

Eine Host-Definition wird benutzt, um einen Server, eine Workstation, ein Gerät usw. zu definieren, die sich in Ihrem Netzwerk befinden.

Format der Definition:

Grundsätzlich ist die Funktion `prerequisites` benötigt, die anderen sind optional.

host Gruppe → Service	
<u>host_name</u>	<i>host_name</i>
<u>alias</u>	<i>alias</i>
<u>display_name</u>	<i>display_name</i>
<u>address</u>	<i>address</i>
<u>parents</u>	<i>host_names</i>
<u>hostgroup_names</u>	<i>hostgroup_names</i>
<u>check_command</u>	<i>command_name</i>
<u>initial_state</u>	[o,d,u]
<u>max_check_attempts</u>	#
<u>check_interval</u>	#
<u>retry_interval</u>	#
<u>active_checks_enabled</u>	[0/1]
<u>passive_checks_enabled</u>	[0/1]
<u>check_period</u>	<i>timeperiod_name</i>
<u>obsess_over_host</u>	[0/1]
<u>check_freshness</u>	[0/1]
<u>freshness_threshold</u>	#
<u>event_handler</u>	<i>command_name</i>
<u>event_handler_enabled</u>	[0/1]
<u>low_flap_threshold</u>	#
<u>high_flap_threshold</u>	#
<u>flap_detection_enabled</u>	[0/1]
<u>flap_detection_options</u>	[o,d,u]
<u>process_perf_data</u>	[0/1]
<u>retain_status_information</u>	[0/1]
<u>retain_nonstatus_information</u>	[0/1]
<u>contacts</u>	<i>contacts</i>
<u>contact_groups</u>	<i>contact_groups</i>
<u>notification_interval</u>	#
<u>notification_delay</u>	#
<u>notification_period</u>	<i>timeperiod_name</i>
<u>notification_options</u>	[d,u,r,f,s]
<u>notifications_enabled</u>	[0/1]
<u>stalking_options</u>	[o,d,u]
<u>notes</u>	<i>note_string</i>
<u>notes_url</u>	<i>url</i>
<u>action_url</u>	<i>url</i>
<u>icon_image</u>	<i>image_file</i>
<u>icon_image_alt</u>	<i>alt_string</i>
<u>vrml_image</u>	<i>image_file</i>

Definitionen

Objekttypen

Host-Definitionen

Service-Definitionen

Kontakt-Definitionen

Kontaktgruppen-Definitionen

Zeitfenster-Definitionen

Befehlsdefinitionen

Service-Abhängigkeitsdefinitionen

Service-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Host-Abhängigkeitsdefinitionen

Host-Eskalationsdefinitionen

Das soll heißen, dass man Hosts in Gruppen organisieren kann und diese Gruppen einem Service (Check) zuweisen kann. Man kann aber auch die Gruppe übergehen und den Hosts direkt Services zuweisen. Bei großen Installationen ist das aber nicht sehr empfehlenswert.

statusmap_image	<i>image_file</i>
2d_coords	<i>x_coord,y_coord</i>
3d_coords	<i>x_coord,y_coord,z_coord</i>
}	

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 Domännennamen (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.
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 <i>host_name</i> -Direktive benutzt. Anmerkung: die aktuellen CGIs nutzen diese Option nicht, zukünftige Versionen des Web-Interfaces werden das tun.
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 Nagios-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 <i>Kurznamen</i> 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 <i>members</i> -Direktive in den hostgroup -Definitionen genutzt werden.
check_command:	Diese Direktive wird benutzt, um den <i>Kurznamen</i> des Befehls 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 Nagios annehmen, dass der Host „down“ ist. Wenn Sie diesen Wert leer lassen, wird der Host <i>nicht</i> aktiv geprüft. Dadurch wird Nagios 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 Nagios 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 Nagios den Host-Prüfbefehl wiederholt, wenn er einen anderen als einen OK-Zustand zurückliefert. Bei einem Wert von 1 wird Nagios 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 Nagios automatisch einen Frische-Schwellwert festlegen.

event_handler:	Diese Direktive wird benutzt, um den <i>Kurznamen</i> des Befehls 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 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 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 Werte: 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 Nagios 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 Nagios 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 Nagios 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. <code>/cgi-bin/nagios/</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 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. <code>/cgi-bin/nagios/</code>).
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 40×40 Pixel groß ist. Bilder für Hosts werden im logos -Unterverzeichnis Ihres HTML-Images-Verzeichnis gesucht (d.h. <code>/usr/local/nagios/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.
vrml_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 als Textur-Map für den angegebenen Host in der statuswrl -CGI benutzt. Anders als das Bild, das Sie in der <code><icon_image></code> -Variable angeben, sollte dieses möglichst <i>keinerlei</i> Transparenz haben. Wenn es das tut, wird das Host-Objekt ein wenig komisch aussehen. Bilder für Hosts werden im logos -Unterverzeichnis Ihres HTML-Images-Verzeichnis gesucht (d.h. <code>/usr/local/nagios/share/images/logos</code>).

