optionen der Hauptkonfigurationsdatei](#)

Externe Befehle

- ACKNOWLEDGE_HOST_PROBLEM, [Liste der externen Befehle](#)
- ACKNOWLEDGE_SVC_PROBLEM, [Liste der externen Befehle](#)
- ADD_HOST_COMMENT, [Liste der externen Befehle](#)
- ADD_SVC_COMMENT, [Liste der externen Befehle](#)
- CHANGE_CONTACT_HOST_NOTIFICATION_TIMEPERIOD, [Liste der externen Befehle](#)
- CHANGE_CONTACT_MODATTR, [Liste der externen Befehle](#)
- CHANGE_CONTACT_MODHATTR, [Liste der externen Befehle](#)
- CHANGE_CONTACT_MODSATTR, [Liste der externen Befehle](#)
- CHANGE_CONTACT_SVC_NOTIFICATION_TIMEPERIOD, [Liste der externen Befehle](#)
- CHANGE_CUSTOM_CONTACT_VAR, [Liste der externen Befehle](#)
- CHANGE_CUSTOM_HOST_VAR, [Liste der externen Befehle](#)
- CHANGE_CUSTOM_SVC_VAR, [Liste der externen Befehle](#)
- CHANGE_GLOBAL_HOST_EVENT_HANDLER, [Liste der externen Befehle](#)
- CHANGE_GLOBAL_SVC_EVENT_HANDLER, [Liste der externen Befehle](#)
- CHANGE_HOST_CHECK_COMMAND, [Liste der externen Befehle](#)
- CHANGE_HOST_CHECK_TIMEPERIOD, [Liste der externen Befehle](#)
- CHANGE_HOST_EVENT_HANDLER, [Liste der externen Befehle](#)
- CHANGE_HOST_MODATTR, [Liste der externen Befehle](#)
- CHANGE_MAX_HOST_CHECK_ATTEMPTS, [Liste der externen Befehle](#)
- CHANGE_MAX_SVC_CHECK_ATTEMPTS, [Liste der externen Befehle](#)
- CHANGE_NORMAL_HOST_CHECK_INTERVAL, [Liste der externen Befehle](#)
- CHANGE_NORMAL_SVC_CHECK_INTERVAL, [Liste der externen Befehle](#)
- CHANGE_RETRY_HOST_CHECK_INTERVAL, [Liste der externen Befehle](#)
- CHANGE_RETRY_SVC_CHECK_INTERVAL, [Liste der externen Befehle](#)
- CHANGE_SVC_CHECK_COMMAND, [Liste der externen Befehle](#)
- CHANGE_SVC_CHECK_TIMEPERIOD, [Liste der externen Befehle](#)
- CHANGE_SVC_EVENT_HANDLER, [Liste der externen Befehle](#)
- CHANGE_SVC_MODATTR, [Liste der externen Befehle](#)
- CHANGE_SVC_NOTIFICATION_TIMEPERIOD, [Liste der externen Befehle](#)
- DELAY_HOST_NOTIFICATION, [Liste der externen Befehle](#)
- DELAY_SVC_NOTIFICATION, [Liste der externen Befehle](#)
- DEL_ALL_HOST_COMMENTS, [Liste der externen Befehle](#)
- DEL_ALL_SVC_COMMENTS, [Liste der externen Befehle](#)
- DEL_DOWNTIME_BY_HOSTGROUP_NAME, [Liste der externen Befehle](#)
- DEL_DOWNTIME_BY_HOST_NAME, [Liste der externen Befehle](#)
- DEL_DOWNTIME_BY_START_TIME_COMMENT, [Liste der externen Befehle](#)
- DEL_HOST_COMMENT, [Liste der externen Befehle](#)
- DEL_HOST_DOWNTIME, [Liste der externen Befehle](#)
- DEL_SVC_COMMENT, [Liste der externen Befehle](#)
- DEL_SVC_DOWNTIME, [Liste der externen Befehle](#)
- DISABLE_ALL_NOTIFICATIONS_BEYOND_HOST, [Liste der externen Befehle](#)
- DISABLE_CONTACTGROUP_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
- DISABLE_CONTACTGROUP_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
- DISABLE_CONTACT_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
- DISABLE_CONTACT_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
- DISABLE_EVENT_HANDLERS, [Liste der externen Befehle](#)

DISABLE_FAILURE_PREDICTION, [Liste der externen Befehle](#)
 DISABLE_FLAP_DETECTION, [Liste der externen Befehle](#)
 DISABLE_HOSTGROUP_HOST_CHECKS, [Liste der externen Befehle](#)
 DISABLE_HOSTGROUP_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
 DISABLE_HOSTGROUP_PASSIVE_HOST_CHECKS, [Liste der externen Befehle](#)
 DISABLE_HOSTGROUP_PASSIVE_SVC_CHECKS, [Liste der externen Befehle](#)
 DISABLE_HOSTGROUP_SVC_CHECKS, [Liste der externen Befehle](#)
 DISABLE_HOSTGROUP_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 DISABLE_HOST_AND_CHILD_NOTIFICATIONS, [Liste der externen Befehle](#)
 DISABLE_HOST_CHECK, [Liste der externen Befehle](#)
 DISABLE_HOST_EVENT_HANDLER, [Liste der externen Befehle](#)
 DISABLE_HOST_FLAP_DETECTION, [Liste der externen Befehle](#)
 DISABLE_HOST_FRESHNESS_CHECKS, [Liste der externen Befehle](#)
 DISABLE_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
 DISABLE_HOST_SVC_CHECKS, [Liste der externen Befehle](#)
 DISABLE_HOST_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 DISABLE_NOTIFICATIONS, [Liste der externen Befehle](#)
 DISABLE_PASSIVE_HOST_CHECKS, [Liste der externen Befehle](#)
 DISABLE_PASSIVE_SVC_CHECKS, [Liste der externen Befehle](#)
 DISABLE_PERFORMANCE_DATA, [Liste der externen Befehle](#)
 DISABLE_SERVICEGROUP_HOST_CHECKS, [Liste der externen Befehle](#)
 DISABLE_SERVICEGROUP_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
 DISABLE_SERVICEGROUP_PASSIVE_HOST_CHECKS, [Liste der externen Befehle](#)
 DISABLE_SERVICEGROUP_PASSIVE_SVC_CHECKS, [Liste der externen Befehle](#)
 DISABLE_SERVICEGROUP_SVC_CHECKS, [Liste der externen Befehle](#)
 DISABLE_SERVICEGROUP_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 DISABLE_SERVICE_FLAP_DETECTION, [Liste der externen Befehle](#)
 DISABLE_SERVICE_FRESHNESS_CHECKS, [Liste der externen Befehle](#)
 DISABLE_SVC_CHECK, [Liste der externen Befehle](#)
 DISABLE_SVC_EVENT_HANDLER, [Liste der externen Befehle](#)
 DISABLE_SVC_FLAP_DETECTION, [Liste der externen Befehle](#)
 DISABLE_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_ALL_NOTIFICATIONS_BEYOND_HOST, [Liste der externen Befehle](#)
 ENABLE_CONTACTGROUP_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_CONTACTGROUP_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_CONTACT_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_CONTACT_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_EVENT_HANDLERS, [Liste der externen Befehle](#)
 ENABLE_FAILURE_PREDICTION, [Liste der externen Befehle](#)
 ENABLE_FLAP_DETECTION, [Liste der externen Befehle](#)
 ENABLE_HOSTGROUP_HOST_CHECKS, [Liste der externen Befehle](#)
 ENABLE_HOSTGROUP_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_HOSTGROUP_PASSIVE_HOST_CHECKS, [Liste der externen Befehle](#)
 ENABLE_HOSTGROUP_PASSIVE_SVC_CHECKS, [Liste der externen Befehle](#)
 ENABLE_HOSTGROUP_SVC_CHECKS, [Liste der externen Befehle](#)
 ENABLE_HOSTGROUP_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_HOST_AND_CHILD_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_HOST_CHECK, [Liste der externen Befehle](#)
 ENABLE_HOST_EVENT_HANDLER, [Liste der externen Befehle](#)
 ENABLE_HOST_FLAP_DETECTION, [Liste der externen Befehle](#)
 ENABLE_HOST_FRESHNESS_CHECKS, [Liste der externen Befehle](#)
 ENABLE_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_HOST_SVC_CHECKS, [Liste der externen Befehle](#)

