第 3 章 Nagios 3.0 新特性

重要

Important: Make sure you read through the documentation and the FAQs at http://www.nagios.org/ before sending a question to the mailing lists.

3.1. 更新日志

Nagios 的更新日志可以在这里的在线文件或是在源程序的发行包的根目录里找到。

3.2. 变更与新特征

- 文档：
  1. 更新了文档 — 很抱歉我对文档的更新工作进展迟缓。这会花些时间来做，因为有很多文档而且写这些文档并不是我喜欢的事情（我更不喜欢整天翻译，这也不是我喜欢的事情）。期待一些文档与其他的有所不同，而这些不同会对于那些新人或有经验的 Nagios 使用者起些作用。
- 内嵌宏：
  1. 新 加宏 — 加入了一些新宏，包 括：$TEMPPATH$、$LONGHOSTOUTPUT$、$LONGSERVICEOUTPUT$、$HOSTNOTIFICATIONID$、$SERVICENOTIFICATIONID$、$HOSTEVENTID$、$SERVICEEVENTID$、$SERVICEISVOLATILE$、$LASTHOSTEVENTID$、$LASTSERVICEEVENTID$、$HOSTDISPLAYNAME$、$SERVICEDISPLAYNAME$、$MAXHOSTATTEMPTS$、$MAXSERVICEATTEMPTS$、$TOTALHOSTSERVICES$、$TOTALHOSTSERVICESOK$、$TOTALHOSTSERVICESWARNING$、$TOTALHOSTSERVICESUNKNOWN$、$TOTALHOSTSERVICESCRITICAL$、$CONTACTGROUPNAME$、$CONTACTGROUPNAMES$、$CONTACTGROUPALIAS$、$CONTACTGROUPMEMBERS$、$NOTIFICATIONRECIPIENTS$、$NOTIFICATIONISESCALATED$、$NOTIFICATIONAUTHOR$、$NOTIFICATIONAUTHORNAME$、$NOTIFICATIONAUTHORALIAS$、$NOTIFICATIONCOMMENT$、$EVENTSTARTTIME$、$HOSTPROBLEMID$、$LASTHOSTPROBLEMID$、$SERVICEPROBLEMID$、$LASTSERVICEPROBLEMID$、$LASTHOSSTATE$、$LASTHOSTSTATEID$、$LASTSERVICESTATE$、$LASTSERVICESTATEID$。 加入了两个特殊的守护时间宏：$ISVALIDTIME:$和$NEXTVALIDTIME:$。

2. 移除的宏 — 原有的宏$NOTIFICATIONNUMBER$被分离为两个新宏$HOSTNOTIFICATIONNUMBER$和$SERVICENOTIFICATIONNUMBER$。
3. 变更的宏 — 现有的$HOSTNOTES$和$SERVICENOTES$宏包括自身外，还包括$HOSTNOTESURL$、$HOSTACTIONURL$、$SERVICENOTESURL$和$SERVICEACTIONURL$等几个宏。
4. 在检测、事件句柄处理、告警和其他外部命令执行时，宏可以获取环境变量。这可会使 Nagios 在大型部署方案时占用较高的 CPU 处理能力，你可以设置 enable_environment_macros 选项来不使能它。
5. 有关宏的更新信息可以在这里查到。

- 预定义停机时间：
  1. 预定义停机时间不再保存在各自文件(之前是由主配置文件里的 **downtime_file** 来指定)。当前的和保留的预定义停机时间将分别保存于状态文件和保留文件 retention file 中。
- 注释：
  1. 主机和服务的注释不再保存于各自的文件(之前在主配置文件中的 **comment_file** 来指定)。当前的和保留的注释将分别保存于状态文件 status file 和保留文件 retention file 之中。
  2. Acknowledgement comments that are marked as non-persistent are now only deleted when the acknowledgement is removed. They were previously automatically deleted when Nagios restarted, which was not ideal.
- State Retention Data:
  1. Status information for individual contacts is now retained across program restarts.
  2. Comment and downtime IDs are now retained across program restarts and should be unique unless the retention data is deleted or ignored.
  3. Added retained_host_attribute_mask and retained_service_attribute_mask variables to control what host/service attributes are retained globally across program restarts.
  4. Added retained_process_host_attribute_mask and retained_process_service_attribute_mask variables to control what process attributes are retained across program restarts.
  5. Added retained_contact_host_attribute_mask and retained_contact_service_attribute_mask variables to control what contact attributes are retained globally across program restarts.
- Flap Detection:
  1. Added **flap_detection_options** directive to host and service definitions to allow you to specify what host/service states should be used by the flap detection logic (by default all states are used).
  2. Percent state change and state history are now retained and recorded even when flap detection is disabled.
  3. Hosts and services are immediately checked for flapping when flap detection is enabled program-wide.
  4. Hosts and services that are flapping when flap detection is disabled program-wide are now logged.

