

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 `pr Direktiv` zu verwenden 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>
parents	<i>host_names</i>
hostgroups	<i>hostgroup_names</i>
check_command	<i>command_name</i>
initial_state	[o,d,u]
max_check_attempts	#
check_interval	#
retry_interval	#
active_checks_enabled	[0/1]
passive_checks_enabled	[0/1]
host _ <u>period</u>	<i>timeperiod_name</i>
obsess_over_host	[0/1]
check_freshness	[0/1]
<u>freshness_threshold</u>	#
<u>event_handler</u>	<i>command_name</i>
event_handler_enabled	[0/1]
low_flap_threshold	#
high_flap_threshold	#
flap_detection_enabled	[0/1]
flap_detection_options	[o,d,u]
process_perf_data	[0/1]
retain_status_information	[0/1]
retain_nonstatus_information	[0/1]
<u>contacts</u>	<i>contacts</i>
<u>contact_groups</u>	<i>contact_groups</i>
<u>notification_interval</u>	#
first_notification_delay	#
<u>notification_period</u>	<i>timeperiod_name</i>
notification_options	[d,u,r,f,s]
notifications_enabled	[0/1]
stalking_options	[o,d,u]
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>
vrml_image	<i>image_file</i>

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.

Definitionen

[Host-Definitionen](#)
[Hostgruppen-Definitionen](#)
[Service-Definitionen](#)

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 <i>interval_length</i> -Direktive mit einem Default-Wert von 60 nicht verändert haben, wird diese Zahl Minuten bedeuten. Mehr Informationen zu diesem Wert finden Sie in der check scheduling -Dokumentation.
retry_interval:	Diese Direktive wird benutzt, um die Anzahl von „Zeiteinheiten“ zu definieren, die zwischen erneuten Überprüfungen 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 <i>interval_length</i> -Direktive mit einem Default-Wert von 60 nicht verändert haben, wird diese Zahl Minuten bedeuten. Mehr Informationen zu diesem Wert finden Sie in der check scheduling -Dokumentation.
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=1307868553>

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 <u>Direktive werden benötigt, da andere sind optional</u>. 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 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 andere sind optional</p>	
<pre>define hostgroup{ hostgroup_name statusmap_image: alias members hostgroup_members notes notes_url action_url }</pre>	
<p>Beispieldefinition:</p>	
<pre>define hostgroup{ 2d_coords hostgroup_name alias members }</pre>	<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>
<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 kann eine positive oder negative reelle Zahl sein. Der Ursprung für die Zeichnung ist die Hostgruppe (0,0,0). Die Größe des Host-angeboters kann mit Sie, um die Host-Gruppe Text identifizieren mehr Platz). Die Koordinaten, die Sie hier angeben, beziehen sich auf das Zentrum des Host-Klaus.</p>
<p>members:</p>	<p>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>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. <i>/cgi-bin/nagios/</i>). 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. <i>/cgi-bin/nagios/</i>).
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. <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 <i><icon_image></i> -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).

notes_url:	Diese Variable wird benutzt, um einen optionalen URL anzugeben, der benutzt werden kann, um weitere Informationen zu dieser Servicegruppe zu liefern. Wenn Sie einen URL angeben, werden Sie ein rotes Verzeichnis-Icon in den CGIs sehen (wenn Sie Servicegruppen-Informationen betrachten), das auf den URL verweist, den Sie hier angeben. Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. <i>/cgi-bin/nagios/</i>). Dies kann sehr nützlich sein, wenn Sie detaillierte Informationen zu dieser Servicegruppe, Notfallkontaktmethoden usw. für anderes Support-Personal zur Verfügung stellen wollen.
action_url:	Diese Direktive wird benutzt, um einen optionalen URL anzugeben, der benutzt werden kann, um weitere Aktionen für diese Servicegruppe zu ermöglichen. Wenn Sie einen URL angeben, werden Sie einen roten „Klecks“ in den CGIs sehen (wenn Sie Servicegruppen-Informationen betrachten). Jeder gültige URL kann benutzt werden. Wenn Sie relative Pfade benutzen, wird der Basis-Pfad der gleiche sein, der benutzt wird, um auf die CGIs zuzugreifen (d.h. <i>/cgi-bin/nagios/</i>).

From:

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

Permanent link:

<https://wiki.da-checka.de/doku.php/wiki/programme/nagios/definitionen?rev=1307868553>

Last update: 2011/06/12 10:49

