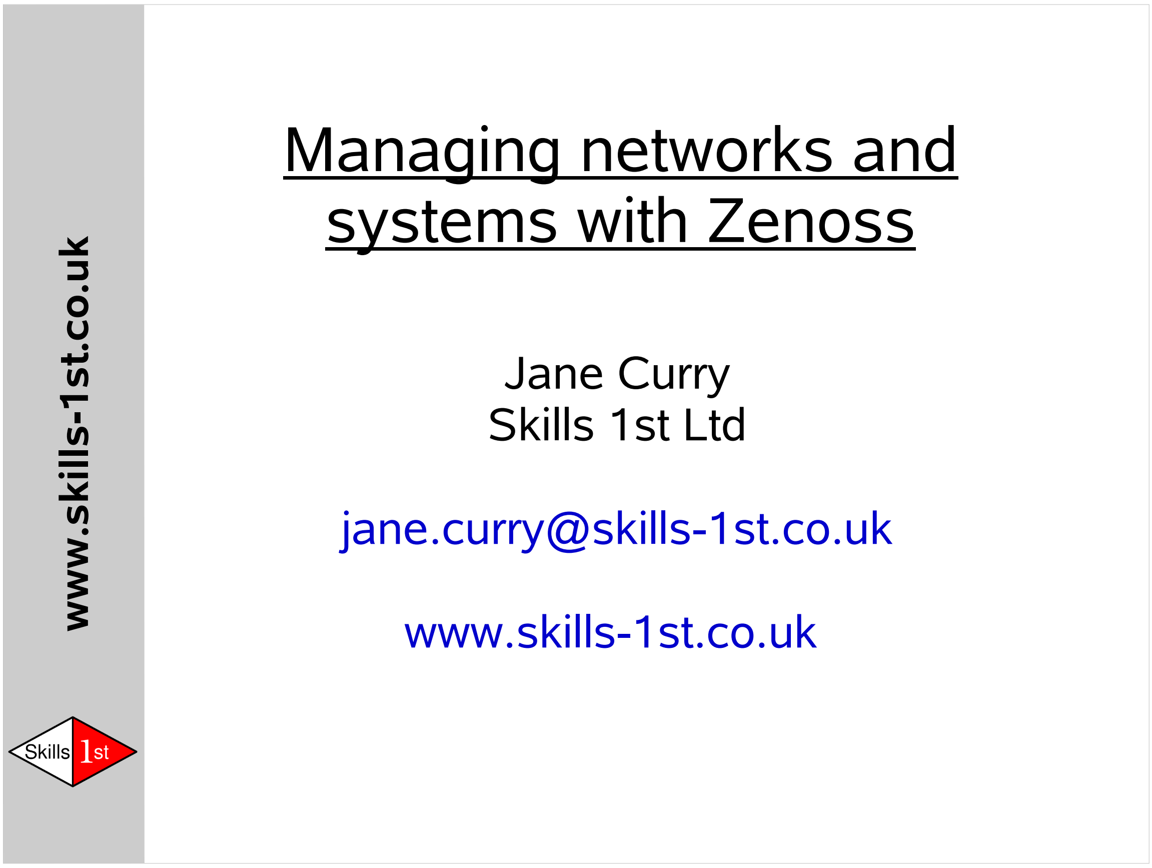


www.skills-1st.co.uk

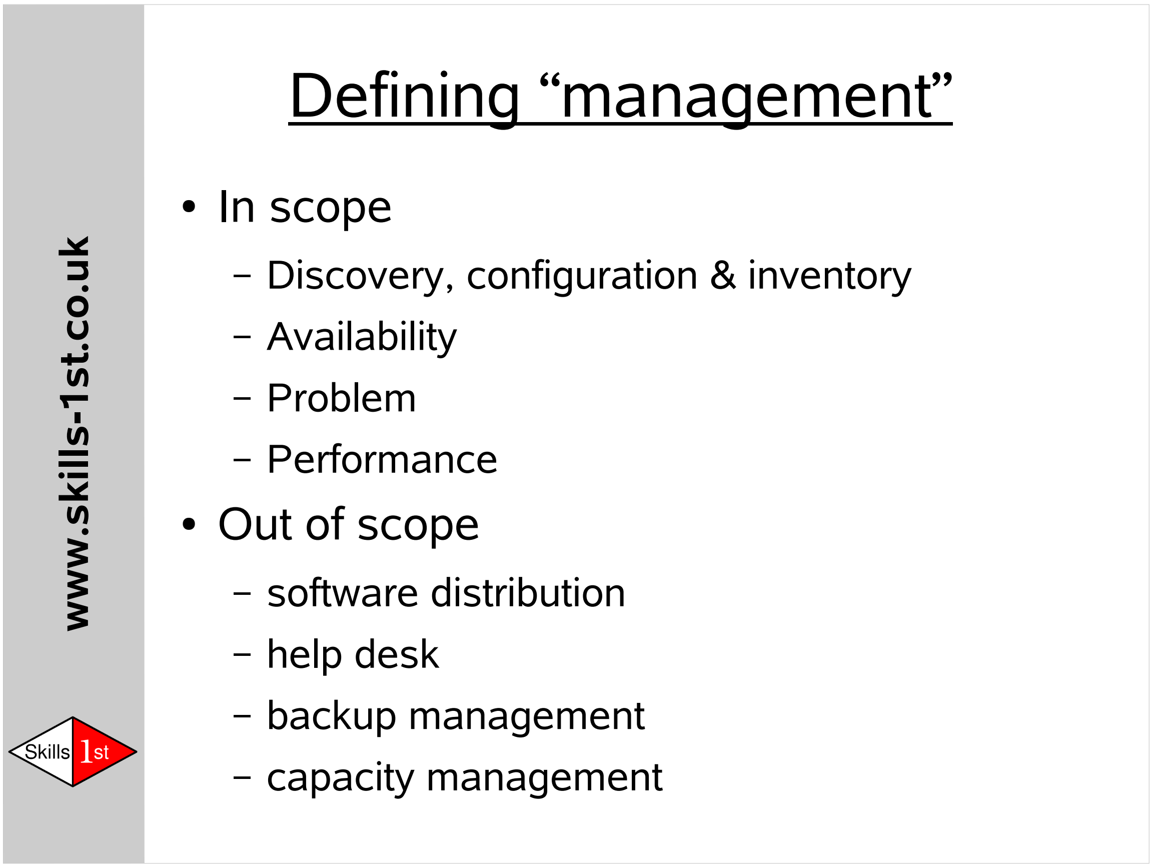
Managing networks and systems with Zenoss

Jane Curry
Skills 1st Ltd

jane.curry@skills-1st.co.uk

www.skills-1st.co.uk

- For the purposes of this paper, Zenoss Core 2.3.2 was used, running on Open SuSE 10.3



www.skills-1st.co.uk

Skills 1st

Defining “management”