5. More information on flap detection can be found here.

- External Commands:
  1. Added a new PROCESS_FILE external command to allow processing of external commands found in an eternal (regular) file. Useful for processing large amounts of passive checks with long output, or for scripting regular commands. More information can be found here.
  2. Custom commands may now be submitted to Nagios. Custom command names are prefixed with an underscore and are not processed internally by the Nagios daemon. They may, however, be processed by a loaded NEB module.
  3. The check_external_commands option is now enabled by default, which means Nagios is configured to check for external "commands out of the box". All 2.x and earlier versions of Nagios had this option disabled by default.
- Status Data:
  1. Contact status information (last notification times, notifications enabled/disabled, etc.) is now saved in the status and retention files, although it is not processed by the CGIs.
- Embedded Perl:
  1. Added new enable_embedded_perl and use_embedded_perl_implicitly variables to control use of the embedded Perl interpreter.
  2. Perl scripts/plugins can now explicitly tell Nagios whether or not they should be run under the embedded Pel interpreter. This is useful if you have troublesome scripts that don't function well under the ePN.
  3. More information about these new optios can be found here.
- Adaptive Monitoring:
  1. The check timeperiod for hosts and services can now be modified on-the-fly with the appropriate external command (CHANGE_HOST_CHECK_TIMEPERIOD or CHANGE_SVC_CHECK_TIMEPERIOD).查阅这个网页以取得更多可用的适应性检测命令。
- Notifications:
  1. A **first_notification_delay** option has been added to host and service definitions to (what else) introduce a delay between when a host/service problem first occurs and when the first problem notification goes out. In previous versions you had to use some mighty config-fu with escalations to accomplish this. Now this feature is available to normal mortals.
  2. Notifications are now sent out for hosts/services that are flapping when flap detection is disabled on a host- or service-specific basis or on a program-wide basis. The $NOTIFICATIONTYPE$ macro will be set to "FLAPPINGDISABLED" in this situation.
  3. Notifications can now be sent out when scheduled downtime start, ends, and is cancelled for hosts and services. The $NOTIFICATIONTYPE$ macro will be set to "DOWNTIMESTART", "DOWNTIMEEND", or "DOWNTIMECANCELLED", respectively. In order to received notifications on scheduled downtime events, specify "s" or "downtime" in your contact, host, and/or service notification options.

4. More information on notifications can be found here.

- Object Definitions:
  1. Service dependencies can now be created to easily define "same host" dependencies for different services on one or more hosts. (Read more)
  2. Extended host and service definitions (hostextinfo and serviceextinfo, respectively) have been deprecated. All values that from extended definitions have been merged with host or service definitions, as appropriate. Nagios 3 will continue to read and process older extended information definitions, but will log a warning. Future versions of Nagios (4.x and later) will not support separate extended info definitions.
  3. New **hostgroup_members**, **servicegroup_members**, and **contactgroup_members** directives have been added to hostgroup, servicegroup, and contactgroups definitions, respectively. This allows you to include hosts, services, or contacts from sub-groups in your group definitions.
  4. New **notes**, **notes_url**, and **action_url** have been added to hostgroup and servicegroup definition.
  5. Contact definitions have the new **host_notifications_enabled**, **service_notifications_enabled**, and **can_submit_commands** directives to better control notifications and determine whether or not they can submit commands through the web interface.
  6. Host and service dependencies now support an optional **dependency_period** directive. This allows you to limit the times during which dependencies are valid.
  7. The **parallelize** directive in service definitions is now deprecated and no longer used. All service checks are run in parallel in Nagios 3.
  8. There are no longer any inherent limitations on the length of host names or service descriptions.
  9. Extended regular expressions are now used if you enable the use_regexp_matching config option. Regular expression matching is only used in certain object definition directives that contain *, ?, +, or \..
  10. A new **initial_state** directive has been added to host and service definitions, so you can tell Nagios that a host/service should default to a specific state when Nagios starts, rather than UP or OK (which is still the default).
- Object Inheritance:
  1. You can now inherit object variables/values from multiple templates by specifying more than one template name in the **use** directive of object definitions. This can allow for some very powerful (and complex) inheritance setups. (Read more)
  2. Services now inherit contact groups, notification interval, and notification period from their associated host if not otherwise specified. (Read more)
  3. Host and service escalations now inherit contact groups, notification interval, and escalation timeperiod fro their associated host or service if not otherwise specified. (Read more)