Beschreibung:

Last update:

2011/06/12 wiki:programme:nagios:definitionen <https://wiki.da-checka.de/doku.php/wiki/programme/nagios/definitionen?rev=1307876160>

Die Hostgruppen-Definition wird benutzt, um einen oder mehrere Hosts zu gruppieren, um die

<p>Konfiguration mit Objekt-Tricks zu vereinfachen oder für Anzeigezwecke in den CGIs</p> <p>Format der Definition:</p>	<p>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 Direktive werden benötigt, da Änderungen in anderen 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 <i>unkomprimierten</i> Format erstellt, um die CPU-Belastung zu minimieren, während das Statusmap-CGI das Netzwerkartenbild 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-Verzeichnis gesucht (d.h. <i>/usr/local/nagios/share/images/logos</i>).</p>
<p>Anmerkung: die <u>unterstrichenen</u> Direktive werden benötigt, da Änderungen in anderen 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 <i>unkomprimierten</i> Format erstellt, um die CPU-Belastung zu minimieren, während das Statusmap-CGI das Netzwerkartenbild 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-Verzeichnis gesucht (d.h. <i>/usr/local/nagios/share/images/logos</i>).</p>	
<pre>define hostgroup{ hostgroup_name statusmap_image alias members hostgroup_members notes notes_url action_url }</pre>	
<p>Beispieldefinition:</p>	<p>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.</p>
<pre>define hostgroup{ 2d_coords hostgroup_name alias members }</pre>	
<p>Beschreibung der Direktiven:</p>	
<p>hostgroup_name:</p>	<p>Diese Direktive wird benutzt, um einen Kurznamen zu definieren, der die Hostgruppe identifiziert. Diese Variable wird benutzt, um Koordinaten anzugeben, die beim Zeichnen des Hosts im statuswrl-CGI verwendet werden.</p>
<p>3d coords: alias:</p>	<p>Diese Direktive benennt eine positive oder negative reelle Zahl als Ursprung für die Zeichnung der Hostgruppe. Die Größe des Host-angeboters kann mit Sie, um die Größe des Host-angeboters zu bestimmen. 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 <i>hostgroups</i>-Direktive in den Host-Definitionen verwendet werden.</p>
<p>members:</p>	
<p>hostgroup hostgroup_members:</p>	<p>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.</p>
<p>Hostgruppen-Definition</p>	
<p>notes:</p>	<p>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).</p>

<p>notes_url:</p>	<p>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. <i>/cgi-bin/nagios/</i>). Dies kann sehr nützlich sein, wenn Sie detaillierte Infomationen zu dieser Hostgruppe, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.</p>
<p>action_url:</p>	<p>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. <i>/cgi-bin/nagios/</i>).</p>

service

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.

Format der Definition:

Anmerkung: die unterstrichenen Direktiven werden benötigt, die anderen sind optional.

define service{	
<u>host_name</u>	<i>host_name</i>
hostgroup_name	<i>hostgroup_name</i>
<u>service_description</u>	<i>service_description</i>
display_name	<i>display_name</i>
servicegroups	servicegroup_names
is_volatile	[0/1]
<u>check_command</u>	<i>command_name</i>
initial_state	[o,w,u,c]
<u>max_check_attempts</u>	#
<u>check_interval</u>	#
<u>retry_interval</u>	#
active_checks_enabled	[0/1]
passive_checks_enabled	[0/1]
<u>check_period</u>	<i>timeperiod_name</i>
obsess_over_service	[0/1]

check_freshness	[0/1]
freshness_threshold	#
event_handler	<i>command_name</i>
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	<i>timeperiod_name</i>
notification_options	[w,u,c,r,f,s]
notifications_enabled	[0/1]
contacts	<i>contacts</i>
contact_groups	<i>contact_groups</i>
stalking_options	[o,w,u,c]
notes	<i>note_string</i>
notes_url	<i>url</i>
action_url	<i>url</i>
icon_image	<i>image_file</i>
icon_image_alt	<i>alt_string</i>
}	

