Digital Payments
for a Trusted World

boom 5.10.0
User and Administration Manual

Version 5.10.0, 2024-02-29

Table of Content

1. Introduction . . . 1
2. Installation . . . 2
  2.1. Installation Prerequisites . . . 2
    2.1.1. Resolving Hostnames . . . 2
    2.1.2. Server Requirements . . . 2
    2.1.3. Agent Requirements . . . 3
    2.1.4. User Interface Requirements . . . 4
    2.1.5. Platform Notes . . . 5
      2.1.5.1. Mac OS X . . . 5
      2.1.5.2. Linux . . . 6
  2.2. Installation Steps . . . 6
    2.2.1. Server Installation . . . 6
      2.2.1.1. MySQL: . . . 6
        2.2.1.1.1. For MySQL Version 8 . . . 7
        2.2.1.1.2. For MySQL before Version 8 . . . 7
      2.2.1.2. Oracle . . . 8
      2.2.1.3. Installing on Linux/Unix Systems . . . 8
      2.2.1.4. Installing on Microsoft Windows systems . . . 12
    2.2.2. Server running as "non-root" . . . 13
    2.2.3. Agent Installation . . . 14
      2.2.3.1. Installing on Linux/Unix Systems . . . 15
      2.2.3.2. Silent Installation on Linux/Unix Systems . . . 18
      2.2.3.3. Agent running as "non-root" . . . 18
      2.2.3.4. Installing on Windows systems . . . 20
      2.2.3.5. Postinstallation Tasks . . . 21
    2.2.4. Agent Uninstallation . . . 22
      2.2.4.1. Uninstalling the Agent on Unix Systems . . . 22
      2.2.4.2. Uninstalling the Agent on Microsoft Windows Systems . . . 22
    2.2.5. *boom* Client (User Interface) installation . . . 22
  2.3. Licensing . . . 23
  2.4. *boom* Upgrade . . . 25
    2.4.1. *boom* Server Upgrade . . . 26
      2.4.1.1. *boom* Server Upgrade Example Unix/Linux . . . 27
      2.4.1.2. *boom* Server Upgrade Example Windows . . . 31
    2.4.2. *boom* Client / UI Upgrade . . . 34
    2.4.3. *boom* Agent Upgrade . . . 37
  2.5. Firewalled Environment . . . 37
  2.6. TLS Encryption . . . 39
    2.6.1. Enabling TLS on the Agent . . . 40
    2.6.2. Boom PKI . . . 40
    2.6.3. File Based PKI . . . 41
      2.6.3.1. Setting up a *boom* Server to run with File Based PKI . . . 41
      2.6.3.2. Switching a Server from Boom PKI to File Based PKI . . . 42
  2.7. Example: Running *boom* as part of a Pacemaker Cluster . . . 43

2.7.1. Requirements . . . 43
2.7.2. Folder structure . . . 43
2.7.3. MySQL Installation and Configuration . . . 43
2.7.4. *boom* Installation . . . 45
2.7.5. Cluster Installation . . . 45
3. Architecture . . . 47
3.1. Architecture Overview . . . 47
3.2. *boom* Primary / Backup Concept (Single-Tier) . . . 48
3.2.1. *boom* Primary-Backup Server Scenario . . . 48
3.2.2. Synchronization Primary / Backup . . . 49
3.3. *boom* Master / Slave Concept (Multi-Tier) . . . 50
3.3.1. *boom* Master-Slave Server Scenarios . . . 51
3.3.2. Synchronization Master-Proxy server . . . 53
4. Concept . . . 54
4.1. Concept Overview . . . 54
4.1.1. *boom* Agent . . . 55
4.1.2. *boom* Server . . . 55
4.2. Communication matrix . . . 56
4.3. Indications . . . 56
4.3.1. Indication Processing on the *boom* Server . . . 57
5. Monitoring . . . 59
5.1. Monitor Policy (Threshold monitoring) . . . 59
5.1.1. General Information . . . 59
5.1.2. Output Parser Monitor (OPM) . . . 64
5.1.3. SNMP Monitor . . . 73
5.1.4. Calculation Syntax for OPM and SNMP Variables . . . 75
5.1.5. Java Monitor . . . 78
5.1.6. Exec Monitor . . . 80
5.1.7. External Monitor . . . 82
5.2. Indication Policy (Event triggered/Text related monitoring) . . . 82
5.2.1. General Information . . . 82
5.2.2. boomindi . . . 85
5.2.3. Hybrid Policy . . . 86
5.2.3.1. Logfile Monitors . . . 86
5.2.3.1.1. Logfile Monitor . . . 86
5.2.3.1.2. Logfile Transaction Monitor . . . 89
5.2.3.1.3. Logfile Transaction Monitor (multiline) . . . 96
5.2.3.2. MultiPath (MP) Logfile Monitor . . . 99
5.2.3.3. NAGIN . . . 104
5.2.4. SNMP Traps . . . 108
5.3. Patterns . . . 109
5.3.1. Simplified Patterns . . . 109
5.3.2. Java Patterns . . . 111
5.3.3. Supported Variables and Functions . . . 112
5.3.3.1. Global Variables . . . 113
5.3.3.2. Supported Variables . . . 113
5.3.3.3. Supported pre-process Functions . . . 114

