Starent Networks Corporation

ST16 Preventative Maintenance Guidelines

Version 1.1

Last updated: 9/15/2008

Dave Damerjian
Starent Networks Corporation
30 International Place
Tewksbury, MA 01876

http://www.starentnetworks.com

Revision History

Issue Date	Rev	Description of Change	Initiated By
8/29/08	1.0	Initial release	Dave Damerjian
9/15/08	1.1	Additional details, cleanup/re-wording, quick reference section	Dave Damerjian

Table Of Contents

Overview

This document gives guidelines on maintaining the ST16 in working order. It is by no means a complete document, but it does seek to cover the most important areas related to routine maintenance.

Most of the information in here can also be found in the user manuals, spread out in many places though not necessarily focused on this topic. So, the greatest value of this

document is that points out the areas that may need attention in one place. But, it is still your responsibility to delve deeper in the documentation in the appropriate sections for more specifics and explanations. Note that although most of the references in this document are to the PDSN Administration Guide, the other manuals are valuable and necessary to consult as well.

ST16 does not stand on its own without interaction with other network components, and so other components are mentioned along with suggestions for maintaining them.

The writing of this guide is from the perspective of someone working in technical support dealing with customer issues on a daily basis. Therefore it should be practical for addressing the everyday issues that would need to be addressed by those maintaining a PDSN. And although this is not a troubleshooting guide, it does touch on material that would be useful for troubleshooting. In summary, this document is designed to be easy to follow, to give solid direction on maintenance, and be useful for learning some basics at the same time.

Chassis Access and Command Line Interface

In order to do the software maintenance mentioned in this document, one will need to have chassis access. ST16 supports telnet, ssh2, or serial connection via cable. You can use any client that supports these applications, and you will need a username and password, preferably one that has been configured with administrative access rights, but at least with Operator rights.

For telnet/ssh, in theory you can specify any ip address on any interface on the chassis, but normally you would specify the management ip address specified in the local context, which is tied to the redundant ports on the active Switch Processor I/O (SPIO) card.

For console access, the following settings should be used: 115 kbps, 8 data bits, No Parity, 1 Stop bit, No flow control.

You may want to save a history of the CLI commands and output from your maintenance sessions for archiving purposes. You may need to enable this capability in your client. Also, you may want to timestamp the running of all commands. This is done by running the timestamps command in Exec mode.

The commands mentioned in this guide are both of the types Exec and Config. So, there could be situations where you will need to enter Config mode, implying that you will be changing the configuration of the chassis. In that case you will need administrative privilege and you will need to feel comfortable with changing a configuration on a live chassis.

See Chapter 1, Command Line Interface Overview for an extensive discussion on chassis access and CLI usage. You will need the basic navigational skills in order to do maintenance.

Password/Credential updates

The ST6 has a password aging facility to force the changing of passwords after a specified time. You can use this feature or keep track of password expirations using some other method.

See Chapter 39, Software Management Operations, specifically section Managing Local-User Administrative Accounts, for more information.

System time

It is critical for proper chassis functioning to maintain the proper time. The A11 protocol for example, used for call establishment, actually depends on the time between the PCF and the PDSN to be within a certain range. Though this range can be expanded with the timestamp-tolerance parameter in the “spi remote-address” of the pdsn service configuration, it is better not to risk the time differential drifting beyond the range. Also, for troubleshooting purposes, having accurate timestamps between chassis that are being troubleshot is critical in avoiding frustration in matching traces from those chassis.

The best solution is to enable NTP (Network Time Protocol) in the local context. Proper maintenance would be making sure not to ignore SNMP traps related to NTP not synching properly. If not using NTP, then check the time on a weekly basis and correct if necessary. As well, specify the timestamp-tolerance especially if not using NTP to account for unexpected time drifts.

Flash File System

The file system on the flash should be maintained as clean as possible. Over time, various types of files may get deposited there. That includes minicore crashes, possible full core crashed if specified in the config, old configurations, old build images, etc. On a monthly basis the file system should be checked and non-needed files deleted. Note that from a size perspective, builds and full crashes take the most space.

Check the boot system priority (command show boot) to see a listing of all configuration files and associated builds that have been specified over the more recent time (though only up to nine past priorities can be specified). If getting close to a priority value of 1,

then be prepared to delete priorities that will never be needed, and to restart numbering at a high value (such as 50).

For lab chassis, if multiple people work on the chassis, putting each person's configurations and personal files in separate folders is not a bad idea.

Make sure that the physical file systems, most importantly the primary and standby SPC/SMCs, are always synchronized if you make any changes to the active SPCs file system. Use the command "card <spc | smc> synchronize filesystem …".