ENABLE_HOST_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_PASSIVE_HOST_CHECKS, [Liste der externen Befehle](#)
 ENABLE_PASSIVE_SVC_CHECKS, [Liste der externen Befehle](#)
 ENABLE_PERFORMANCE_DATA, [Liste der externen Befehle](#)
 ENABLE_SERVICEGROUP_HOST_CHECKS, [Liste der externen Befehle](#)
 ENABLE_SERVICEGROUP_HOST_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_SERVICEGROUP_PASSIVE_HOST_CHECKS, [Liste der externen Befehle](#)
 ENABLE_SERVICEGROUP_PASSIVE_SVC_CHECKS, [Liste der externen Befehle](#)
 ENABLE_SERVICEGROUP_SVC_CHECKS, [Liste der externen Befehle](#)
 ENABLE_SERVICEGROUP_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 ENABLE_SERVICE_FRESHNESS_CHECKS, [Liste der externen Befehle](#)
 ENABLE_SVC_CHECK, [Liste der externen Befehle](#)
 ENABLE_SVC_EVENT_HANDLER, [Liste der externen Befehle](#)
 ENABLE_SVC_FLAP_DETECTION, [Liste der externen Befehle](#)
 ENABLE_SVC_NOTIFICATIONS, [Liste der externen Befehle](#)
 PROCESS_FILE, [Liste der externen Befehle](#)
 PROCESS_HOST_CHECK_RESULT, [Liste der externen Befehle](#)
 PROCESS_SERVICE_CHECK_RESULT, [Liste der externen Befehle](#)
 READ_STATE_INFORMATION, [Liste der externen Befehle](#)
 REMOVE_HOST_ACKNOWLEDGEMENT, [Liste der externen Befehle](#)
 REMOVE_SVC_ACKNOWLEDGEMENT, [Liste der externen Befehle](#)
 RESTART_PROGRAM, [Liste der externen Befehle](#)
 SAVE_STATE_INFORMATION, [Liste der externen Befehle](#)
 SCHEDULE_AND_PROPAGATE_HOST_DOWNTIME, [Liste der externen Befehle](#)
 SCHEDULE_AND_PROPAGATE_TRIGGERED_HOST_DOWNTIME, [Liste der externen Befehle](#)
 SCHEDULE_FORCED_HOST_CHECK, [Liste der externen Befehle](#)
 SCHEDULE_FORCED_HOST_SVC_CHECKS, [Liste der externen Befehle](#)
 SCHEDULE_FORCED_SVC_CHECK, [Liste der externen Befehle](#)
 SCHEDULE_HOSTGROUP_HOST_DOWNTIME, [Liste der externen Befehle](#)
 SCHEDULE_HOSTGROUP_SVC_DOWNTIME, [Liste der externen Befehle](#)
 SCHEDULE_HOST_CHECK, [Liste der externen Befehle](#)
 SCHEDULE_HOST_DOWNTIME, [Liste der externen Befehle](#)
 SCHEDULE_HOST_SVC_CHECKS, [Liste der externen Befehle](#)
 SCHEDULE_HOST_SVC_DOWNTIME, [Liste der externen Befehle](#)
 SCHEDULE_SERVICEGROUP_HOST_DOWNTIME, [Liste der externen Befehle](#)
 SCHEDULE_SERVICEGROUP_SVC_DOWNTIME, [Liste der externen Befehle](#)
 SCHEDULE_SVC_CHECK, [Liste der externen Befehle](#)
 SCHEDULE_SVC_DOWNTIME, [Liste der externen Befehle](#)
 SEND_CUSTOM_HOST_NOTIFICATION, [Liste der externen Befehle](#)
 SEND_CUSTOM_SVC_NOTIFICATION, [Liste der externen Befehle](#)
 SET_HOST_NOTIFICATION_NUMBER, [Liste der externen Befehle](#)
 SET_SVC_NOTIFICATION_NUMBER, [Liste der externen Befehle](#)
 SHUTDOWN_PROGRAM, [Liste der externen Befehle](#)
 START_ACCEPTING_PASSIVE_HOST_CHECKS, [Liste der externen Befehle](#)
 START_ACCEPTING_PASSIVE_SVC_CHECKS, [Liste der externen Befehle](#)
 START_EXECUTING_HOST_CHECKS, [Liste der externen Befehle](#)
 START_EXECUTING_SVC_CHECKS, [Liste der externen Befehle](#)
 START_OBSESSING_OVER_HOST, [Liste der externen Befehle](#)
 START_OBSESSING_OVER_HOST_CHECKS, [Liste der externen Befehle](#)
 START_OBSESSING_OVER_SVC, [Liste der externen Befehle](#)
 START_OBSESSING_OVER_SVC_CHECKS, [Liste der externen Befehle](#)

STOP_ACCEPTING_PASSIVE_HOST_CHECKS, Liste der externen Befehle
 STOP_ACCEPTING_PASSIVE_SVC_CHECKS, [Liste der externen Befehle](#)
 STOP_EXECUTING_HOST_CHECKS, [Liste der externen Befehle](#)
 STOP_EXECUTING_SVC_CHECKS, [Liste der externen Befehle](#)
 STOP_OBSESSING_OVER_HOST, [Liste der externen Befehle](#)
 STOP_OBSESSING_OVER_HOST_CHECKS, [Liste der externen Befehle](#)
 STOP_OBSESSING_OVER_SVC, [Liste der externen Befehle](#)
 STOP_OBSESSING_OVER_SVC_CHECKS, [Liste der externen Befehle](#)
 Externe Befehle (external commands), [Externe Befehle](#), Adaptive Überwachung
 extinfo_show_child_hosts=
 Zeigt Child-Host(s) des gewählten Hosts, [Optionen CGI-Konfigurationsdatei](#)

F

Failover
 Redundante und Failover-Netzwerk-Überwachung, [Redundante und Failover-Netzwerk-Überwachung](#)
 Fast Startup Options, [Schnellstart-Optionen](#)
 file_rotation_command=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
 file_rotation_interval=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
 file_rotation_timeout=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
 first_day_of_week=
 Ersten Tag der Woche festlegen, [Optionen CGI-Konfigurationsdatei](#)
 Flattern (flapping)
 Erkennung und Behandlung von Status-Flattern, [Erkennung und Behandlung von Status-Flattern](#)
 free_child_process_memory=
 Child Process Memory, [Optionen der Hauptkonfigurationsdatei](#)
 Frische (freshness)
 Service- und Host-Frischeprüfungen, [Service- und Host-Frische-Prüfungen](#)

G

Gleicher-Host-Abhängigkeiten, [Zeitsparende Tricks für Objektdefinitionen](#)
 Gleicher-Host-Abhängigkeiten mit Servicegruppen, [Zeitsparende Tricks für Objektdefinitionen](#)
 global_host_event_handler=
 globaler Host-Eventhandler, [Optionen der Hauptkonfigurationsdatei](#)
 global_service_event_handler=
 globaler Service-Eventhandler, [Optionen der Hauptkonfigurationsdatei](#)
 Gnokii, [Benachrichtigungen](#)

