# Release Notes for Open-AudIT v1.10

## Executive Summary

New feature - Baselines. Database changes which will affect groups and reports. Upgrading will remove noncurrent database rows. IBM DSxxxx SAN auditing.

Open-AudIT Enterprise - New - Baselines.

Open-AudIT - Bug - Fixed software license editing by a non-Admin user.

Open-AudIT - New - Audit IBM DS 5xxx SANs.

Open-AudIT - Change - New SQL schema naming convention and associated changes.

Open-AudIT - Change - Audit Log renamed to Edit Log. Audits renamed to Audit Log. Alert Log renamed to Change Log.

Open-AudIT - New - Added SNMP helpers.

Open-AudIT - New - Group definition for NAS device types.

Open-AudIT - New - Added a function (not exposed in the GUI) to reset all devices ip addresses in a given group.

Open-AudIT - Bug - Fixed audit script for disks in El Capitan.

Open-AudIT - Improved - Improved matching code when processing vm guests.

Open-AudIT - Depreciated - Depreciated the audit_subnet.[vbs|sh] scripts as Discovery performs these functions, only better.

Open-AudIT - Changed - Report and Group definitions removed from SQL script. When a user accesses any Open-AudIT page a check is performed and if there are no reports, the default reports are created from the report definition files. Same for Groups.

## Baselines

Our new major feature for 1.10 is the beginning of our Baselines feature. This is not finished as yet (in 1.10), but we wanted it out there for feedback. Baselines in Open-AudIT Enterprise allow you to take the details of one machine (say software) and use that as a basis for comparison against another machine or group of machines. So you can say "Take the software installed on Machine X and tell me where it's different on the device group for Web Servers." You will get a nice GUI interface showing which machines did or did not meet the expected software install state. You can also apply this to users and netstat ports. Other tables will be introduced in the future.

How to use the Baseline feature is detailed here.

*WARNING - When creating a baseline using software policies, at present Centos and RedHat package the kernel using the names 'kernel' and 'kernel-devel'. There can be multiple packages with this name and different versions concurrently installed. Debian based distributions use names like 'linux-image-3.13.0-24-generic', note the version number is included in the package name. Because RedHat based OS's use this format and subsequently have multiple identical package names with different versions we currently exclude 'kernel' and 'kernel-devel' from software policies. This may be addressed in a future update.*

## Database Changes

1.10 also marks the start of our move to a new database structure. Initially I have modified the sub-system tables (sys_hw_bios, sys_sw_windows, etc) to make them consistent in terms of column names and types as well as move away from the "system.system_id = sys_sw_software.system_id AND system. timestamp = sys_sw_software.timestamp" joins.

From 1.10 most sub-system tables have been renamed. sys_sw_software becomes software. sys_hw_bios becomes bios, etc. Column names within these tables have also changed. software_version becomes version. software_name becomes name, etc.

The columns of first_timestamp and timestamp have been renamed to first_seen and last_seen.

Tables will now look similar to the below example.

BIOS goes from

```
CREATE TABLE `sys_hw_bios` (
`bios_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`system_id` int(10) unsigned DEFAULT NULL,
`bios_description` varchar(200) NOT NULL DEFAULT '',
`bios_manufacturer` varchar(200) NOT NULL DEFAULT '',
`bios_serial` varchar(100) NOT NULL DEFAULT '',
`bios_smversion` varchar(100) NOT NULL DEFAULT '',
`bios_version` varchar(100) NOT NULL DEFAULT '',
`bios_asset_tag` varchar(100) NOT NULL DEFAULT '',
`timestamp` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`first_timestamp` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
PRIMARY KEY (`bios_id`),
KEY `system_id` (`system_id`),
CONSTRAINT `sys_hw_bios_system_id` FOREIGN KEY (`system_id`) REFERENCES `system` (`system_id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

to

```
CREATE TABLE `bios` (
`id` int(10) unsigned NOT NULL AUTO_INCREMENT,
`system_id` int(10) unsigned DEFAULT NULL,
`current` enum('y','n') NOT NULL DEFAULT 'y',
`last_seen` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`first_seen` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
`manufacturer` varchar(200) NOT NULL DEFAULT '',
`serial` varchar(100) NOT NULL DEFAULT '',
`description` varchar(200) NOT NULL DEFAULT '',
`smversion` varchar(100) NOT NULL DEFAULT '',
`version` varchar(100) NOT NULL DEFAULT '',
`asset_tag` varchar(100) NOT NULL DEFAULT '',
PRIMARY KEY (`id`),
KEY `system_id` (`system_id`),
CONSTRAINT `sys_hw_bios_system_id` FOREIGN KEY (`system_id`) REFERENCES `system` (`system_id`) ON DELETE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

And for those wondering, yes, I also plan to rename the constraints (foreign keys) in a later version to also come into line with the new naming convention.