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 <i>Kurznamen</i> 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 <i>Kurznamen</i> 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 <code>hostgroup_name</code> kann anstatt oder zusätzlich zur <code>host_name</code> -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 <code>host_name</code> und <code>service_description</code> -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 <code>service_description</code> -Direktive benutzt. Anmerkung: Die aktuellen CGIs nutzen diese Option nicht, zukünftige Versionen des Web-Interfaces werden dies tun.
servicegroups:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> 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 <code>members</code> -Direktive in den servicegroup -Definitionen genutzt werden.
is_volatile:	Diese Direktive wird benutzt, um zu kennzeichnen, dass der Service „sprunghaft“ (volatile) 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 = Services sind nicht sprunghaft, 1 = Service sind sprunghaft.
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 service_check_timeout -Option kontrolliert.
initial_state:	Als Default nimmt Nagios 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 Nagios den Service-Prüfbefehl wiederholt, wenn er einen anderen als einen OK-Zustand zurückliefert. Bei einem Wert von 1 wird Nagios 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 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 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 Nagios automatisch einen Frische-Schwellwert festlegen.
event_handler:	Diese Direktive wird benutzt, um den <i>Kurznamen</i> 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 Nagios 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 Nagios 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 Nagios 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/nagios/</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. <code>/cgi-bin/nagios/</code>).
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 40×40 Pixel groß ist. Bilder für Services werden im logos -Unterverzeichnis Ihres HTML-Images-Verzeichnis gesucht (d.h. <code>/usr/local/nagios/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.

servicegroup

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](#).

Format der Definition:

Anmerkung: die unterstrichenen Direktiven werden benötigt, die anderen sind optional.

define servicegroup{	
<u>servicegroup_name</u>	<u>servicegroup_name</u>
<u>alias</u>	<u>alias</u>
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): members=<host1>,<service1>,<host2>,<service2>,...,<hostn>,<servicen>
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).

Eine Kontakt-Definition wird in Ihrem Netzwerk nachfolgend beschrieben. Format der Definition: notes_url: Anmerkung: die unterstrichenen Direktiven werden benötigt, die anderen sind optional	Diese Variable verwendet werden soll. 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. <code>/cgi-bin/nagios/</code>). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu dieser Servicegruppe, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.	Die Variable verwendet werden soll. Die verschiedenen Parameter dieser Kontakt-Definition werden benutzt, 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. <code>/cgi-bin/nagios/</code>).
<code>define contact{</code>	Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung	
<code>contact_name</code>	<code>contact_name</code>	
<code>alias</code>	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“	
<code>contactgroups</code>	<code>contactgroup_names</code>	
<code>host_notifications_enabled</code>	[0/1]	
<code>action_url:</code>		
<code>service_notifications_enabled</code>	[0/1]	
<code>host_notification_period</code>	<code>timeperiod_name</code> (d.h. <code>/cgi-bin/nagios/</code>).	
<code>service_notification_period</code>	<code>timeperiod_name</code>	
<code>host_notification_options</code>	[d,u,r,f,s,n]	
<code>service_notification_options</code>	[w,u,c,r,f,s,n]	
<code>host_notification_commands</code>	<code>command_name</code>	
contact		
<code>service_notification_commands</code>	<code>command_name</code>	
<code>email</code>	<code>email_address</code>	
Kontakt-Definition	<code>pager_number or pager_email_gateway</code>	
<code>pager</code>		
<code>addressx</code>	<code>additional_contact_address</code>	
<code>can_submit_commands</code>	[0/1]	
<code>retain_status_information</code>	[0/1]	
<code>retain_nonstatus_information</code>	[0/1]	
<code>}</code>		

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.xyxy@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 <code>\$CONTACTNAME\$-Makro</code> 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 <code>\$CONTACTALIAS\$-Makro</code> diesen Alias/diese Beschreibung enthalten. Falls nicht angegeben, wird der <code>contact_name</code> als Alias verwendet.
contactgroups:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> 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 <i>members</i> -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 <i>Kurznamen</i> von Befehlen anzugeben, die zur Benachrichtigung von Kontakten über <i>Host</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.