H

Herunterladen der letzten Version, [Über Icinga](#)
 highlight_table_rows=
 Zeilen in status.cgi hervorheben, [Optionen CGI-Konfigurationsdatei](#)
 high_host_flap_threshold=
 hoher Schwellwert Host-Flattern, [Optionen der Hauptkonfigurationsdatei](#)
 high_service_flap_threshold=

hoher Schwellwert Service-Flattern, [Optionen der Hauptkonfigurationsdatei](#)
 Host-Abhangigkeits-Definition (hostdependency-Definition), Host-Abhangigkeits-Definition
[\(hostdependency\)](#)
 Host-Abhangigkeits-Definitionen, [Zeitsparende Tricks fur Objektdefinitionen](#)
 Host-Definition, [Host-Definition](#)
 Host-Eskalations-Definition (hostescalation-Definition), [Host-Eskalations-Definition](#)
 Host-Eskalations-Definitionen, [Zeitsparende Tricks fur Objektdefinitionen](#)
 Host-Prfungen, [Host-Prfungen \(Host checks\)](#)
 Hostgroups, [Zeitsparende Tricks fur Objektdefinitionen](#)
 Hostgruppen-Definition, [Hostgruppen-Definition](#)
 hostprops, [Informationen zu den CGI-Parametern](#)
 hoststatustypes,
 host_check_timeout=
 Host-Prfungs-Zeit overschreitung, [Optionen der Hauptkonfigurationsdatei](#)
 host_freshness_check_interval=
 Intervall Host-Frischeprfung, [Optionen der Hauptkonfigurationsdatei](#)
 host_inter_check_delay_method=
 Verzogerungsmethode bei Host-Prfungen, [Optionen der Hauptkonfigurationsdatei](#)
 host_perfdata_command=
 Host-Performance-Daten-Verarbeitung, [Optionen der Hauptkonfigurationsdatei](#)
 host_perfdata_file=
 Host-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 host_perfdata_file_mode=
 Modus Host-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 host_perfdata_file_processing_command=
 Verarbeitungsbefehl Host-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 host_perfdata_file_processing_interval=
 Verarbeitungsintervall Host-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 host_perfdata_file_template=
 Vorlage Host-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 host_perfdata_process_empty_results
 Leere Host-Performance-Daten-Ergebnisse verarbeiten, [Optionen der Hauptkonfigurationsdatei](#)
Howtos
 Developing cronks for Icinga-Web, [Links zu weiteren Howtos](#)
 Developing modules for Icinga-Web, [Links zu weiteren Howtos](#)
 Icinga and Oracle, [Links zu weiteren Howtos](#)
 Icinga Installation, quick and dirty, without database, [Links zu weiteren Howtos](#)
 Icinga, IDOUtils und MySQL-Installations-HowTo, [Links zu weiteren Howtos](#)
 Icinga- und Icinga-Web-Installation unter Ubuntu, [Links zu weiteren Howtos](#)
 Icinga-Reporting installation guide (with JasperServer), [Links zu weiteren Howtos](#)
 Installation and configuration Icinga on CentOS, [Links zu weiteren Howtos](#)
 Installation unter Debian 5 (lenny) mit IDOUtils und MySQL DB, [Links zu weiteren Howtos](#)
 Installing Icinga with IDOUtils, MySQL and NConf, [Links zu weiteren Howtos](#)
 mod_gearman and Icinga, [Links zu weiteren Howtos](#)
 Setting up AD authentication for Icinga-Web, [Links zu weiteren Howtos](#)
 Setting up DNX with Icinga, [Links zu weiteren Howtos](#)
 Setting up mod gearman with Icinga, [Links zu weiteren Howtos](#)
 Step By Step install using RPMS for fc13, [Links zu weiteren Howtos](#)
 http_charset=, [Optionen CGI-Konfigurationsdatei](#)

Icinga für maximale Leistung optimieren, [Icinga für maximale Leistung optimieren](#)
Icinga Kommandozeilenoptionen

 Option -d (Daemon-Modus), [Icinga starten und stoppen](#)

 Option -v (Konfiguration überprüfen), [Überprüfen Ihrer Icinga-Konfiguration](#)

Icinga Log-Optionen, [Icinga starten und stoppen](#)

Icinga starten und stoppen, [Icinga starten und stoppen](#)

Icinga-API, [Installation und Benutzung der Icinga-API](#)

 API/Icinga-Web, [Die Icinga-Web REST API](#)

Icinga-Kommandozeilenoptionen

 Option -p (Precache von Objekten), [Schnellstart-Optionen](#)

 Option -S (Timing- und Scheduling-Informationen sowie Scheduling-Queue), [Schnellstart-Optionen](#)

 Option -s (Timing- und Scheduling-Informationen), [Schnellstart-Optionen](#)

 Option -u (Precached-Objekte benutzen), [Schnellstart-Optionen](#)

 Option -x (nicht auf zirkuläre Pfade prüfen), [Schnellstart-Optionen](#)

Icinga-Pakete, [Schnellstart-Installationsanleitungen](#)

Icinga-Reporting with JasperServer

 Icinga-Reporting, [Installation von Icinga-Reporting mit JasperServer](#)

Icinga-Web

 Aktualisierung von, [Aktualisierung von Icinga-Web und Icinga-Web Datenbank](#)

 Authentifizierung, [Konfigurationsübersicht Icinga-Web](#)

 Datenbank-Update, [Aktualisierung von Icinga-Web und Icinga-Web Datenbank](#)

 Icinga-API connection, [Konfigurationsübersicht Icinga-Web](#)

 Modul-Konfiguration, [Konfigurationsübersicht Icinga-Web](#)

 Überblick Version 1.6.x, Einführung in Icinga-Web (>= 1.6)

Icinga-Web Zeitzone, [Konfigurationsübersicht Icinga-Web](#)

Icinga-Web, benutzerdefinierte Konfiguration, [Konfigurationsübersicht Icinga-Web](#)

Icingastats

 benutzen des Icingastats-Utilitys, [Nutzung des Icingastats-Utilitys](#)

icinga_group=

 Icinga Gruppe, [Optionen der Hauptkonfigurationsdatei](#)

icinga_user=

 Icinga Benutzer, [Optionen der Hauptkonfigurationsdatei](#)

ido2db_group=

 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)

ido2db_user=

 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)

IDOUtils

 Buffer file, [IDOMOD Konfigurationsoptionen](#)

 Central Tables, [Central Tables](#)

 Config output options, [IDOMOD Konfigurationsoptionen](#)

 Configuration of, [Konfiguration der IDOUtils](#)

 Configuration of IDOUtils, [Konfiguration der IDOUtils](#)

 Configuration Tables, [Configuration Tables](#)

 Current Status Tables, [Current Status Tables](#)

 Data processing options, [IDOMOD Konfigurationsoptionen](#)

 Database host, [IDO2DB Konfigurationsoptionen](#)

 Database name, [IDO2DB Konfigurationsoptionen](#)

 Database password, [IDO2DB Konfigurationsoptionen](#)

 Database port, [IDO2DB Konfigurationsoptionen](#)

 Database prefix, [IDO2DB Konfigurationsoptionen](#)