- In scope
  - Discovery, configuration & inventory
  - Availability
  - Problem
  - Performance
- Out of scope
  - software distribution
  - help desk
  - backup management
  - capacity management

- Zenoss satisfies all the in-scope requirements

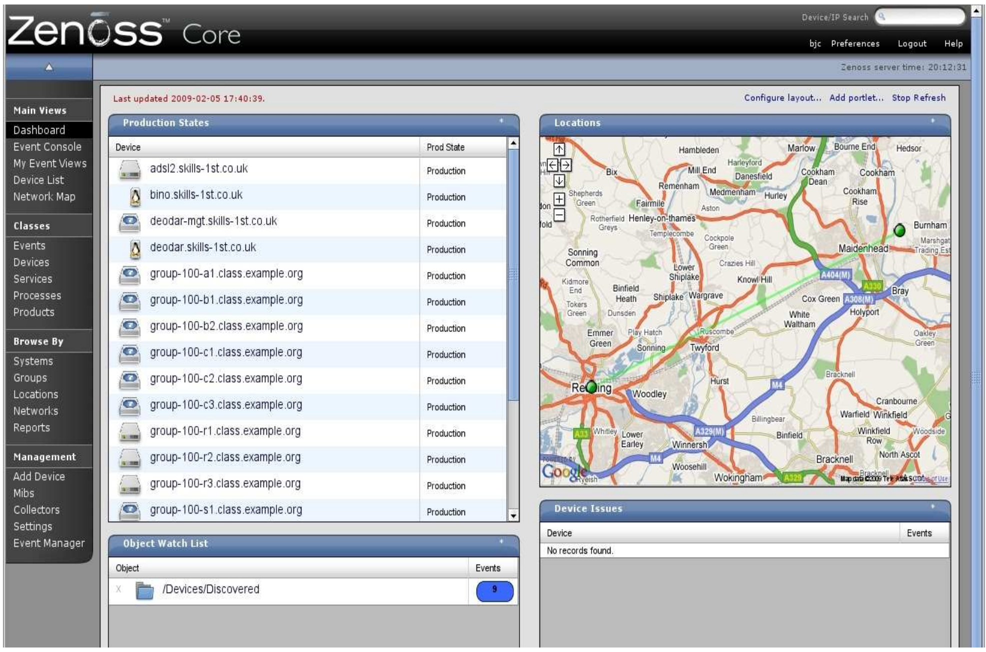
Zenoss at a glance

- Dashboard
- Events Console
- Network Map
- Reports

www.skills-1st.co.uk

Skills 1st

Zenoss Dashboard



Last updated 2009-02-05 17:40:39.

Production States

Device	Prod State
adsl2.skills-1st.co.uk	Production
bino.skills-1st.co.uk	Production
deodar-mgt.skills-1st.co.uk	Production
deodar.skills-1st.co.uk	Production
group-100-a1.class.example.org	Production
group-100-b1.class.example.org	Production
group-100-b2.class.example.org	Production
group-100-c1.class.example.org	Production
group-100-c2.class.example.org	Production
group-100-c3.class.example.org	Production
group-100-r1.class.example.org	Production
group-100-r2.class.example.org	Production
group-100-r3.class.example.org	Production
group-100-s1.class.example.org	Production

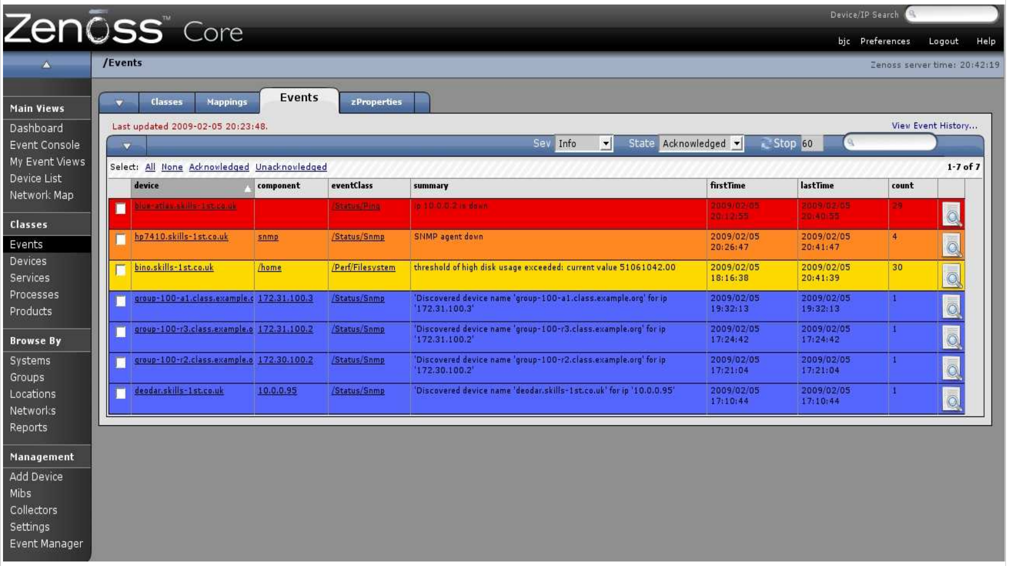
Device	Events
No records found.	

Object	Events
/Devices/Discovered	9

Skills 1st

- Configurable per-user for which portlets are available
- Layout of dashboard is configurable
- Google maps can be used to display Zenoss locations
  - Colours of locations and links reflect health of devices
- Each of the portlets can be configured using the * icon at the top right of the portlet
- Portlets for:
  - Device issues – includes events of severity Critical and Error
  - Google Maps showing Zenoss Locations on google map background
  - Production states showing devices in Production / Pre-Production, Test, Maintenance, Decommissioned
  - Top Level Organizers – can show Device Classes, locations, Systems or Groups
  - Zenoss Issues – show problems with Zenoss daemons
  - Object Watch List – monitors for events in configured device classes

Zenoss Event Console



/Events

Zenoss server time: 20:42:19

Last updated 2009-02-05 20:23:48.