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 Fluttern 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, u = Benachrichtigungen bei einem UNKNOWN-Zustand versenden, r = Benachrichtigungen bei Erholungen (OK-Zustand) versenden, f = Benachrichtigungen versenden, wenn der Service mit Fluttern 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-Probleme</i> 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 <code>\$CONTACTEMAIL\$-Makro</code> 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. <code>pagejoe@pagenet.com</code>) sein. Abhängig von Ihren Benachrichtigungsbefehlen kann sie benutzt werden, um eine Alarm-Page an den Kontakt zu versenden. Bei korrekter Anwendung wird das <code>\$CONTACTPAGER\$-Makro</code> 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 <code>\$CONTACTADDRESSx\$-Makro</code> wird diesen Wert enthalten.

can_submit_commands:	Diese Direktive wird benutzt, um festzulegen, ob dieser Kontakt über die CGIs externe Befehle an Nagios 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.

contactgroup

Kontaktgruppen-Definition
Beschreibung:

Eine Kontaktgruppen-Definition wird benutzt, um einen oder mehrere [Kontakte](#) zu gruppieren, um Alarm-/Erholungs-[Benachrichtigungen](#) zu versenden.

Format der Definition:

Anmerkung: die unterstrichenen Direktiven werden benötigt, die anderen sind optional.

define contactgroup{	
<u>contactgroup_name</u>	<i>contactgroup_name</i>
<u>alias</u>	<i>alias</i>
<u>members</u>	<i>contacts</i>
<u>contactgroup_members</u>	<i>contactgroups</i>
}	

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.

timeperiod

Zeitfenster-Definition

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.

Format der Definition:

Anmerkung: die unterstrichenen Direktiven werden benötigt, die anderen sind optional.

define timeperiod{	
<u>timeperiod_name</u>	<i>timeperiod_name</i>
<u>alias</u>	<i>alias</i>
[weekday]	<i>timeranges</i>
[exception]	<i>timeranges</i>
exclude	[<i>timeperiod1,timeperiod2,...,timeperiodn</i>]
}	

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 („ <i>sunday</i> “ bis „ <i>saturday</i> “) 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 („ <i>Sunday</i> “ bis „ <i>Saturday</i> “). 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]:	<p>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, zeige ich Ihnen anhand der o.g.</p> <p>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.</p>
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.

command

Befehls-Definition

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.

Format der Definition:

Anmerkung: die unterstrichenen Direktiven werden benötigt, die anderen sind optional.

define command{	
<u>command_name</u>	<i>command_name</i>
<u>command_line</u>	<i>command_line</i>
}	

Beschreibung:

Last update:

2011/06/12 wiki:programme:nagios:definitionen https://wiki.da-checka.de/doku.php/wiki/programme/nagios/definitionen?rev=1307876160

Service-Abhängigkeiten sind ein fortgeschrittenes Feature von Nagios, 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](#).

define command

Format der Definition:

```
command_name check_pop
command_line /usr/local/nagios/libexec/check_pop -H $HOSTADDRESS$
```

Anmerkung: die unterstrichenen Direktiven werden benötigt, die anderen sind optional. Trotz allem müssen Sie mindestens ein Kriterium angeben, damit die Definition von Nutzen ist.

Beschreibung der Direktiven:

define servicedependency{	
dependent_host_name	Diese Direktive ist der Kurzname, der zur Identifizierung des Befehls benutzt wird.
dependent_hostgroup_name	Ein Name, der auf Hostnamen, Hostgruppen und Service-Definitionen (in notification-, check-, und eventhandler-Direktiven) verwendet.
dependent_service_description	Die Beschreibung des abhängigen Service.
host_name	Diese Direktive wird benutzt, um zu definieren, was wirklich durch Nagios 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 <i>nicht</i> 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).
hostgroup_name	Die Hostgruppe, die benutzt wird, um zu definieren, was wirklich durch Nagios 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 <i>nicht</i> 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).
service_description	Die Beschreibung des abhängigen Service.
inherits_parent	Ersetzt die Beschreibung des Eltern-Service durch die Beschreibung des abhängigen Service.
execution_failure_criteria	Die Kriterien, die verwendet werden, um zu bestimmen, ob ein Service als ausgefallen betrachtet werden soll. Die Kriterien sind w, u, c.
notification_failure_criteria	Die Kriterien, die verwendet werden, um zu bestimmen, ob ein Service als ausgefallen betrachtet werden soll. Die Kriterien sind w, u, c.
dependency_period	Die Periode, die verwendet wird, um zu bestimmen, ob ein Service als ausgefallen betrachtet werden soll.
}	
command line:	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 <i>command_line</i> -Direktive auf das entsprechende \$USER\$-Makro verweisen.
Beispieldefinition:	Wenn Sie während der Laufzeit Parameter an Befehle übergeben wollen, können Sie die **\$ARGn\$** -Makros in der <i>command_line</i> -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 .
define servicedependency{	
host_name	www1
service_description	Apache Web Server
dependent_host_name	www1
dependent_service_description	Main Web Site
execution_failure_criteria	w,u,c
notification_failure_criteria	w,u,c
}	