Database server type, [IDO2DB Konfigurationsoptionen](#)
Database socket, [IDO2DB Konfigurationsoptionen](#)
Database user, [IDO2DB Konfigurationsoptionen](#)
Debug file name (ido2db.cfg), [IDO2DB Konfigurationsoptionen](#)
Debug file name (idomod.cfg), [IDOMOD Konfigurationsoptionen](#)
Debug level (ido2db.cfg), [IDO2DB Konfigurationsoptionen](#)
Debug level (idomod.cfg), [IDOMOD Konfigurationsoptionen](#)
Debug verbosity, [IDO2DB Konfigurationsoptionen](#)
Debug verbosity (idomod.cfg), [IDOMOD Konfigurationsoptionen](#)
Debugging Tables, [Debugging Tables](#)
Enable SLA data collection, [IDO2DB Konfigurationsoptionen](#)
File rotation command, [IDOMOD Konfigurationsoptionen](#)
File rotation interval, [IDOMOD Konfigurationsoptionen](#)
File rotation timeout, [IDOMOD Konfigurationsoptionen](#)
Hints on performance / troubleshooting, [Konfiguration der IDOUtils](#)
Historical Tables, [Historical Tables](#)
IDO2DB config options, [IDO2DB Konfigurationsoptionen](#)
IDO2DB group, [IDO2DB Konfigurationsoptionen](#)
IDO2DB user, [IDO2DB Konfigurationsoptionen](#)
ido2db.cfg, [Konfiguration der IDOUtils](#)
IDOMOD config options, [IDOMOD Konfigurationsoptionen](#)
idomod.cfg, [Konfiguration der IDOUtils](#)
Instance name, [IDOMOD Konfigurationsoptionen](#)
Instanznamen ändern, [Datenbank-Anpassungen/Änderungen](#)
Lock file, [IDO2DB Konfigurationsoptionen](#)
Maximum debug file size (ido2db.cfg), [IDO2DB Konfigurationsoptionen](#)
Maximum debug file size (idomod.cfg), [IDOMOD Konfigurationsoptionen](#)
OCI errors to syslog, [IDO2DB Konfigurationsoptionen](#)
Oracle trace level, [IDO2DB Konfigurationsoptionen](#)
Output, [IDOMOD Konfigurationsoptionen](#)
Output buffer size, [IDOMOD Konfigurationsoptionen](#)
Output type, [IDOMOD Konfigurationsoptionen](#)
Readable debug timestamp, [IDO2DB Konfigurationsoptionen](#)
Reconnect interval, [IDOMOD Konfigurationsoptionen](#)
Reconnect warning interval, [IDOMOD Konfigurationsoptionen](#)
Socket name, [IDO2DB Konfigurationsoptionen](#)
Socket type, [IDO2DB Konfigurationsoptionen](#)
Some hints on performance and troubleshooting, [Konfiguration der IDOUtils](#)
TCP-port (ido2db.cfg), [IDO2DB Konfigurationsoptionen](#)
TCP-port (idomod.cfg), [IDOMOD Konfigurationsoptionen](#)
Use encryption (ido2db.cfg), [IDO2DB Konfigurationsoptionen](#)
Use encryption (idomod.cfg), [IDOMOD Konfigurationsoptionen](#)
IDOUtils / Zeitzonen-Unterstützung, [Icinga-Schnellstart mit IDOUtils](#), [Icinga-Schnellstart mit IDOUtils](#)
illegal_macro_output_chars=
 Illegal Makroausgabe, [Optionen der Hauptkonfigurationsdatei](#)
illegal_object_name_chars=
 Illegal Objektnamen, [Optionen der Hauptkonfigurationsdatei](#)
include_dir, [Überblick Objektkonfiguration](#)
include_file, [Überblick Objektkonfiguration](#)
Installation des Webinterfaces, [Installation des Icinga-Web Frontend](#)
instance_name=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)

Integration

SNMP-Trap-Integration, [SNMP-Trap-Integration](#)
 TCP-Wrapper-Integration, [TCP-Wrapper-Integration](#)
 Überblick, [Integrationsüberblick](#)

interval_length=

Intervalllänge, [Optionen der Hauptkonfigurationsdatei](#)

K

Kompatibilität, [Über Icinga](#)

Konfiguration

check-config, [Überprüfen Ihrer Icinga-Konfiguration](#)
 Optionen der Hauptkonfigurationsdatei, [Optionen der Hauptkonfigurationsdatei](#)
 show-errors, [Überprüfen Ihrer Icinga-Konfiguration](#)
 Überblick Objektkonfiguration, [Überblick Objektkonfiguration](#)
 Überprüfung Ihrer Konfiguration, [Überprüfen Ihrer Icinga-Konfiguration](#)
 Konfigurationsoptionen Icinga-Web, [Konfigurationsübersicht Icinga-Web](#)
 Konfigurationsüberblick, [Konfigurationsüberblick](#)
 Kontakt-Definition, [Kontakt-Definition](#)
 Kontaktgruppen-Definition, [Kontaktgruppen-Definition](#)

L

Large Installation Tweaks, [Large Installation Tweaks](#)

LConf, [Icinga Addons](#)

Lilac, [Icinga Addons](#)

Lizenzierung, [Über Icinga](#)

lock_author=

Autorennamen sperren, [Optionen CGI-Konfigurationsdatei](#)

lock_file=

ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
 Sperrdatei, [Optionen der Hauptkonfigurationsdatei](#)

log_archive_path=

Protokollarchiv-Pfad, [Optionen der Hauptkonfigurationsdatei](#)

log_event_handlers=

protokollieren Eventhandler, [Optionen der Hauptkonfigurationsdatei](#)

log_external_commands=

protokollieren externe Befehle, [Optionen der Hauptkonfigurationsdatei](#)

log_file=

Log-Datei, [Optionen der Hauptkonfigurationsdatei](#)

log_host_retries=

protokollieren Host-Prüfungswiederholungen, [Optionen der Hauptkonfigurationsdatei](#)

log_initial_states=

protokollieren initiale Zustände, [Optionen der Hauptkonfigurationsdatei](#)

log_notifications=

Benachrichtigungsprotokollierung, [Optionen der Hauptkonfigurationsdatei](#)

log_passive_checks=

protokollieren passive Prüfungen, [Optionen der Hauptkonfigurationsdatei](#)

log_rotation_method=

Protokoll-Rotationsmethode, [Optionen der Hauptkonfigurationsdatei](#)

log_service_retries=

protokollieren Service-Prüfungswiederholungen, [Optionen der Hauptkonfigurationsdatei](#)

low_host_flap_threshold=

niedriger Schwellwert Host-Flattern, [Optionen der Hauptkonfigurationsdatei](#)
low_service_flap_threshold=
 niedriger Schwellwert Service-Flattern, [Optionen der Hauptkonfigurationsdatei](#)

M

main_cfg_file=

Position der Hauptkonfigurationsdatei, [Optionen CGI-Konfigurationsdatei](#)

Makros

\$ADMINEMAIL\$, [Standard-Makros in Icinga](#)
 \$ADMINPAGER\$, [Standard-Makros in Icinga](#)
 \$ARGn\$, [Standard-Makros in Icinga](#)
 \$COMMANDFILE\$, [Standard-Makros in Icinga](#)
 \$COMMENTDATAFILE\$, [Standard-Makros in Icinga](#)
 \$CONTACTADDRESSn\$, [Standard-Makros in Icinga](#)
 \$CONTACTALIAS\$, [Standard-Makros in Icinga](#)
 \$CONTACTEMAIL\$, [Standard-Makros in Icinga](#)
 \$CONTACTGROUPALIAS\$, [Standard-Makros in Icinga](#)
 \$CONTACTGROUPMEMBERS\$, [Standard-Makros in Icinga](#)
 \$CONTACTGROUPNAME\$, [Standard-Makros in Icinga](#)
 \$CONTACTGROUPNAMES\$, [Standard-Makros in Icinga](#)
 \$CONTACTNAME\$, [Standard-Makros in Icinga](#)
 \$CONTACTPAGER\$, [Standard-Makros in Icinga](#)
 \$DATE\$, [Standard-Makros in Icinga](#)
 \$DOWNTIMEDATAFILE\$, [Standard-Makros in Icinga](#)
 \$EVENTSTARTTIME\$, [Standard-Makros in Icinga](#)
 \$HOSTACKAUTHOR\$, [Standard-Makros in Icinga](#)
 \$HOSTACKAUTHORALIAS\$, [Standard-Makros in Icinga](#)
 \$HOSTACKAUTHORNAME\$, [Standard-Makros in Icinga](#)
 \$HOSTACKCOMMENT\$, [Standard-Makros in Icinga](#)
 \$HOSTACTIONURL\$, [Standard-Makros in Icinga](#)
 \$HOSTADDRESS\$, [Standard-Makros in Icinga](#)
 \$HOSTALIAS\$, [Standard-Makros in Icinga](#)
 \$HOSTATTEMPT\$, [Standard-Makros in Icinga](#)
 \$HOSTCHECKCOMMAND\$, [Standard-Makros in Icinga](#)
 \$HOSTDISPLAYNAME\$, [Standard-Makros in Icinga](#)
 \$HOSTDOWNTIME\$, [Standard-Makros in Icinga](#)
 \$HOSTDURATION\$, [Standard-Makros in Icinga](#)
 \$HOSTDURATIONSEC\$, [Standard-Makros in Icinga](#)
 \$HOSTEVENTID\$, [Standard-Makros in Icinga](#)
 \$HOSTEXECUTIONTIME\$, [Standard-Makros in Icinga](#)
 \$HOSTGROUPACTIONURL\$, [Standard-Makros in Icinga](#)
 \$HOSTGROUPALIAS\$, [Standard-Makros in Icinga](#)
 \$HOSTGROUPMEMBERS\$, [Standard-Makros in Icinga](#)
 \$HOSTGROUPNAME\$, [Standard-Makros in Icinga](#)
 \$HOSTGROUPNAMES\$, [Standard-Makros in Icinga](#)
 \$HOSTGROUPNOTES\$, [Standard-Makros in Icinga](#)
 \$HOSTGROUPNOTESURL\$, [Standard-Makros in Icinga](#)
 \$HOSTLATENCY\$, [Standard-Makros in Icinga](#)
 \$HOSTNAME\$, [Standard-Makros in Icinga](#)
 \$HOSTNOTESS\$, [Standard-Makros in Icinga](#)
 \$HOSTNOTESURL\$, [Standard-Makros in Icinga](#)
 \$HOSTNOTIFICATIONID\$, [Standard-Makros in Icinga](#)