Sev Info State Acknowledged 60

Select: All None Acknowledged Unacknowledged

1-7 of 7

	device	component	eventClass	summary	firstTime	lastTime	count
□	blue-atlas.skills-1st.co.uk		/Status/Ping	ip 10.0.0.2 is down	2009/02/05 20:12:55	2009/02/05 20:40:55	29
□	hp7410.skills-1st.co.uk	snmp	/Status/Snmp	SNMP agent down	2009/02/05 20:26:47	2009/02/05 20:41:47	4
□	bino.skills-1st.co.uk	/home	/Perf/Filesystem	threshold of high disk usage exceeded: current value 51061042.00	2009/02/05 18:16:38	2009/02/05 20:41:39	30
□	group-100-a1.class.example.c	172.31.100.3	/Status/Snmp	'Discovered device name 'group-100-a1.class.example.org' for ip '172.31.100.3'	2009/02/05 19:32:13	2009/02/05 19:32:13	1
□	group-100-r3.class.example.o	172.31.100.2	/Status/Snmp	'Discovered device name 'group-100-r3.class.example.org' for ip '172.31.100.2'	2009/02/05 17:24:42	2009/02/05 17:24:42	1
□	group-100-r2.class.example.o	172.30.100.2	/Status/Snmp	'Discovered device name 'group-100-r2.class.example.org' for ip '172.30.100.2'	2009/02/05 17:21:04	2009/02/05 17:21:04	1
□	deodar.skills-1st.co.uk	10.0.0.95	/Status/Snmp	'Discovered device name 'deodar.skills-1st.co.uk' for ip '10.0.0.95'	2009/02/05 17:10:44	2009/02/05 17:10:44	1

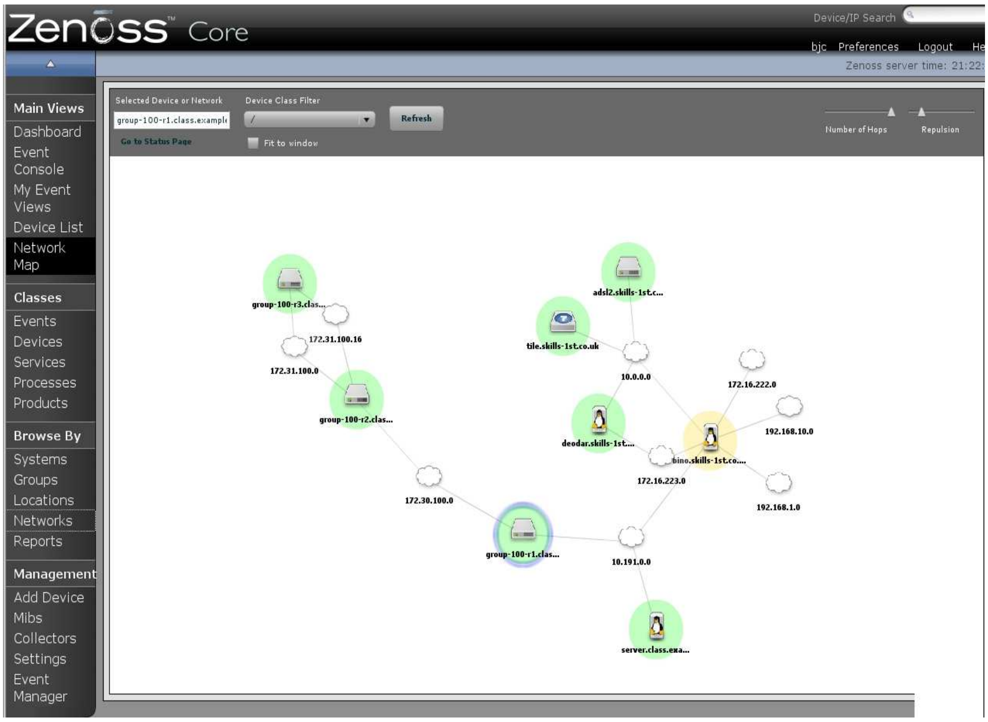
- Zenoss events are held in a MySQL database
  - status table for active events
  - history table for closed events
- Authorised users can Acknowledge and / or Close events
- Event detail available from icon on right
- Displayed fields can be controlled
- Sort events by clicking on column header
- “View Event History” link (top right) to swap to closed events
- Access to events from many parts of Zenoss with automatic filters applied; such as by device, by event type, by device type, by System
- Events are colour-coded by severity:

Critical	Red	5
Error	Orange	4
Warning	Yellow	3
Info	Blue	2
Debug	Grey	1
Clear	Green	0

- Can also configure event consoles for specific users limiting what is seen, by device, device class, event, Location, System, Group, ...

www.skills-1st.co.uk

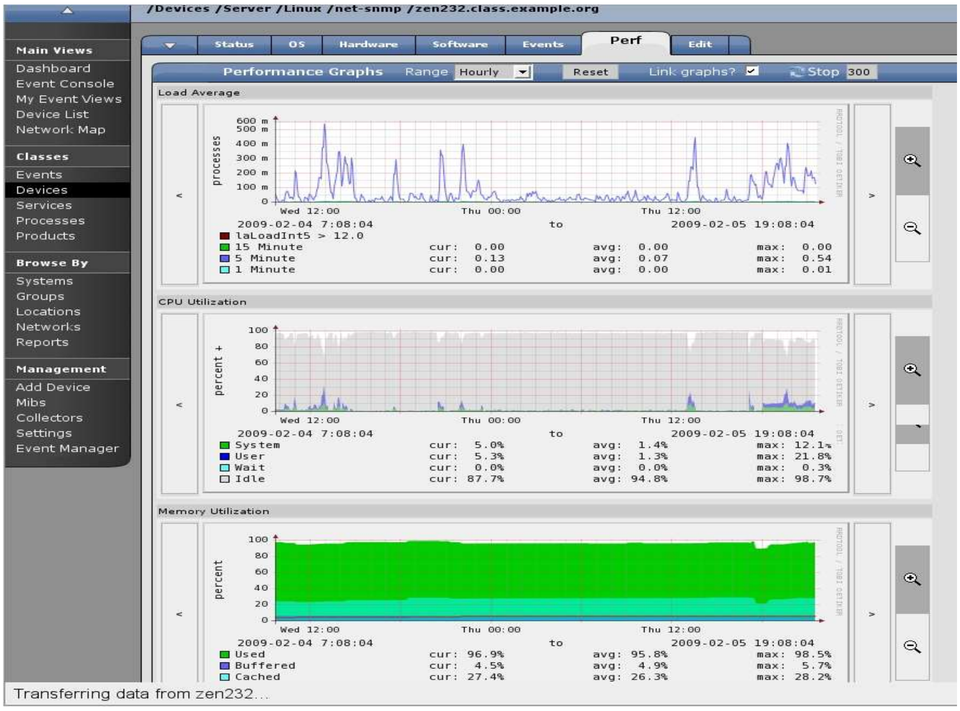
Zenoss Network Map



- Uses Flash plugin
- Limited to 4 hops from a device
- Layout algorithm avoids “crossed lines”
- Repulsion factor a little twitchy
- Copes with meshed networks providing the underlying topology has been constructed correctly

www.skills-1st.co.uk

Zenoss Performance & Reports



/Devices /Server /Linux /net-snmp /zen232.class.example.org

Load Average

```
2009-02-04 7:08:04                    to            2009-02-05 19:08:04
■ laLoadInt5 > 12.0
■ 15 Minute               cur:  0.00          avg:  0.00          max:  0.00
■ 5 Minute                cur:  0.13          avg:  0.07          max:  0.54
□ 1 Minute                cur:  0.00          avg:  0.00          max:  0.01
```