As well as the naming convention change, the join (as mentioned) now removes the 'timestamp' requirement. I have introduced a new column called 'current' which will always be 'y' or 'n'. The logic in code now determines whether an attribute reported is in the database for the device or not and inserts a new row or updates an existing row. Alerts still work as before - only now are more reliable. This change was necessary by the ever increasing sources of data we process. A good example - if we see a Linux device and audit it using the audit script, we get lots of nice information. If we then see the same device and retrieve information using SNMP, we don't get any items back for software (for example). When we process the SNMP result we have to (manually, in code) realise that "oh, this is an SNMP result - we don't have software. For all the current rows in the software table, update their timestamps so they're still considered current". Taken as a one off, this isn't a big deal. But it's becoming more and more unwieldy to keep track of which data sources provide which data types and update or not for every combination.

So from 1.10 onwards we have a simple 'current' column which is used to determine if an item is currently associated with a device. Joins go from "system.system_id = sys_sw_software.system_id AND system.timestamp = sys_sw_software.timestamp" to "software.system_id = system.system_id and software.current = 'y'". This also has the side affect of enabling us to select attributes from a table without an actual table join for a specific device like this "SELECT * from software where system_id = ? and current = 'y'". No table join required :-)

The only sub-system table not modified for 1.10 is the sys_hw_network_card_ip table. I do have code for this, but am holding this back for a future release. As you can imagine, it's a rather large change that affects a lot of groups and code.

In conjunction with the database naming convention is data output. When data is retrieved from MySQL using PHP, all data is returned as strings. As part of the new naming convention, certain columns with specified names will be converted into their correct type. If the column name is 'id' or 'free' or 'used' or 'size' or 'speed' or 'total' it will be converted to an integer data type. If the column name ends with '_id' or '_count' or '_percent' or '_size' it will be converted to an integer. If the column name is 'ip' or 'next_hop' or 'destination' it will be a left as a string and de-padded to a regular ip address and another column called 'ip_padded' or 'next_hop_padded' or 'destination_padded' will also be returned containing the padded ip address.

I have not altered the system table.

As part of the 1.10 upgrade process, all non-current attributes are deleted from tables. This will have a beneficial effect of speeding up your database as well :-)

Also part of the 1.10 upgrade process is refreshing all groups and reports with new definitions to match the new naming convention. Any custom groups or reports you have made will not be altered so you will need to export these, modify them to suit the new naming convention and re-import them. Apologies for any inconvenience this causes, but there is no way around that particular item :-(

Moving forward, other tables in the database will also adopt the new naming convention - oa_groups, oa_location, oa_organizations, oa_users, sys_hw_network_card_ip, system etc.

# Moving Forward

Post 1.10, the Baseline feature will be the next completed task.

After that the sys_hw_network_card_ip table will be bought into line with the other tables.

Looking further ahead, also on the roadmap is changing the system table. Currently we store two attributes for anything which the user can change from the web interface (description and man_desciption for example). This was done so that a user could set the value they want (man_description) and an audit result would not over write it (description). When a device is first audited both values are set, but on subsequent audits, only the description value is updated. This system will be replaced. A single column will be kept (say 'description') instead of two. Any process that can change this value (a user, an audit, an snmp result, etc) is given a weight. When the column has new data, the change sources weight is compared to the current weight. If the weight is greater, the column value is allowed to be changed. This will have the benefit that only a single column will be required and an even more extensive edit log will be kept. I have code for this, but again, 1.10 has enough changes for now. I'll be doing some more testing and tweaking - but that is the plan.

And finally, another planned change will be linking an Open-AudIT user with an organisation. This will essentially allow multi-tenant Open-AudIT installations to work much better and have more fine grained controls. An Open-AudIT user will have permission on one or more Organizations. When Open-AudIT shows devices, only those devices belonging to an Organization (or an Organization's children) that the user has group rights on will appear to that user. I have code that does this now and it works, but as said this is still in the early planning stages. If anyone would like input into this, please email me while it's still in development.