\$HOSTNOTIFICATIONNUMBER\$, Standard-Makros in Icinga
 \$HOSTOUTPUT\$, Standard-Makros in Icinga
 \$HOSTPERCENTCHANGE\$, Standard-Makros in Icinga
 \$HOSTPERFDATA\$, Standard-Makros in Icinga
 \$HOSTPERFDATAFILE\$, Standard-Makros in Icinga
 \$HOSTPROBLEMID\$, Standard-Makros in Icinga
 \$HOSTSTATE\$, Standard-Makros in Icinga
 \$HOSTSTATEID\$, Standard-Makros in Icinga
 \$HOSTSTATETYPE\$, Standard-Makros in Icinga
 \$LASTHOSTCHECK\$, Standard-Makros in Icinga
 \$LASTHOSTDOWN\$, Standard-Makros in Icinga
 \$LASTHOSTEVENTID\$, Standard-Makros in Icinga
 \$LASTHOSTPROBLEMID\$, Standard-Makros in Icinga
 \$LASTHOSTSTATE\$, Standard-Makros in Icinga
 \$LASTHOSTSTATECHANGE\$, Standard-Makros in Icinga
 \$LASTHOSTSTATEID\$, Standard-Makros in Icinga
 \$LASTHOSTUNREACHABLE\$, Standard-Makros in Icinga
 \$LASTHOSTUP\$, Standard-Makros in Icinga
 \$LASTSERVICECHECK\$, Standard-Makros in Icinga
 \$LASTSERVICECRITICAL\$, Standard-Makros in Icinga
 \$LASTSERVICEEVENTID\$, Standard-Makros in Icinga
 \$LASTSERVICEOK\$, Standard-Makros in Icinga
 \$LASTSERVICEPROBLEMID\$, Standard-Makros in Icinga
 \$LASTSERVICESTATE\$, Standard-Makros in Icinga
 \$LASTSERVICESTATECHANGE\$, Standard-Makros in Icinga
 \$LASTSERVICESTATEID\$, Standard-Makros in Icinga
 \$LASTSERVICEUNKNOWN\$, Standard-Makros in Icinga
 \$LASTSERVICEWARNING\$, Standard-Makros in Icinga
 \$LOGFILE\$, Standard-Makros in Icinga
 \$LONGDATETIME\$, Standard-Makros in Icinga
 \$LONGHOSTOUTPUT\$, Standard-Makros in Icinga
 \$LONGSERVICEOUTPUT\$, Standard-Makros in Icinga
 \$MAINCONFIGFILE\$, Standard-Makros in Icinga
 \$MAXHOSTATTEMPTS\$, Standard-Makros in Icinga
 \$MAXSERVICEATTEMPTS\$, Standard-Makros in Icinga
 \$NOTIFICATIONAUTHOR\$, Standard-Makros in Icinga
 \$NOTIFICATIONAUTHORALIAS\$, Standard-Makros in Icinga
 \$NOTIFICATIONAUTHORNAME\$, Standard-Makros in Icinga
 \$NOTIFICATIONCOMMENT\$, Standard-Makros in Icinga
 \$NOTIFICATIONISESCALATED\$, Standard-Makros in Icinga
 \$NOTIFICATIONRECIPIENTS\$, Standard-Makros in Icinga
 \$NOTIFICATIONTYPE\$, Standard-Makros in Icinga
 \$OBJECTCACHEFILE\$, Standard-Makros in Icinga
 \$PROCESSSTARTTIME\$, Standard-Makros in Icinga
 \$RESOURCEFILE\$, Standard-Makros in Icinga
 \$RETENTIONDATAFILE\$, Standard-Makros in Icinga
 \$SERVICEACKAUTHOR\$, Standard-Makros in Icinga
 \$SERVICEACKAUTHORALIAS\$, Standard-Makros in Icinga
 \$SERVICEACKAUTHORNAME\$, Standard-Makros in Icinga
 \$SERVICEACKCOMMENT\$, Standard-Makros in Icinga
 \$SERVICEACTIONURL\$, Standard-Makros in Icinga
 \$SERVICEATTEMPT\$, Standard-Makros in Icinga
 \$SERVICECHECKCOMMAND\$, Standard-Makros in Icinga

\$SERVICEDESC\$, Standard-Makros in Icinga
 \$SERVICEDISPLAYNAME\$, Standard-Makros in Icinga
 \$SERVICEDOWNTIME\$, Standard-Makros in Icinga
 \$SERVICEDURATION\$, Standard-Makros in Icinga
 \$SERVICEDURATIONSEC\$, Standard-Makros in Icinga
 \$SERVICEEVENTID\$, Standard-Makros in Icinga
 \$SERVICEEXECUTIONTIME\$, Standard-Makros in Icinga
 \$SERVICEGROUPACTIONURL\$, Standard-Makros in Icinga
 \$SERVICEGROUPALIAS\$, Standard-Makros in Icinga
 \$SERVICEGROUPMEMBERS\$, Standard-Makros in Icinga
 \$SERVICEGROUPNAME\$, Standard-Makros in Icinga
 \$SERVICEGROUPNAMES\$, Standard-Makros in Icinga
 \$SERVICEGROUPNOTES\$, Standard-Makros in Icinga
 \$SERVICEGROUPNOTESURL\$, Standard-Makros in Icinga
 \$SERVICEISVOLATILE\$, Standard-Makros in Icinga
 \$SERVICELATENCY\$, Standard-Makros in Icinga
 \$SERVICENOTES\$, Standard-Makros in Icinga
 \$SERVICENOTESURL\$, Standard-Makros in Icinga
 \$SERVICENOTIFICATIONID\$, Standard-Makros in Icinga
 \$SERVICENOTIFICATIONNUMBER\$, Standard-Makros in Icinga
 \$SERVICEOUTPUT\$, Standard-Makros in Icinga
 \$SERVICEPERCENTCHANGE\$, Standard-Makros in Icinga
 \$SERVICEPERFDATA\$, Standard-Makros in Icinga
 \$SERVICEPERFDATAFILE\$, Standard-Makros in Icinga
 \$SERVICEPROBLEMD\$, Standard-Makros in Icinga
 \$SERVICESTATE\$, Standard-Makros in Icinga
 \$SERVICESTATEID\$, Standard-Makros in Icinga
 \$SERVICESTATETYPE\$, Standard-Makros in Icinga
 \$SHORTDATETIME\$, Standard-Makros in Icinga
 \$STATUSDATAFILE\$, Standard-Makros in Icinga
 \$TEMPFILE\$, Standard-Makros in Icinga
 \$TEMPPATH\$, Standard-Makros in Icinga
 \$TIME\$, Standard-Makros in Icinga
 \$TIMET\$, Standard-Makros in Icinga
 \$TOTALHOSTPROBLEMS\$, Standard-Makros in Icinga
 \$TOTALHOSTPROBLEMSUNHANDLED\$, Standard-Makros in Icinga
 \$TOTALHOSTSDOWN\$, Standard-Makros in Icinga
 \$TOTALHOSTSDOWNUNHANDLED\$, Standard-Makros in Icinga
 \$TOTALHOSTSERVICES\$, Standard-Makros in Icinga
 \$TOTALHOSTSERVICESCRITICAL\$, Standard-Makros in Icinga
 \$TOTALHOSTSERVICESOK\$, Standard-Makros in Icinga
 \$TOTALHOSTSERVICESUNKNOWN\$, Standard-Makros in Icinga
 \$TOTALHOSTSERVICESWARNING\$, Standard-Makros in Icinga
 \$TOTALHOSTSUNREACHABLE\$, Standard-Makros in Icinga
 \$TOTALHOSTSUNREACHABLEUNHANDLED\$, Standard-Makros in Icinga
 \$TOTALHOSTSUP\$, Standard-Makros in Icinga
 \$TOTALSERVICEPROBLEMS\$, Standard-Makros in Icinga
 \$TOTALSERVICEPROBLEMSUNHANDLED\$, Standard-Makros in Icinga
 \$TOTALSERVICESCRITICAL\$, Standard-Makros in Icinga
 \$TOTALSERVICESCRITICALUNHANDLED\$, Standard-Makros in Icinga
 \$TOTALSERVICESOK\$, Standard-Makros in Icinga
 \$TOTALSERVICESUNKNOWN\$, Standard-Makros in Icinga
 \$TOTALSERVICESUNKNOWNUNHANDLED\$, Standard-Makros in Icinga