CPU Utilization

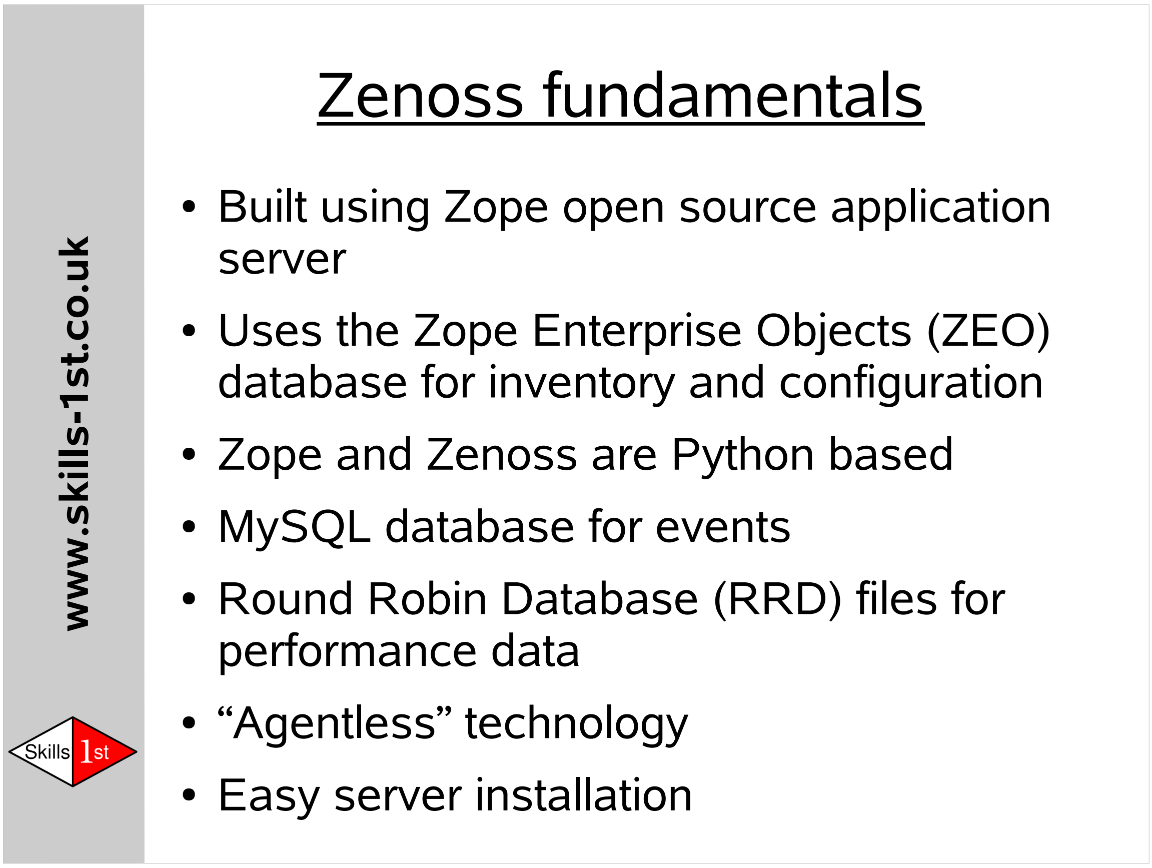
```
2009-02-04 7:08:04                    to            2009-02-05 19:08:04
■ System                  cur:  5.0%          avg:  1.4%          max: 12.1%
■ User                    cur:  5.3%          avg:  1.3%          max: 21.8%
□ Wait                    cur:  0.0%          avg:  0.0%          max:  0.3%
□ Idle                    cur: 87.7%          avg: 94.8%          max: 98.7%
```

Memory Utilization

```
2009-02-04 7:08:04                    to            2009-02-05 19:08:04
■ Used                    cur: 96.9%          avg: 95.8%          max: 98.5%
■ Buffered                cur:  4.5%          avg:  4.9%          max:  5.7%
□ Cached                  cur: 27.4%          avg: 26.3%          max: 28.2%
```

Transferring data from zen232...

- Lots of data collection available out-of-the-box (OOTB)
- Data collection controlled by *templates* that are assigned based on the device class
  (eg. /Server/Solaris, /Server/Windows, /Network/Switch)
- Only interface data collection active, OOTB
- Basic data collection applied to /Device/Server classes of devices based on SNMP MIB-2 and Host Resources MIB
  - CPU utilisation
  - Memory utilisation
  - Filesystem utilisation
  - Interface traffic
- Data collection *data sources, thresholds* and *graphs* are all provided and can all be modified, either for individual devices or for classes of devices
- Windows server templates require SNMP Informant MIB and subagent installed on targets for memory, disk and CPU utilisation data
- Separate Reports menu for reporting across all devices:
  - Availability
  - CPU / Memory / Filesystem / Interface utilisation
  - Thresholds breached
  - Inventory
- Lots of scope to change existing data collection and graphs and to add new ones