Beschreibung der Direktiven: servicedependency

Service-Abhängigkeits-Definition dependent_host_name:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> des/der Hosts anzugeben, auf dem der <i>abhängige</i> Service „läuft“ oder mit dem er verbunden ist. Mehrere Hosts werden durch Kommata von einander getrennt. Bleibt die Direktive leer, so kann dadurch eine <i>"same host"-Abhängigkeit</i> erstellt werden.
dependent_hostgroup_name:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> der Hostgruppe(n) anzugeben, auf der/denen der <i>abhängige</i> Service „läuft“ oder mit dem er verbunden ist. Mehrere Hostgruppen werden durch Kommata von einander getrennt. Die <i>dependent_hostgroup</i> -Direktive kann statt der (oder zusätzlich zur) <i>dependent_host</i> -Direktive benutzt werden.
dependent_service_description:	Diese Direktive wird benutzt, um die <i>Beschreibung</i> des <i>abhängigen</i> 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 <code>hostgroup_name</code> kann statt oder zusätzlich zur <code>host_name</code> -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 <i>Master-Service</i> 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.

Beschreibung:

Last update:

2011/06/12 wiki:programme:nagios:definitionen https://wiki.da-checka.de/doku.php/wiki/programme/nagios/definitionen?rev=1307876160

Serviceeskalationen sind *komplett optional* und werden benutzt, um Benachrichtigungen für einen bestimmten Service zu eskalieren. Mehr Informationen, wie Eskalationen arbeiten, finden Sie [hier](#).

Format der Definition:

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 *benötigt* die *gültig*.

Anmerkung: die unterstrichenen Direktiven sind *benötigt* die *gültig* sind optional.

define serviceescalation{	
<u>host_name</u>	<i>host_name</i>
<u>hostgroup_name</u>	<i>hostgroup_name</i>
<u>service_description</u>	<i>service_description</i>
<u>contacts</u>	<i>contacts</i>
<u>contact_groups</u>	<i>contactgroup_name</i>
<u>first_notification</u>	#
<u>last_notification</u>	#
<u>notification_interval</u>	#
<u>escalation_period</u>	timeperiod_name
<u>escalation_options</u>	[w,u,c,r]
}	

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,themanagers
}
```

Beschreibung der Direktiven:

host_name:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> 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 <code>hostgroup_name</code> kann statt oder zusätzlich zur <code>host_name</code> -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 Serviceeskalations-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 Serviceeskalations-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 Nagios 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.

die unterstrichenen Direktiven werden benötigt, die anderen sind optional.

hostdependency

Host-Abhängigkeits-Definition

Beschreibung:

Host-Abhängigkeiten sind ein fortgeschrittenes Feature von Nagios, 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](#).

Format der Definition:

define hostdependency{	
<u>dependent_host_name</u>	<i>host_name</i>
<u>dependent_hostgroup_name</u>	<i>hostgroup_name</i>

host_name	<i>host_name</i>
hostgroup_name	<i>hostgroup_name</i>
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 <i>dependent_hostgroup_name</i> kann statt oder zusätzlich zur <i>dependent_host_name</i> -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 <i>hostgroup_name</i> kann statt oder zusätzlich zur <i>host_name</i> -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.

<p>execution_failure_criteria:</p>	<p>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.</p>
<p>notification_failure_criteria:</p>	<p>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.</p>
<p>dependency_period:</p>	<p>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.</p>

hostescalation

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](#).
 Format der Definition:

Anmerkung: die unterstrichenen Direktiven werden benötigt, die anderen sind optional.