\$TOTALSERVICESWARNING\$, Standard-Makros in Icinga
 \$TOTALSERVICESWARNINGUNHANDLED\$, Standard-Makros in Icinga
 \$USERn\$, Standard-Makros in Icinga
 Auswertungsmakros, Standard-Makros in Icinga
 Standardmakros in Icinga, Standard-Makros in Icinga
 Summary Macros, Standard-Makros in Icinga
 verstehen von Makros und wie sie funktionieren, Makros verstehen und wie sie arbeiten
max_check_result_file_age=
 max. Alter von Prüfergebnisdateien, Optionen der Hauptkonfigurationsdatei
max_check_result_reaper_time=
 maximale Prüfergebnis-Erntezeit, Optionen der Hauptkonfigurationsdatei
max_concurrent_checks=
 maximale Anzahl gleichzeitiger Service-Prüfungen, Optionen der Hauptkonfigurationsdatei
max_debug_file_size=
 ido2db.cfg, IDO2DB Konfigurationsoptionen
 idomod.cfg, IDOMOD Konfigurationsoptionen
 max. Debug-Dateigröße, Optionen der Hauptkonfigurationsdatei
max_host_check_spread=
 maximaler Zeitraum, bei Host-Verteilung, Optionen der Hauptkonfigurationsdatei
max_service_check_spread=
 maximaler Zeitraum bei Service-Verteilung, Optionen der Hauptkonfigurationsdatei
 Maßgeschneiderte Objektvariablen, Maßgeschneiderte Objektvariablen
 Ersetzung erst zur Laufzeit, Maßgeschneiderte Objektvariablen
 Migration
 von Nagios nach Icinga, Icinga aktualisieren
 MKLiveStatus-Integration, MKLiveStatus-Integration
 Monitoring
 Linux/Unix-Rechner überwachen, Linux/Unix-Rechner überwachen
 Netware-Server überwachen, Netware-Server überwachen
 Netzwerk-Drucker überwachen, Netzwerk-Drucker überwachen
 Öffentlich zugängliche Dienste überwachen, Öffentlich zugängliche Dienste überwachen
 Redundante und Failover-Netzwerk-Überwachung, Redundante und Failover-Netzwerk-Überwachung
 Router und Switches überwachen, Router und Switches überwachen
 Windows-Rechner überwachen, Windows-Maschinen überwachen
 MultiSite, Icinga Addons

N

Nagios
 von Nagios nach Icinga migrieren, Icinga aktualisieren
 NagiosQL, Icinga Addons
 NagVis, Icinga Addons
 NConf, Icinga Addons
 Netzwerk-Hosts
 Ermitteln des Zustands und der Erreichbarkeit von Netzwerk-Hosts, Ermitteln des Zustands und der Erreichbarkeit von Netzwerk-Hosts
notes_url_target=
 Notes-URL-Ziel, Optionen CGI-Konfigurationsdatei
notification_timeout=
 Benachrichtigungs-Zeitüberschreitung, Optionen der Hauptkonfigurationsdatei
 nrpe

nrpe, [NRPE](#)
 send_nrpe, [NRPE](#)
NSCA
 nsca, [NSCA](#)
 send_nsca, [NSCA](#)
NSClient++, Icinga Addons
 Windows-Rechner überwachen, Windows-Maschinen überwachen

O

Object cache file, [Temporäre Daten](#)
Object Configuration Overview
 include_dir, [Überblick Objektkonfiguration](#)
 include_file, [Überblick Objektkonfiguration](#)
 object_cache_file=
 Objekt-Cache-Datei, [Optionen der Hauptkonfigurationsdatei](#)
Objektdefinitionen, Objektdefinitionen
 Command, [Befehls-Definition \(command\)](#)
 Contact, [Kontakt-Definition](#)
 Contactgroup, [Kontaktgruppen-Definition](#)
 Host, [Host-Definition](#)
 Host Dependency, [Host-Abhängigkeits-Definition \(hostdependency\)](#)
 Host Escalation, [Host-Eskalations-Definition](#)
 Host Extended Information, [erweiterte Hostinformations-Definition \(hostextinfo\)](#)
 Hostgroup, [Hostgruppen-Definition](#)
 Module, [Module-Definition](#)
 Retention, [Objektdefinitionen](#)
 Service, [Service-Definition](#)
 Service Escalation, [Serviceeskalations-Definition](#)
 Service Extended Information, [erweiterte Serviceinformations-Definition \(serviceextinfo\)](#)
 ServiceDependency, [Service-Abhängigkeits-Definition \(servicedependency\)](#)
 Servicegroup, [Servicegruppen-Definition](#)
 Timeperiod, [Zeitfenster-Definition \(timeperiod\)](#)
 Zeitsparende Tricks für Objektdefinitionen, [Zeitsparende Tricks für Objektdefinitionen](#)
Objektvererbung, Objektvererbung
 additive Vererbung von Zeichenketten-Werten, [Objektvererbung](#)
 Implizite Vererbung, [Objektvererbung](#)
 Implizite/additive Vererbung in Eskalationen, [Objektvererbung](#)
 mehrere Vererbungsquellen, [Objektvererbung](#)
 Vererbung für Zeichenketten-Werte aufheben, [Objektvererbung](#)
 obsess_over_hosts=
 Obsess Over Hosts Option, [Optionen der Hauptkonfigurationsdatei](#)
 obsess_over_services=
 Obsess Over Services Option, [Optionen der Hauptkonfigurationsdatei](#)
 ochp_command=
 Obsessive Compulsive Host Processor, [Optionen der Hauptkonfigurationsdatei](#)
 ochp_timeout=
 Obsessive Compulsive Host Processor, [Optionen der Hauptkonfigurationsdatei](#)
 oci_errors_to_syslog=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
 ocsp_command=
 Obsessive Compulsive Service Processor, [Optionen der Hauptkonfigurationsdatei](#)
 ocsp_timeout=

Obsessive Compulsive Service Processor, [Optionen der Hauptkonfigurationsdatei](#)
 oracle_trace_level=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
 output=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
 output_buffer_items=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
 output_type=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)