See Chapter 39, Software Management Operations, specifically the section "Maintaining the Local File System", for more information.

Logs and SNMP Traps

The ST16 maintains logs and traps that store a history on the various activities that have occurred on the chassis. The SNMP traps are probably the most useful for quickly determining the health of the chassis and the history of what has occurred. This is because they are fairly easy to read and normally only contain the more important events taking up minimal space. The trap entries are a short version of the full traps that are actually generated. Logging on the other hand can be very verbose and in many cases difficult or impossible to read for someone not (intimately) familiar with the underlyings and architecture of the chassis. In some cases log entries will make sense, while in other case only Starent engineering will be able to interpret their meanings.

SNMP Traps

SNMP traps can be viewed with the command "show snmp trap history". Through v7.1, up to 400 traps can be stored on the chassis as first in first out (FIFO) – beyond 400 is dropped. Starting with v8, the history has been significantly increased to 5000. This was done because we often encounter scenarios where issues are not reported or noticed until many days after they have occurred, or an issue has happened multiple times over an extended period.

The SNMP trap history can be cleared, though doing so does not buy you any benefit because the history is still limited to 400 maximum and will always display as FIFO, so this is not considered part of maintenance. The command to clear traps is "clear snmp trap history".

One thing that can be done is to suppress certain traps that are not necessarily interesting, thereby making more room for traps that are more important to capture by extending the timeframe used by the traps. Use the command "snmp trap suppress …". You can also

restrict the number of a certain trap to be sent over a specified period with “snmp notif-threshold …”

The chassis may be configured to send SNMP traps to a SNMP server application and/or Starent EMS server. This is configured in the local context with commands “snmp target …” , “snmp community …”, etc. The trap server should be maintained to be able to store traps for an extended period of time. One month should be an absolute minimum, while 3 months, or 6 months if space allows, is recommended. The trap server needs to be maintained to be able to handle the trap volume it receives, often from multiple sources besides STxx, and it needs to be configured to delete beyond the time frame that has been decided upon.

Significant traps that should not be ignored include AAA Unreachable, ManagerFailure or TaskFailed (pointing to potential crashes), BGPPeerSessionDown, SRPConfigOutOfSync, CardSPOFAlarm, PortLinkDown, CardOffline.

See Chapter 5, Configuring Management Settings, for more information, as well as the SNMP MIP Reference manual.

System Logging

Logs are stored on the chassis in memory as FIFO. ALL system logs can be viewed with the command “show logs”. It can be further qualified to limit output to a certain level of logging with the “level” qualifier. The range is all logs to just critical logs using qualifiers: debug, trace, info, unusual, warning, error, critical.

The logging level is set in global config mode with the command “logging filter runtime facility <facility> level <level> (critical-info | no-critical-info)”. By default, the level for all facilities is “error”. When troubleshooting a specific problem, Starent may recommend turning on various levels of logging for various facilities. Each facility to be configured is done so separately. Normally only a couple facilities would need to have logging enabled (beyond the default error level). View the current settings in Exec mode using the command “show logging”.

Logs can fill up the system quickly when enabled beyond the error level for any facility. When finished troubleshooting (whatever the time frame may be – hours or days or weeks), one should not forget to disable the logging for the respective facilities. Otherwise, logs unnecessarily fill the buffer space quickly, and when one needs the logs from a certain time frame while troubleshooting another issue, they have already been overwritten with logs that were unnecessarily saved.

The system provides the ability to restrict the sending of a specific event ID or a range of event IDs to minimize the amount of data logged to that which is most useful. Use the command “logging disable eventid …” to further save on buffer space.

Logs can also be saved to a file locally or remotely with the command “save logs …”. If saving locally, the file should be deleted after being moved/saved remotely.

As with SNMP traps, to most safely address logging buffer size constraints, the chassis may be configured to send logging data to a syslog server application so that no data gets lost. This is configured in the local context with command “logging syslog …”. Note that this is not an excuse to not remove unnecessary filters as just discussed, as being able to view the logs directly on the ST16 makes for faster troubleshooting and less wasted efforts and delay in having to retrieve logs from a syslog server. The server should be maintained to be able to store logs for an extended period of time. One month should be an absolute minimum, while 3 months, if space allows, is recommended. The server needs to be maintained to be able to handle the volume it receives, often from multiple sources besides ST16, and it needs to be configured to delete beyond the time frame that has been decided upon.

See Chapter 42, Configuring and Viewing System Logs, for more information.