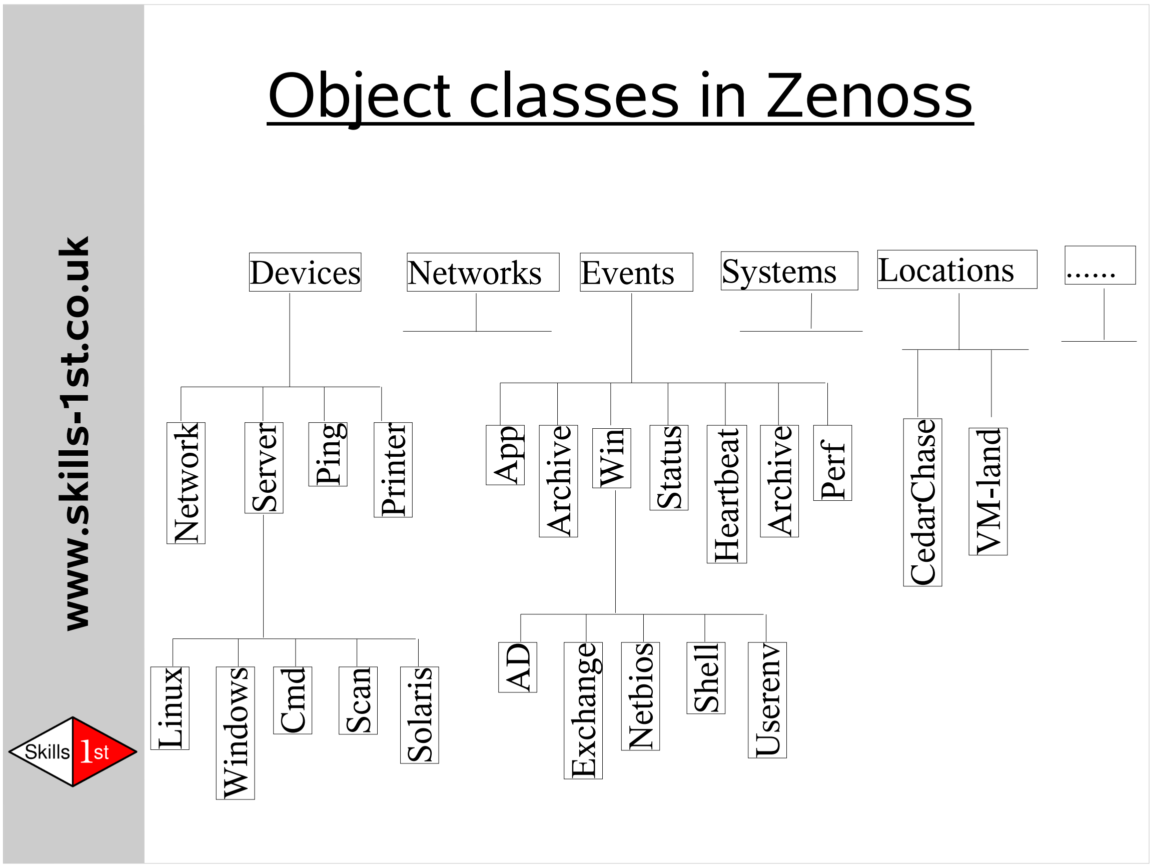


www.skills-1st.co.uk

Zenoss fundamentals

- Built using Zope open source application server
- Uses the Zope Enterprise Objects (ZEO) database for inventory and configuration
- Zope and Zenoss are Python based
- MySQL database for events
- Round Robin Database (RRD) files for performance data
- “Agentless” technology
- Easy server installation

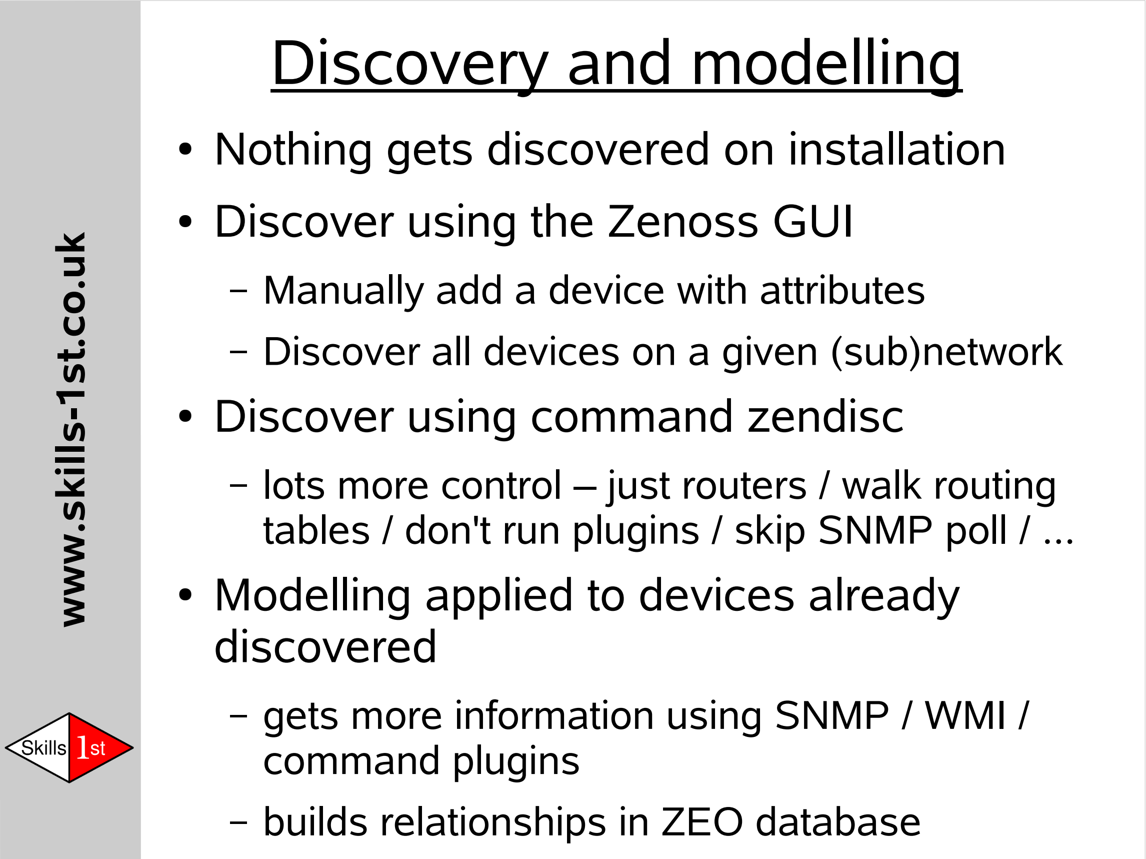
- Download code from http://www.zenoss.com/download/links?nt
  - Stack installers include all dependencies – simply run the executable. Available for:
    - RedHat – commercial and Open Source
    - SuSE – commercial and Open Source
    - Debian
    - Ubuntu
    - Mac OS X
  - Native packages for Red Hat 4 / 5 and CentOS 5
  - Source tarballs for FreeBSD. Gentoo, Solaris 10, Generic Linux
  - VMware appliances for Linux and Windows
- Base protocol is Simple Network Management Protocol (SNMP) -supports v1, v2c and v3
- Also has support for ssh. telnet, Nagios plugins
- All configuration data held in Zope's ZEO database
- Command line access into the Zenoss Python environment using *zendmd*
- Performance data held in Round Robin Database (RRD) archives
- MySQL installed and configured for events database



Object classes in Zenoss

www.skills-1st.co.uk

- All configuration data is object-oriented, arranged in *classes*
- Individual devices can have relationships with several different object classes
  - Device class
  - Location
  - Systems
  - Groups
  - Operating System
  - Hardware product
  - ....
- Each object (device, event, ...) has zProperties.
- Device zProperties (found from the drop-down menu -> More option) control how availability monitoring is performed (SNMP parameters, ssh parameters, Windows login parameters); what should be modelled (interfaces to exclude, number of routing table entries to collect) and what performance data to collect.
- zProperties are inherited down the Device object hierarchy so common parameters specified at the /Device level are inherited by everything; /Device/Server zProperties may change some properties and add others; /Device/Server/Windows will refine the properties further, and so on. zProperties can be changed for specific device instances, if required.