P

Passive Host-Zustandsübersetzung, [Passive Host-Zustandsübersetzung](#)
 Passive Prüfungen, [Passive Prüfungen \(Passive Checks\)](#)
 passive_host_checks_are_soft=
 passive Host-Prüfungen sind SOFT, [Optionen der Hauptkonfigurationsdatei](#)
 perfdata_timeout=
 Performance-Daten-Verarbeitungsbefehl, [Optionen der Hauptkonfigurationsdatei](#)
 Performance-Daten, [Performance-Daten](#)
 persistent_ack_comments=
 Persistente Bestätigungskommentare, [Optionen CGI-Konfigurationsdatei](#)
 physical_html_path=
 vollständiger (physical) HTML-Pfad, [Optionen CGI-Konfigurationsdatei](#)
 ping_syntax=
 Ping-Syntax, [Optionen CGI-Konfigurationsdatei](#)
 Plugins
 Icinga Plugin-API, [Nagios Plugin API](#)
 Icinga-Plugins, [Icinga Plugins](#)
 Integration eines neuen Plugins, [Icinga Plugins](#)
 Plugins und Makros, [Icinga Plugins](#)
 Wie benutze ich Plugin X?, [Icinga Plugins](#)
 PNP4Nagios, [Icinga Addons](#)
 pnp und Icinga-Web, [Integration von PNP4Nagios in das Icinga-Web Frontend](#)
 precached_object_file=
 vorgespeicherte Objektdatei, [Optionen der Hauptkonfigurationsdatei](#)
 process_performance_data=
 Performance-Daten-Verarbeitung, [Optionen der Hauptkonfigurationsdatei](#)

Q

Quickstart, [Schnellstart-Installationsanleitungen](#)
 Icinga auf FreeBSD, [Icinga-Schnellstart auf FreeBSD](#)
 Icinga auf Linux, [Icinga-Schnellstart auf Linux](#)
 Icinga mit IDOUtils, [Icinga-Schnellstart mit IDOUtils](#)
 Icinga und IDOUtils auf FreeBSD, [Icinga-Schnellstart mit IDOUtils auf FreeBSD](#)

R

RAM-Disk, [Temporäre Daten](#)
 reconnect_interval=
 idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
 reconnect_warning_interval=

idomod.cfg, [IDOMOD Konfigurationsoptionen](#)
 Redundanz
 Redundante und Failover-Netzwerk-Überwachung, [Redundante und Failover-Netzwerk-Überwachung](#)
 refresh_rate=
 CGI-Refresh-Rate, [Optionen CGI-Konfigurationsdatei](#)
 resource_file=
 Ressource-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 retained_contact_host_attribute_mask=
 aufbewahrte "Contact Host"-Attributmaske, [Optionen der Hauptkonfigurationsdatei](#)
 retained_contact_service_attribute_mask=
 aufbewahrte "Contact Service"-Attributmaske, [Optionen der Hauptkonfigurationsdatei](#)
 retained_host_attribute_mask=
 aufbewahrte Host-Attributmaske, [Optionen der Hauptkonfigurationsdatei](#)
 retained_process_attribute_mask=
 aufbewahrte Prozess-Attributmaske, [Optionen der Hauptkonfigurationsdatei](#)
 retained_process_host_attribute_mask=
 aufbewahrte "Process Host"-Attributmaske, [Optionen der Hauptkonfigurationsdatei](#)
 retained_scheduling_info=
 benutzen von aufbewahrten Planungsinformationen, [Optionen der Hauptkonfigurationsdatei](#)
 retained_service_attribute_mask=
 aufbewahrte Service-Attributmaske, [Optionen der Hauptkonfigurationsdatei](#)
 retain_state_information=
 aufbewahren von Statusinformationen, [Optionen der Hauptkonfigurationsdatei](#)
 retention_update_interval=
 Aufbewahrungs-Aktualisierungsintervall, [Optionen der Hauptkonfigurationsdatei](#)

S

Scheduling

Host-Prüfungen, [Service- und Host-Prüfungsplanung](#)
 Host-Prüfungsdirektiven, [Service- und Host-Prüfungsplanung](#)
 Initiale Planung, [Service- und Host-Prüfungsplanung](#)
 Inter-Check-Verzögerung (inter-check delay), [Service- und Host-Prüfungsplanung](#)
 Konfigurationsoptionen, [Service- und Host-Prüfungsplanung](#)
 Maximale Zahl gleichzeitiger Service-Prüfungen (maximum concurrent service checks), [Service- und Host-Prüfungsplanung](#)
 Normale Planung, [Service- und Host-Prüfungsplanung](#)
 Planung bei Problemen, [Service- und Host-Prüfungsplanung](#)
 Planungsbeispiel, [Service- und Host-Prüfungsplanung](#)
 Planungsverzögerungen, [Service- und Host-Prüfungsplanung](#)
 Service- und Host-Prüfungsplanung, [Service- und Host-Prüfungsplanung](#)
 Service-Definitionsoptionen, die die Planung beeinflussen, [Service- und Host-Prüfungsplanung](#)
 Service-Verschachtelung (service interleaving), [Service- und Host-Prüfungsplanung](#)
 Zeitbeschränkungen, [Service- und Host-Prüfungsplanung](#)
 Schnellstart, [Schnellstart-Installationsanleitungen](#)
 Icinga auf \$name-linux; [Icinga-Schnellstart auf Linux](#)
 Schnellstartoptionen (Fast Startup Options), [Schnellstart-Optionen](#)
 Service-Abhängigkeits-Definition (servicedependency-Definition), [Service-Abhängigkeits-Definition \(servicedependency\)](#)
 Service-Abhängigkeits-Definitionen, [Zeitsparende Tricks für Objektdefinitionen](#)

Service-Definition, [Service-Definition](#)
 Service-Definitionen, [Zeitsparende Tricks für Objektdefinitionen](#)
 Service-Eskalations-Definitionen, [Zeitsparende Tricks für Objektdefinitionen](#)
 Service-Prüfungen, [Service-Prüfungen \(Service Checks\)](#)
 Serviceescalation-Definition, [Serviceescalations-Definition](#)
 Servicegruppen-Definition, [Servicegruppen-Definition](#)
 serviceprops, [Informationen zu den CGI-Parametern](#)
 servicestatustypes,
 service_check_timeout=
 Service-Prüfungs-Zeitüberschreitung, [Optionen der Hauptkonfigurationsdatei](#)
 service_check_timeout_state=
 Service-Prüfungs-Timeout-Status, [Optionen der Hauptkonfigurationsdatei](#)
 service_freshness_check_interval=
 Intervall Service-Frischeprüfung, [Optionen der Hauptkonfigurationsdatei](#)
 service_interleave_factor=
 Service-Verschachtelungsfaktor, [Optionen der Hauptkonfigurationsdatei](#)
 service_inter_check_delay_method=
 Verzögerungsmethode für Service-Prüfungen, [Optionen der Hauptkonfigurationsdatei](#)
 service_perfdata_command=
 Service-Performance-Daten-Verarbeitung, [Optionen der Hauptkonfigurationsdatei](#)
 service_perfdata_file=
 Service-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 service_perfdata_file_mode=
 Modus Service-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 service_perfdata_file_processing_command=
 Verarbeitungsbefehl Service-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 service_perfdata_file_processing_interval=
 Verarbeitungsintervall Service-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 service_perfdata_file_template=
 Vorlage Service-Performance-Daten-Datei, [Optionen der Hauptkonfigurationsdatei](#)
 service_perfdata_process_empty_results
 Leere Service-Performance-Daten-Ergebnisse verarbeiten, [Optionen der Hauptkonfigurationsdatei](#)
 Session Cookie Lifetime, [Konfigurationsübersicht Icinga-Web](#)
 showlog_current_states=
 Aktuelle Zustände anzeigen, [Optionen CGI-Konfigurationsdatei](#)
 showlog_initial_states=
 Initiale Zustände anzeigen, [Optionen CGI-Konfigurationsdatei](#)
 show_tac_header=
 Tactical Overview-Header anzeigen, [Optionen CGI-Konfigurationsdatei](#)
 show_tac_header_pending=
 Pending-Anzahlen in Tactical Overview-Header anzeigen, [Optionen CGI-Konfigurationsdatei](#)
 Sicherheitsüberlegungen, [Sicherheitsüberlegungen](#)
 sleep_time=
 Ruhezeit zwischen Prüfungen, [Optionen der Hauptkonfigurationsdatei](#)
 socket_name=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
 socket_type=
 ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)
 soft_state_dependencies=