4. String variables in host, service, and contact definitions can now be prevented from being inherited by specifying a value of "null" (without quotes) for the value of the variable. (Read more)
5. Most string variables in local object definitions can now be appended to the string values that are inherited. This is quite handy in large configurations. (Read more)

- Performance Improvements:
  1. Add ability to precache object config files and exclude circular path detection checks from verification process. This can speed up Nagios start time immensely in large environments! Read more here.
  2. A new use_large_installation_tweaks option has been added that should improve performance in large Nagios installations. Read more about this here.
  3. A number of internal improvements have been made with regards to how Nagios deals with internal data structures and object (e.g. host and service) relationships. These improvements should result in a speedup for larger installations.
  4. New external_command_buffer_slots option has been added to allow you to more easily scale Nagios in large environments. For best results you should consider using MRTG to graph Nagios' usage of buffer slots over time.
- Plugin Output:
  1. Multiline plugin output is now supported for host and service checks. Hooray! The plugin API has been updated to support multiple lines of output in a manner that retains backward compatability with older plugins. Additional lines of output (aside from the first line) are now stored in new $LONGHOSTOUTPUT$ and $LONGSERVICEOUTPUT$ macros.
  2. The maximum length of plugin output has been increased to 4K (from around 350 bytes in previous versions). This 4K limit has been arbitrarily chosen to protect again runaway plugins that dump back too much data to Nagios.
  3. More information on the plugins, multiline output, and max plugin output length can be found here.
- Service Checks:
  1. Nagios now checks for orphaned service checks by default.
  2. Added a new enable_predictive_service_dependency_checks option to control whether or not Nagios will initiate predictive check of service that are being depended upon (in dependency definitions). Predictive checks help ensure that the dependency logic is as accurate as possible. (Read more)
  3. A new cached service check feature has been implemented that can significantly improve performance for many people Instead of executing a plugin to check the status of a service, Nagios can often use a cached service check result instead. More information on this can be found here.
- Host Checks:
  1. Host checks are now run in parallel! Host checks used to be run in a serial fashion, which meant they were a major holdup in terms of performance. No longer! (Read more)

2. Host check retries are now performed like service check retries. That is to say, host definitions now have a new **retry_interval** that specifies how much time to wait before trying the host check again. :-)
3. Regularly scheduled host checks now longer hinder performance. In fact, they can help to increase performance with the new cached check logic (see below).
4. Added a new check_for_orphaned_hosts option to enable checks of orphaned host checks. This is need now that host checks are run in parallel.
5. Added a new enable_predictive_host_dependency_checks option to control whether or not Nagios will initiate predictive check of hosts that are being depended upon (in dependency definitions). Predictive checks help ensure that the dependency logic is as accurate as possible. (Read more)
6. A new cached host check feature has been implemented that can significantly improve performance for many people Instead of executing a plugin to check the status of a host, Nagios can often use a cached host check result instead. More information on this can be found here.
7. Passive host checks that have a DOWN or UNREACHABLE result can now be automatically translated to their proper state from the point of view of the Nagios instance that receives them. This is very useful in failover and distributed monitoring setups. More information on passive host check state translation can be found here.
8. Passive host checks normally put a host into a HARD state. This can now be changed by enabling the passive_host_checks_are_soft option.