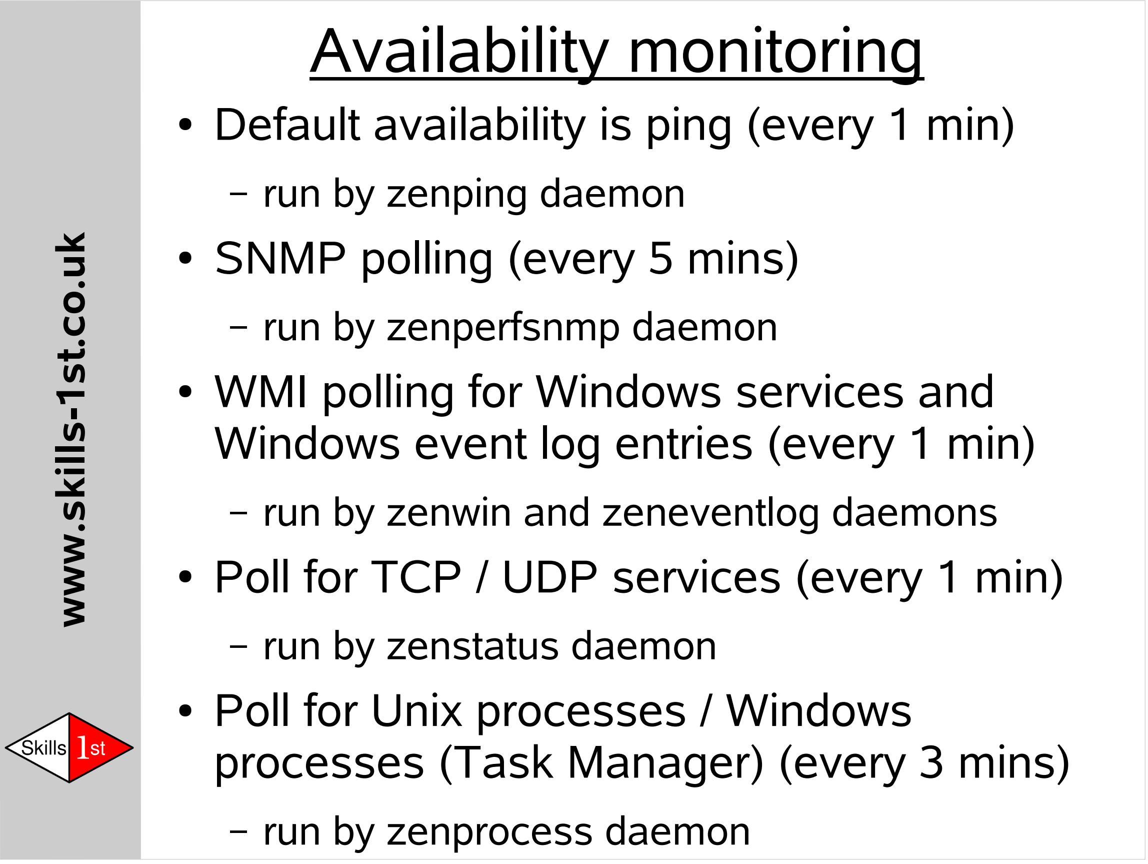


www.skills-1st.co.uk

Discovery and modelling

- Nothing gets discovered on installation
- Discover using the Zenoss GUI
  - Manually add a device with attributes
  - Discover all devices on a given (sub)network
- Discover using command zendisc
  - lots more control – just routers / walk routing tables / don't run plugins / skip SNMP poll / ...
- Modelling applied to devices already discovered
  - gets more information using SNMP / WMI / command plugins
  - builds relationships in ZEO database

- Discovery can be run through the Zenoss GUI, using the zendisc command or can use data files to batch-import devices
- GUI discovery allows many attributes and relations to be specified:
  - Device class
  - SNMP community / port
  - Tag, Rack Slot, Serial Number, comment
  - Hardware manufacturer / product, OS manufacturer / product
  - Location / System(s) / Group(s)
- Discovery of networks more problematical
  - No way to specify exclusion ranges for networks
  - Using zendisc, you can specify to only find routers
  - Devices go into the / Discovered device class on network discovery
- Discovery process builds a network topology by getting routing tables using SNMP – if this information is not available or is incorrect, then the Zenoss topology will not be accurate or complete
- If the topology *is* good then Zenoss will suppress events beyond known points-of-failure thus giving root-cause analysis
- Modelling is run by the zenmodeler daemon – a complete re-model is performed every 12 hours, by default
- Lots of modeller collector plugins available out of the box based on SNMP, WMI, commands – can add to these. Active collector plugins automatically assigned based on device class.
- Large collection of hardware / software manufacturers and products already populate the Products object hierarchy
- Locations, Systems and Groups can be used to reflect local environment