Crashes list, crash dumps/full core and mini-cores

In the event of a crash, the system is designed to recover, restarting processes, re-establishing connections and communication, possibly switching to a redundant card (PAC/PSC and/or SPC/SMC), etc. Nonetheless, the crash information may be very useful to Starent Support and Engineering for determining if this is a known or new problem, and for helping re-create the problem at Starent and ultimately fix it in a future build. Always report new crash information to Starent so that it can be evaluated for seriousness and to avoid potential issues in the future.

The list of crashes can be viewed with the command “show crash list”, while the details of the crash can be viewed with the command “show crash number X”. Through v6, the system is able to store the history list of up to 30 crashes. But the list is NOT FIFO, and new crashes are simply not saved to the list, which remains static until it is cleared. For these older versions, the crash list should be checked monthly to see if the list of crashes has grown. (Certainly one should be keeping up with any crash occurrences via the SNMP trap history, and report (unknown) issues to Starent.). Often the same crash will happen multiple times, and knowing the frequency can be valuable for troubleshooting and tracking, but the actual crash data is not so useful because you already know about it and have reported it.

So, on a monthly basis you should clear the crash list with the command “clear crash list”, which will also clear the actual crash data at the same time. Running “show support details” will save the crash list along with all the crash data for all the crashes. Otherwise run the commands individually if you want to save the data.

In addition to the crash list and associated crash data, there is abridged crash information known as a mini-core that gets stored locally in the /flash/crash directory. Run the command “dir /flash/crash” to see all the files and the date/times. Match the files with the crash list just described above. Sometimes these can be valuable for Starent Engineering in troubleshooting, though normally the Full crash information (described next) is the most useful. On a monthly basis, clear out the mini-core files, saving any for troubleshooting specific crash issues.

The FULL crash information is very large and may also be useful in saving for analysis by Starent Engineering. This information is too large to store in memory, and so a location can be specified either locally or remotely for storage using the local context config command “crash enable url …”. It is recommended to store remotely so as to avoid the scenario of running out of the memory on the flash. If it is decided to store locally, then on a weekly basis crashes should be checked for, and if found, the crash log file(s) on the flash should be deleted and possibly saved to another location if needed for troubleshooting.

See Chapter 42, Configuring and Viewing System Logs, specifically the section “Configuring and Viewing Software Crash Logging Parameters”, for more information.

Alerts and Alarms

Thresholding is used to monitor the system and alert of potentially bad conditions. Alerts are sent at the end of every polling period where the threshold value set for a given condition is exceeded. Alarms on the other hand are triggered once when a “high” threshold is met, and then cleared when another “low” threshold is met. Alarms can be viewed via the “show alarm all” command, and can be cleared with “clear alarm …”.

Regardless of whether alarm or alert mode is chosen, both SNMP traps and logs will be created. Normally systems monitoring will depend on SNMP traps (more than logs) being sent to a trap server that is being monitored by personnel who are expected to respond with action. Alarms have the extra advantage in that all of the current alarms can be monitored with the show alarm command to give a *quick* snapshot of all issues, whereas with traps and logs, one must review a list and make a determination of what is still currently an issue and what is not. Both have their place. But one should not depend solely on the CLI-based alarm system, that is - the proactive running of the show alarm command on even a very regular basis (i.e. every 15 minutes), to make determinations of

system health, as a lot can happen in short period of time. Traps notifications (and bulkstats data discussed below) should be the primary mechanism to determine that a problem has occurred.

Preventative maintenance involves setting up thresholds that make sense, adjusting thresholds that may trigger unnecessarily, and, this is very important, not ignoring traps and alarms that are being reported.

The seriousness and expected time to react to various traps should vary significantly. A trap server that is able to display the information in a manner that allows for distinguishing severity is important so as to not continually alarm users unnecessarily (i.e. cry wolf), but also to alert with urgency in situations that could quickly become detrimental, is a critical component to a complete PDSN solution.

See Chapter 43, Configuring Thresholding, for more information.

Counters, hardware health check, status, and bulkstats

The ST16 maintains many many counters that are used for statistic gathering and troubleshooting. In general, the counters should be left alone and allowed to increase over time; they are not like other show commands that give the current state (i.e. current number of calls), but rather account for the entire history since the chassis booted. There may be times where it is decided to clear the counters in order to more quickly troubleshoot issues that require looking at and comparing a lot of counters at a glance. The commands to clear start with the word “clear”; See the online help for a list of choices. A partial list of counters to choose from would be:

show port datalink counters
show port npu counters
show radius counters all
show rp statistics
show mipfa statistics
show mipha statistics
show lt2p full statistics
show session disconnect reasons

show session counters historical all (this is an excellent command to see the call volume history for past 3 days)

To check the integrity of the hardware (and associated software), there are a number of commands that would be good to run on a regular basis or when a trap or log notifies that a card (regardless of type), port, fan, or CPU is having problems. The value of running

these commands proactively is that if a trap is missed/overlooked, the issue will still be caught. A partial list of commands to choose from would be:

show power chassis
show fans
show card table all
show hardware inventory
show hardware version board
show hardware card X
show card diag X
show card info X
show leds all
show port table all
show port info <slot/port>
show temperature
show cpu errors verbose
show cpu table

Note that the only component that needs to be proactively checked (and subsequently replaced) in a specific timeframe (6 months), is the air filter.