```
define hostescalation{
    host_name      host_name
    hostgroup_name hostgroup_name
```

<u>contacts</u>	<i>contacts</i>
<u>contact_groups</u>	<i>contactgroup_name</i>
<u>first_notification</u>	#
<u>last_notification</u>	#
<u>notification_interval</u>	#
<u>escalation_period</u>	timeperiod_name
<u>escalation_options</u>	[d,u,r]
	}

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 <i>Kurznamen</i> des/der Hosts anzugeben, für den die Eskalation gilt.
hostgroup_name:	Diese Direktive wird benutzt, um den/die <i>Kurznamen</i> 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 Mitglieder der angegebenen Hostgruppe(n) sind.
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 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 Nagios 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 angegeben 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.

hostextinfo

erweiterte Hostinformations-Definition
Beschreibung:

Einträge für erweiterte Hostinformationen sind grundsätzlich dazu gedacht, die Ausgaben der [status](#)-, [statusmap](#)-, [statuswrl](#)- und [extinfo](#)-CGIs schöner aussehen zu lassen. Sie haben keinen Einfluss auf die Überwachung und sind vollständig optional.

Hinweis:



Ab Nagios 3.x sind alle Direktiven der erweiterten Hostinformations-Definition 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.

Format der Definition:

Anmerkung: die unterstrichenen 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
    vrml_image   novell40.png
    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. <i>/cgi-bin/nagios/</i>). 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. <i>/cgi-bin/nagios/</i>). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu diesem Host, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.
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 40×40 Pixel groß ist. Bilder für Hosts werden im logos -Unterverzeichnis Ihres HTML-Images-Verzeichnis gesucht (d.h. <i>/usr/local/nagios/share/images/logos</i>).
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.
vrml_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 als Textur-Map für den angegebenen Host in der statuswrl -CGI benutzt. Anders als das Bild, das Sie in der <code><icon_image></code> -Variable angeben, sollte dieses möglichst <i>keinerlei</i> Transparenz haben. Wenn es das tut, wird das Host-Objekt ein wenig komisch aussehen. Bilder für Hosts werden im logos -Unterverzeichnis Ihres HTML-Images-Verzeichnis gesucht (d.h. <i>/usr/local/nagios/share/images/logos</i>).
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 ich würde 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 <i>unkomprimierten</i> Format erstellt, um die CPU-Belastung zu minimieren, während das Statusmap-CGI das Netzwerkartenbild erstellt. Das Bild wird am besten aussehen, wenn es 40×40 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-Verzeichnis gesucht (d.h. <i>/usr/local/nagios/share/images/logos</i>).
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 40×40 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 größtlichen X- und Y-Koordinaten festlegen, die Sie angegeben haben.
3d_coords:	Diese Variable wird benutzt, um Koordinaten anzugeben, die beim Zeichnen des Hosts im statuswrl -CGI verwendet werden. Koordinaten können positive oder negative reelle Zahlen sein. Der Ursprung für die Zeichnung ist (0.0,0.0,0.0). Die Größe des Host-Kubus ist 0,5 Einheiten auf jeder Seite (Text benötigt etwas mehr Platz). Die Koordinaten, die Sie hier angeben, beziehen sich auf das Zentrum des Host-Kubus.

Beschreibung:

Last update:

2011/06/12 wiki:programme:nagios:definitionen <https://wiki.da-checka.de/doku.php/wiki/programme/nagios/definitionen?rev=1307876160>

Frage 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.

erweiterte Serviceinformations-Definition

Hinweis:



Ab Nagios 3.x sind alle Direktiven der erweiterten Serviceinformations-Definition 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.

Format der Definition:

Anmerkung: die unterstrichenen Direktiven werden benötigt, die anderen sind optional. Trotz allem müssen Sie mindestens ein Kriterium angeben, damit die Definition von Nutzen ist.

define serviceextinfo{	
<u>host_name</u>	<i>host_name</i>
<u>service_description</u>	<i>service_description</i>
notes	<i>note_string</i>
notes_url	<i>url</i>
action_url	<i>url</i>
icon_image	<i>image_file</i>
icon_image_alt	<i>alt_string</i>
}	

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 <i>Kurznamen</i> 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. <i>/cgi-bin/nagios/</i>). 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. <i>/cgi-bin/nagios/</i>). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu diesem Host, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.
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 40×40 Pixel groß ist. Bilder für Service werden im logos -Unterverzeichnis Ihres HTML-Images-Verzeichnis gesucht (d.h. <i>/usr/local/nagios/share/images/logos</i>).
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.

From:

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

Permanent link:

<https://wiki.da-checka.de/doku.php/wiki/programme/nagios/definitionen?rev=1307876160>

Last update: 2011/06/12 12:56