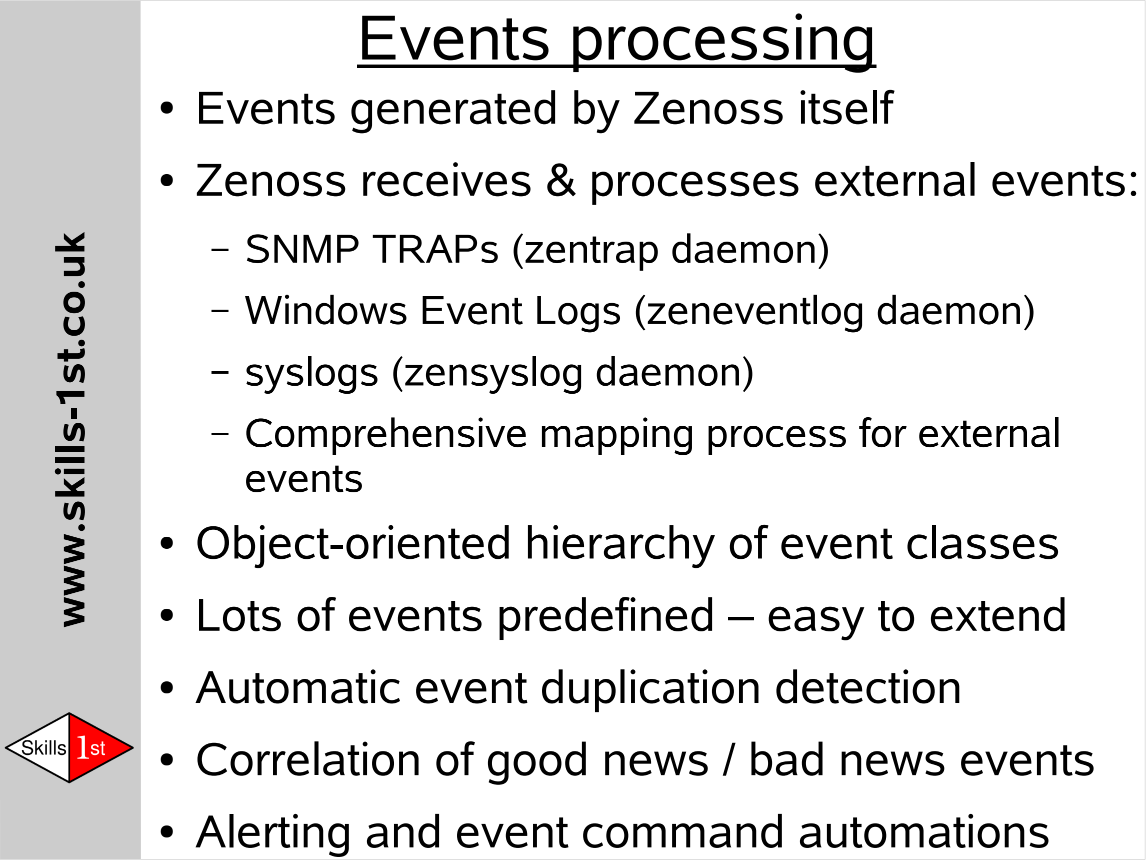


www.skills-1st.co.uk

Availability monitoring

- Default availability is ping (every 1 min)
  - run by zenping daemon
- SNMP polling (every 5 mins)
  - run by zenperfsnmp daemon
- WMI polling for Windows services and Windows event log entries (every 1 min)
  - run by zenwin and zeneventlog daemons
- Poll for TCP / UDP services (every 1 min)
  - run by zenstatus daemon
- Poll for Unix processes / Windows processes (Task Manager) (every 3 mins)
  - run by zenprocess daemon

- Ping-polling is the default but can be disabled for devices behind an ICMP-blocking firewall
- If ping test active and it fails, then all other availability monitoring is suspended until ping test is successful
- zenprocess daemon uses SNMP to poll for process information using the Host Resources MIB. If the SNMP agent doesn't respond then process availability monitoring is suspended until SNMP agent back up
- Huge number of Windows services configured out-of-the-box (but none active)
- Huge number of TCP / UDP services configured out-of-the-box (but none active)
- TCP / UDP service monitoring, process monitoring and windows service monitoring can be enabled for specific devices or device classes
- With Zenoss Core, it is non-trivial to configure more than one collector so having different polling intervals for different collections of systems is not easy – Zenoss Enterprise offers this



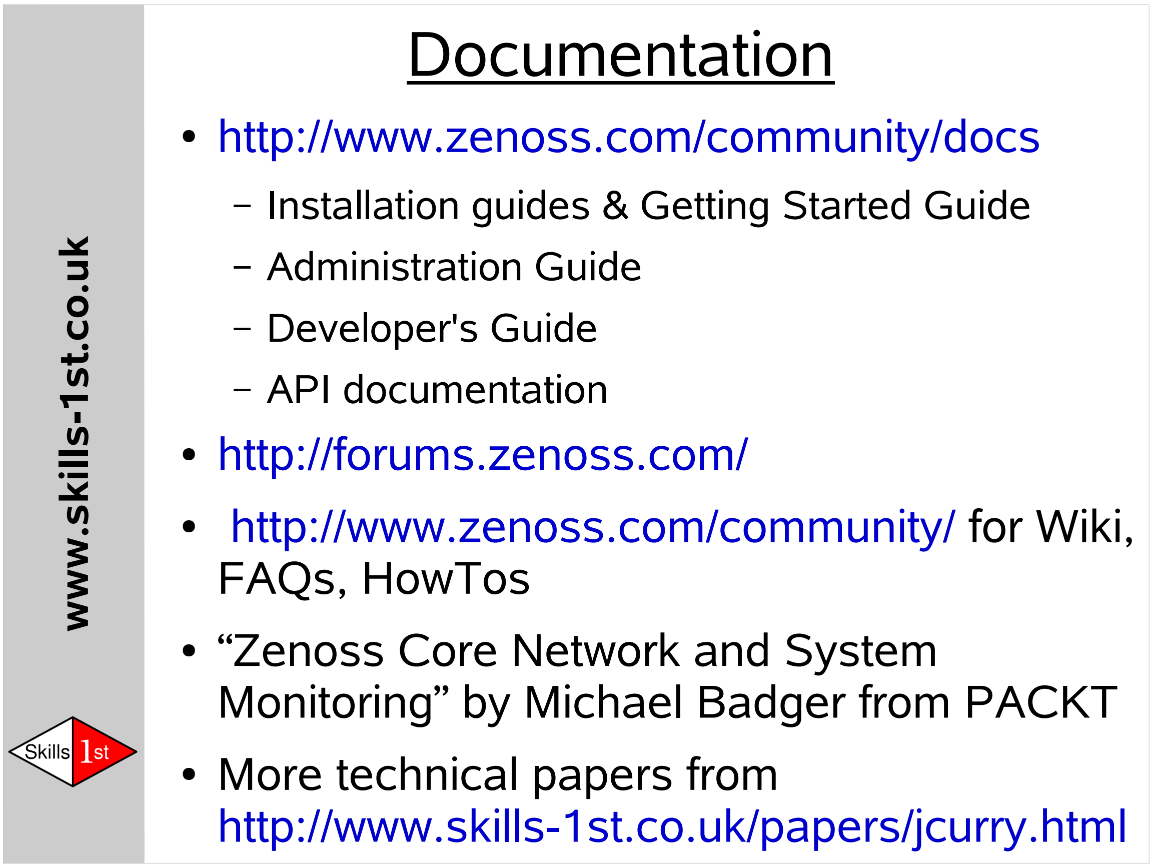
Events processing

- Events generated by Zenoss itself
- Zenoss receives & processes external events:
  - SNMP TRAPs (zentrap daemon)
  - Windows Event Logs (zeneventlog daemon)
  - syslogs (zensyslog daemon)
  - Comprehensive mapping process for external events
- Object-oriented hierarchy of event classes
- Lots of events predefined – easy to extend
- Automatic event duplication detection
- Correlation of good news / bad news events
- Alerting and event command automations