Many commands exist for checking the health status of various services, processes, CPU load/utilization, call volume, etc. A partial list of basic commands would be:

show session progress
show session duration
show subscriber summary
show port table utilization
show ip pool
show context
show ip interface
show ip route
show task resources
show pdsn-service all
show pdsnclosedrp-service all
show fa-service all
show ha-service all
show lac-service all
show lns-service all
show session recovery status
show active-charging file-space-usage

For inter-chassis session recovery (ICSR):
show srp info

show srp monitor all
show srp checkpoint statistics

There is also a bulk statistics feature that allows pushing a huge array of statistical data to a remote server. Starent makes an EMS server for the purpose of viewing this information in an easy to view manner. Implementing Bulkstats, especially with EMS, can indirectly help with maintenance because the condition of the chassis, even the most minute details, especially over extended periods of time, can be monitored efficiently, accurately, and graphically.

See Chapter 40, Monitoring the System, and Chapter 41, Configuring Bulk Statistics, and the Hardware Maintenance Procedures section of the Hardware Installation Guide, for more information.

Licensing and ip pool utilization

Licensing is mentioned separately here because it is so important that licensing values on the system are high enough to handle the call volume during peak time, or even possibly during failover scenarios where the chassis may need to handle all the traffic from another chassis. Configure the threshold for license utilization with “threshold license-remaining-sessions …”. Related to this is ip pool utilization where the amount of pool resources for a specific pool can be monitored.

Make sure that licenses are applied to the chassis in a timely manner. Do not delay in applying a license – there is no point in not using what you have paid for, and if you delay, you are at risk in running out of licenses eventually. Remember after applying to save the configuration to the current config file (lowest boot priority) AND to do an SPC synchronize to ensure that both SPC/SMCs are updated, otherwise if a switchover occurs, the old limits will be in effect on the card that has been switched to.

See Chapter 43, Configuring Thresholding, and Chapter 39, Software Management Operations, specifically the section Managing License Keys, for more information.

Software Upgrades

One of the best ways to maintain a robust system is to make sure the latest released version/build is active. Starent is continuing to improve the software quality in new releases. Understand the possibility that new bugs may be introduced in new releases, but, in general, you are better off with later versions that should be more robust. Also, fixes for existing bugs will not always be applied to older versions. You may decide that

internal lab testing is required before deploying live, and that is fine, but do not allow these restrictions to result in your moving forward to new versions very infrequently.

Whenever moving a new build onto the chassis, check its integrity with the command "show version /flash/<filename>".

See Chapter 39, Software Management Operations, specifically the section Software Upgrades, for more information.

Quick Reference

This section contains a quick reference of the frequency with which to perform various maintenance operations on the STxx chassis. Note that this is a *very rough guideline* that will vary according to various factors not controlled by Starent, including, but not limited to load on the chassis, number of operators regularly accessing it, importance of the chassis within your network, available staff to perform the maintenance, support level agreements within your organization, the specifics of your chassis configuration, and your own collective experience with the types of issues (subscriber, network, etc.) that are regularly encountered over time.

Constant 24 hour attention

- watch SNMP traps for alarms/thresholds and take appropriate action. The traps will tell you just about all serious problems that can occur on the system, whether or not the STxx is at fault.
- If you have an EMS server or equivalent monitoring system based on bulkstats and other data, pay attention to alarms, call load, etc.

Daily

- analyze system logs for any unusual entries
- look at call volume and throughput for consistency/expected patterns

Weekly

- check clock if NTP not enabled

Monthly

- clean flash file system of files not needed
- clear crash list after saving what you need (you may need to do this more often if getting more crashes)
- clear minicores after saving what you need

6 Months

- change fan filter

Other (no specific time frame)

- if making a config change that you want permanent
    - save to flash
    - SPC/SMC sync
    - especially important for licensing
- expired password
    - re-enable operator as soon as possible
- boot system priority reaching low value
    - reset to make priority a higher value
- if finished troubleshooting with runtime logging
    - remove the logging commands from the config
- Maintain your SNMP trap server as appropriate
- Maintain your syslog server as appropriate