- Freshness checks:
  1. A new additional_freshness_latency option has been added to allow to you specify the number of seconds that should be added to any host or service freshness threshold that is automatically calculated by Nagios.
- IPC:
  1. The IPC mechanism that is used to transfer host/service check results back to the Nagios daemon from (grand)child processes has changed! This should help to reduce load/latency issues related to processing large numbers of passive checks in distributed monitoring environments.
  2. Check results are now transferred by writing check results to files in directory specified by the check_result_path option. Files that are older that the max_check_result_file_age option will be mercilessly deleted without further processing.
- Timeperiods:
  1. Timeperiods were overdue for a major overhaul and have finally been extended to allow for date exceptions, skip dates (every 3 days), etc! This should help you out when defining notification timeperiods for pager rotations.
  2. More information on the new timeperiod directives can be found here and here.
- Event Broker:
  1. Updated NEB API version
  2. Modified callback for adaptive program status data
  3. Added callback for adaptive contact status data

4. Added precheck callbacks for hosts and services to allow modules to cancel/override internal host/service checks.

- Web Interface:

  enable_splunk_integrationsplunk_url

  1. Hostgroup and servicegroup summaries now show important/unimportant problem breakdowns liek the TAC CGI.
  2. Minor layout changes to host and service detail views in extinfo CGI.
  3. New check statistics and have been added to the "Performance Info" screen.
  4. Added Splunk
  5. Added new notes_url_target and action_url_target options to control what frame notes and action URLs are opened in.
  6. Added new lock_author_names option to prevent alteration of author names when users submit comments, acknowledgements, and scheduled downtime.
- Deubbing Info:
  1. The DEBUGx compile options available in the configure script for have been removed.
  2. Debugging information can now be written to a separate debug file, which is automatically rotated when it reaches a user-defined size. This should make debugging problems much easier, as you don't need to recompiled Nagios. Full support for writing debugging information to file is being added during the alpha development phase, so it may not be complete when you try it.
  3. Variables that affect the debug log in debug_file, debug_level, debug_verbosity, and max_debug_file_size.
- Misc:
  1. Temp path variable - A new temp_path variable has been added to specify a scratch directory that Nagios can use for temporary scratch space.
  2. Unique notification and event ID numbers - A unique ID number is now assigned to each host and service notification. Another unique ID is now assigned to all host and service state changes as well. The unique IDs can be accessed using the following respective macros: $HOSTNOTIFICATIONID$, $SERVICENOTIFICATIONID$, $HOSTEVENTID$, $SERVICEEVENTID$, $LASTHOSTEVENTID$, $LASTSERVICEEVENTID$.
  3. New macros - A few new macros (other than those already mentioned elsewhere above) have been added. They include $HOSTGROUPNAMES$, $SERVICEGROUPNAMES$, $HOSTACKAUTHORNAME$, $HOSTACKAUTHORALIAS$, $SERVICEACKAUTHORNAME$, and $SERVICEACKAUTHORALIAS$.
  4. Reaper frequency - The old **service_reaper_frequency** variable has been renamed to check_result_reaper_frequency, as it is now also used to process host check results.
  5. Max reaper time - A new max_check_result_reaper_time variable has been added to limit the amount of time a single reaper event is allowed to run.

6. Fractional intervals - Fractional notification and check intervals (e.g. "3.5" minutes) are now supported in host, service, host escalation, and service escalation definitions.
7. Escaped command arguments - You can now pass bang (!) characters in your command arguments by escaping them with a backslash (\). If you need to include backslashes in your command arguments, they should also be escaped with a backslash.
8. Multiline system command output - Nagios will now read multiple lines out output from system commands it runs (notification scripts, etc.), up to 4K. This matches the limits on plugin output mentioned earliar. Output from system commands is not directly processed by Nagios, but support for it is there nonetheless.
9. Better scheduling information - More detailed information is given when Nagios is executed with the -s command line option. This information can be used to help reduce the time it takes to start/restart Nagios.
10. Aggregated status file updates - The old **aggregate_status_updates** option has been removed. All status file updates are now aggregated at a minimum interval of 1 second.
11. New performance data file mode - A new "p" option has been added to the host_perfdata_file_mode and service_perfdata_file_mode options. This new mode will open the file in non-blocking read/write mode, which is useful for pipes.
12. Timezone offset - A new use_timezone option has been added to allow you to run different instances of Nagios in timezones different from the local zone.