www.skills-1st.co.uk

Skills 1st

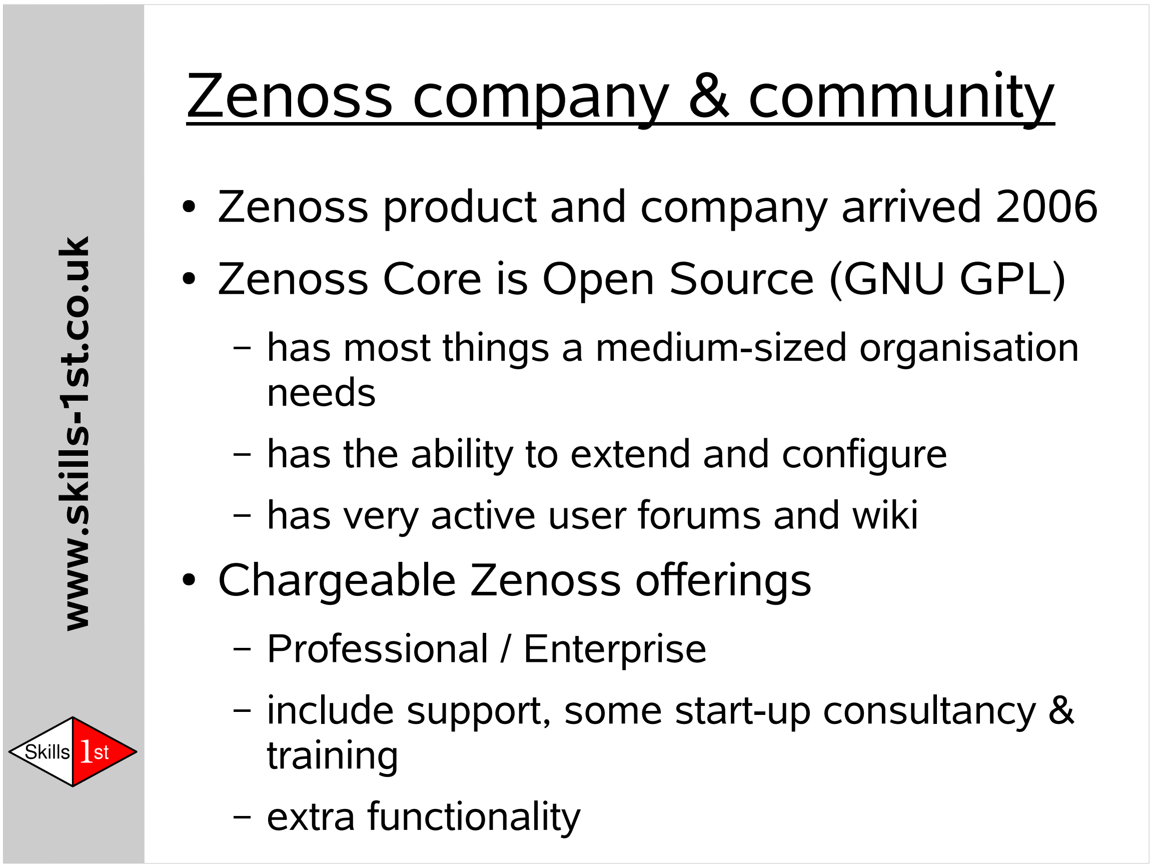
- Events generated external to Zenoss are parsed to interpret the native event into some of the fields of a Zenoss event
- They then go through an event mapping process to assign an event class is to the incoming event - the event class hierarchy allows more specific actions to be taken for particular events
- An event class mapping can consist of:
  - A Rule section (Python statements) to test fields of both the incoming event and attributes of the device that sent the event
  - A Regex section that uses Python regular expressions to parse the *summary* field of the event, optionally creating user-defined fields for the event
  - A transform section, written in Python, which can manipulate many attributes of both the incoming event and the device that sent it
- An event class has zProperties to define zEventSeverity, zEventAction - what to do with the event (leave in the status table of the events database, clear to the history table or drop entirely), and zEventClearClasses – other events that a “good news” event will clear in addition to “bad news” events similar to itself
- Rules and transforms can also make use of event “device context” - the production state, device class,, location, group(s) and system(s) that the device sending the event belongs to.
- Alerting rules can be configured on a per-user basis – typically email or paging. They can include a wide variety of filters and each user can have different schedules for when their alerting rules are active.
- Event commands can run any shellscript to automate responses to particular events.



Documentation

- http://www.zenoss.com/community/docs
  - Installation guides & Getting Started Guide
  - Administration Guide
  - Developer's Guide
  - API documentation
- http://forums.zenoss.com/
- http://www.zenoss.com/community/ for Wiki, FAQs, HowTos
- “Zenoss Core Network and System Monitoring” by Michael Badger from PACKT
- More technical papers from http://www.skills-1st.co.uk/papers/jcurry.html

- Original “Open Source Management Options” paper published September 2008 compared Nagios, OpenNMS and Zenoss with a quick look at MRTG, Cacti and the Dude - http://www.skills-1st.co.uk/papers/jane/open_source_mgmt_options.pdf
- Conclusion was that OpenNMS and Zenoss are both full-function, open source, network and systems management offerings but Zenoss was slightly ahead (partly because Zenoss is *not* written in Java and OpenNMS is – personal preference!)
- Negative points on Zenoss were code stability and quality of documentation.
- By February 2009, code stability feels much better; 2.3 versions of the documentation is improving but still needs more work
- Zenoss community website is hard to navigate to find technical help
- “Getting Started” guide is good for just that!
- Zenoss Administration Guide and book by Michael Badger are good to get basic administration techniques
- No good documentation on more detailed, technical aspects
- Detailed papers written by Jane Curry - http://www.skills-1st.co.uk/papers/jcurry.html :
  - “Zenoss Event Management” - 80 page paper with architecture diagrams. screenshots, lots of transform examples
  - “Crafting Zenoss Core users for events and zProperties”
  - “Zenoss Discovery and Classification”
- 2-day Zenoss Administration course available from Zenoss (currently US only)
- 3-day Zenoss Event Management workshop from Skills 1st Ltd



www.skills-1st.co.uk

Zenoss company & community

- Zenoss product and company arrived 2006
- Zenoss Core is Open Source (GNU GPL)
  - has most things a medium-sized organisation needs
  - has the ability to extend and configure
  - has very active user forums and wiki
- Chargeable Zenoss offerings
  - Professional / Enterprise
  - include support, some start-up consultancy & training
  - extra functionality

- Find Zenoss at http://www.zenoss.com
- Zenoss based on an earlier project by Erik Dahl, one of the Zenoss co-founders
- Zenoss Core licensed under GNU General Public License
- Statistics from Zenoss for the end of 2008:
  - 110 new Enterprise customers in 2008
  - Revenue for 2008 up 382% on 2007
  - Passed 875,000 total downloads; over 350,000 in 2008
  - Consistently rated in top ten most active projects on SourceForge
- Download Zenoss Core from http://www.zenoss.com/download/links?nt
- Zenoss community website at http://www.zenoss.com/community/
- Zenoss Professional – new in 2008
  - Starting point is minimum 100 monitored devices at $100 / device
  - Extended monitoring capability
  - Role based access control
  - Silver and gold service plans
  - 2 hours deployment planning consultancy
  - 1 free seat on Zenoss Administration course
- Zenoss Enterprise
  - Starting point is 250 monitored devices at $150 / device
  - Even more monitors, global dashboard
  - Distributed, high-availability architecture
  - Gold and platinum service plans
  - 3 hours deployment planning consultancy & 2 seats on course

Comparison of 3 offerings at
http://www.zenoss.com/product/network-monitoring-software#subscription

Conclusions

- Zenoss is a good choice for open source network and systems management
- “Core” offering is free; Professional & Enterprise chargeable offerings with support and greater scalability
- Code stability improved a lot in 6 months
- Still not much exposure outside US, but growing
- Huge community behind it