Soft-Status-Abhängigkeiten, [Optionen der Hauptkonfigurationsdatei](#)
splunk_url=
 Splunk-URL, [Optionen CGI-Konfigurationsdatei](#)
 Sprunghafte Services (volatile services, [sprunghafte Services](#))
state_retention_file=
 Statusaufbewahrungsdatei, [Optionen der Hauptkonfigurationsdatei](#)
 Status file, [Temporäre Daten](#)
statusmap_background_image=
 Statusmap-CGI-Hintergrundbild, [Optionen CGI-Konfigurationsdatei](#)
 Statustypen, [Statustypen](#)
 Statusverfolgung, [Status Stalking](#)
status_file=
 Status-Datei, [Optionen der Hauptkonfigurationsdatei](#)
status_log=
 Statusprotokoll,
status_update_interval=
 Statusdatei-Aktualisierungsintervall, [Optionen der Hauptkonfigurationsdatei](#)
suppress_maintenance_downtime=
 Maintenance/Downtime anzeigen, [Optionen CGI-Konfigurationsdatei](#)
syslog_local_facility=
 Syslog-Local-Facility-Wert, [Optionen der Hauptkonfigurationsdatei](#)
 Systemanforderungen, [Über Icinga](#)

T

Tables

acknowledgements, [Historical Tables](#)
 commands, [Configuration Tables](#)
 commenthistory, [Historical Tables](#)
 comments, [Current Status Tables](#)
 configfiles, [Configuration Tables](#)
 configfilevariables, [Configuration Tables](#)
 conninfo, [Debugging Tables](#)
 contactgroups, [Configuration Tables](#)
 contactgroup_members, [Configuration Tables](#)
 contactnotificationmethods, [Configuration Tables](#)
 contactnotifications, [Historical Tables](#)
 contactnotificationsmethods, [Historical Tables](#)
 contacts, [Configuration Tables](#)
 contactstatus, [Current Status Tables](#)
 contact_addresses, [Configuration Tables](#)
 contact_notificationcommands, [Configuration Tables](#)
 customvariables, [Configuration Tables](#)
 customvariablestatus, [Current Status Tables](#)
 dbversion, [Historical Tables](#)
 downtimehistory, [Historical Tables](#)
 eventhandlers, [Historical Tables](#)
 externalcommands, [Historical Tables](#)
 flappinghistory, [Historical Tables](#)
 hostchecks, [Historical Tables](#)
 hostdependencies, [Configuration Tables](#)
 hostescalations, [Configuration Tables](#)
 hostescalation_contactgroups, [Configuration Tables](#)

T

hostescalation_contacts, Configuration Tables
hostgroups, Configuration Tables
hostgroup_members, Configuration Tables
hosts, Configuration Tables
hoststatus, Current Status Tables
host_contactgroups, Configuration Tables
host_contacts, Configuration Tables
host_parenthosts, Configuration Tables
instances, Central Tables
logentries, Historical Tables
notifications, Historical Tables
objects, Central Tables
processevents, Historical Tables
programstatus, Current Status Tables
runtimevariables, Current Status Tables
scheduledowntime, Current Status Tables
servicechecks, Historical Tables
servicedependencies, Configuration Tables
serviceescalations, Configuration Tables
serviceescalation_contactgroups, Configuration Tables
serviceescalation_contacts, Configuration Tables
servicegroups, Configuration Tables
servicegroup_members, Configuration Tables
services, Configuration Tables
servicestatus, Current Status Tables
service_contactgroups, Configuration Tables
service_contacts, Configuration Tables
slahistory, Historical Tables
statehistory, Historical Tables
systemcommands, Historical Tables
timedequeue, Current Status Tables
timedeevents, Historical Tables
timeperiods, Configuration Tables
timeperiod_timeranges, Configuration Tables
tab_friendly_titles=
 Objekttyp im Browser-Reiter anzeigen, Optionen CGI-Konfigurationsdatei
tac_show_only_hard_state=
 Tac Show Only Hard State, Optionen CGI-Konfigurationsdatei
tcp_port=
 ido2db.cfg, IDO2DB Konfigurationsoptionen
 idomod.cfg, IDOMOD Konfigurationsoptionen
Temporary Data, Temporäre Daten
temp_file=
 temporäre Datei, Optionen der Hauptkonfigurationsdatei
temp_path=
 temporärer Pfad, Optionen der Hauptkonfigurationsdatei
Thruk, Icinga Addons
translage_passive_host_checks=
 übersetzen von passiven Host-Prüfergebnissen, Optionen der Hauptkonfigurationsdatei

U

Überprüfen Ihrer Konfiguration, [Überprüfen Ihrer Icinga-Konfiguration](#)
Überwachung

Linux/Unix-Rechner überwachen, [Linux/Unix-Rechner überwachen](#)
Netware-Server überwachen, [Netware-Server überwachen](#)
Netzwerk-Drucker überwachen, [Netzwerk-Drucker überwachen](#)
Öffentlich zugängliche Dienste überwachen, [Öffentlich zugängliche Dienste überwachen](#)
Redundante und Failover-Netzwerk-Überwachung, [Redundante und Failover-Netzwerk-Überwachung](#)

Router und Switches überwachen, [Router und Switches überwachen](#)
Windows-Rechner überwachen, [Windows-Maschinen überwachen](#)

Upgrading IDOUtils, [IDOUtils-Datenbank aktualisieren](#)

url_html_path=

URL-HTML-Pfad, [Optionen CGI-Konfigurationsdatei](#)

use_aggressive_host_checking=

Aggressive Host-Prüfung, [Optionen der Hauptkonfigurationsdatei](#)

use_authentication=

Nutzung der Authentifizierung, [Optionen CGI-Konfigurationsdatei](#)

use_embedded_perl_implicitly=

implizite Benutzung von Embedded Perl, [Optionen der Hauptkonfigurationsdatei](#)

use_large_installation_tweaks=

Large Installation Tweaks, [Optionen der Hauptkonfigurationsdatei](#)

use_logging=

CGI-Protokollierung (de)aktivieren, [Optionen CGI-Konfigurationsdatei](#)

use_regexp_matching=

Regular Expression Matching, [Optionen der Hauptkonfigurationsdatei](#)

use_retained_program_state=

benutzen des aufbewahrten Programmzustands, [Optionen der Hauptkonfigurationsdatei](#)

use_ssl=

ido2db.cfg, [IDO2DB Konfigurationsoptionen](#)

idomod.cfg, [IDOMOD Konfigurationsoptionen](#)

use_syslog=

Syslog-Protokollierung, [Optionen der Hauptkonfigurationsdatei](#)

use_syslog_local_facility=

Syslog-Local-Facility-Protokollierung, [Optionen der Hauptkonfigurationsdatei](#)

use_timezone=

Zeitzone, [Optionen der Hauptkonfigurationsdatei](#)

use_true-regexp_matching=

True Regular Expression Matching, [Optionen der Hauptkonfigurationsdatei](#)

V

Verteilte Überwachung, [Verteilte Überwachung](#)

W

Was ist Icinga?, [Über Icinga](#)

Webinterface, [Installation des Icinga-Web Frontend](#)

What's new in Icinga, [What's New in Icinga 1.6](#)

Wie benutze ich Plugin X?, [Icinga Plugins](#)

Wiki

HowTo's, [Links zu weiteren Howtos](#)

Z

Zeitfenster (time periods), [Zeitfenster](#)

Zeitfenster-Definition, [Zeitfenster-Definition \(timeperiod\)](#)

Zwischengespeicherte Prüfungen (Cached Checks), [Zwischengespeicherte Prüfungen](#)

[Zurück](#)

Datenbank-Anpassungen/Änderungen

[Zum Anfang](#)

© 2009-2011 Icinga Development Team, <http://www.icinga.org>