5.3.4. Hiding Cleartext Passwords . . . 116
5.4. Assignments . . . 118
5.5. Binary Packages . . . 119
5.6. Performance Data . . . 120
5.6.1. Overview . . . 120
5.6.2. Agent command based performance data collection . . . 122
5.6.3. Ignore Performance Classes on server . . . 123
5.6.4. Cleaning data. . . . 123
5.7. Actions . . . 123
5.8. Submitting Monitor Values / Indications . . . 124
5.8.1. Submitting Monitor Values . . . 125
5.8.2. Submitting Indications . . . 125
5.8.3. Submitting Performance Data . . . 126
5.8.4. Local Interprocess Communication. . . . 127
5.9. Indication Reduction in *boom* . . . 127
5.9.1. Duplicate Indication Suppression . . . 127
5.9.2. Silence Count for Monitor Policies. . . . 128
5.9.3. Indication Correlation via Indication Key and Close Mask . . . 128
5.10. Heartbeat Polling. . . . 129
5.10.1. Server Heartbeats . . . 129
5.10.2. Agent Heartbeats . . . 130
5.11. Agent Management . . . 131
5.11.1. Add and Approve Agent . . . 131
5.11.2. Delete Agent . . . 133
5.11.3. Virtual Agents . . . 135
5.12. Auditing . . . 139
5.13. Indication Enrichment/Modification . . . 139
5.14. Maintenance Window Management . . . 139
6. *boom* User Interface (Client). . . . 142
6.1. Overview and Basic Operations. . . . 143
6.1.1. General Information . . . 143
6.1.2. Basic Workbench Operations . . . 145
6.1.3. Menu Bar . . . 147
6.1.4. Tool Bar . . . 150
6.1.5. *boom* UI update. . . . 152
6.1.6. Change UI Password . . . 152
6.1.7. Admin Broadcasts & User Messaging . . . 153
6.1.8. *boom* Client Version. . . . 154
6.1.9. Reset UI Layout. . . . 155
6.1.10. General UI Settings . . . 155
6.1.10.1. General Attributes . . . 156
6.1.10.2. Indication Browser . . . 156
6.1.10.3. Predefined Tabs . . . 158
6.1.10.4. Predefined Views . . . 159
6.1.10.5. Web Browser . . . 159
6.1.10.6. Web Browser Tabs . . . 160
6.2. Access and Login . . . 160

6.2.1. Login to the *boom* Workbench . . . 160
6.2.2. Login Problems . . . 162
6.3. Hosts / Agents . . . 163
6.3.1. General Information . . . 163
6.3.2. Agent Details and Agent Operations . . . 164
6.3.3. Agent List . . . 167
6.3.4. Agent Overview . . . 170
6.3.5. Deployed Assignments and single deployed Policies . . . 171
6.3.6. Agent Auto-Inventory . . . 173
6.4. Indications . . . 174
6.4.1. General Information . . . 174
6.4.2. Indication View Layout . . . 175
6.4.3. Indication Operations . . . 176
6.4.3.1. Active Indication . . . 176
6.4.3.2. Closed Indication . . . 177
6.4.4. Status Line . . . 179
6.4.5. Indication Details . . . 179
6.4.6. Indication Filtering . . . 182
6.4.7. Indication Similarity View . . . 183
6.4.8. Archived Indications . . . 184
6.4.9. Indication Export . . . 185
6.4.9.1. Indication Export (Text) . . . 186
6.4.9.2. Indication Export (CSV) . . . 187
6.5. Policy Management . . . 188
6.5.1. General Information . . . 188
6.5.2. Policy Operations . . . 189
6.5.3. Copy/Rename Policies . . . 194
6.5.4. Compare Policies . . . 196
6.5.5. Searching Policies . . . 196
6.5.6. Monitor Policy . . . 199
6.5.6.1. Monitor Policy Details . . . 199
6.5.6.2. Monitor Policy Conditions . . . 203
6.5.6.3. Monitor Types . . . 208
6.5.6.4. Supported variables and Pre-Process functions . . . 208
6.5.6.5. Building a Monitor Policy . . . 208
6.5.7. Indication Policy . . . 211
6.5.7.1. Indication and Hybrid Indication Policy . . . 211
6.5.7.2. Indication Policy Details . . . 213
6.5.7.3. Indication Policy Conditions . . . 214
6.5.7.4. Nagin Trigger and LogfileMonitor Types . . . 219
6.5.7.5. Indication Policy BOOM_Messages . . . 220
6.5.7.6. Indication Policy for SNMP traps . . . 221
6.5.7.7. Indication Policy for SNMP gets . . . 223
6.5.7.8. Indication Policy for Logfile Monitors . . . 224
6.5.7.9. Indication Policy for Multipath Logfile Monitors . . . 225
6.5.7.10. Policy Specific Logfiles . . . 230
6.5.7.11. Supported variables and Pre-process functions . . . 231

6.5.7.12. Policy All Conditions Testing . . . 232
6.5.8. Indication Attribute Sizes. . . . 240
6.6. Packages. . . . 240
6.6.1. General Information . . . 240
6.6.2. Package Operations . . . 241
6.6.3. File Editor. . . . 243
6.6.4. Inventory Package . . . 244
6.7. Assignments . . . 245
6.7.1. General Information . . . 245
6.7.2. Add Policy/Package Links to an Assignment . . . 246
6.7.3. Assignment Operations. . . . 246
6.8. Assignments Summary. . . . 250
6.8.1. General Information . . . 250
6.8.2. Assignment Operations and Filtering . . . 252
6.8.3. Deployment Queue . . . 255
6.9. Server Filters . . . 256
6.9.1. General Information . . . 256
6.9.2. Server Filter Overview . . . 258
6.9.3. Attribute Filters . . . 258
6.9.4. Notification Filters . . . 260
6.9.5. User Filters. . . . 262
6.10. Notifications . . . 263
6.10.1. General Information . . . 263
6.10.2. Notification Overview. . . . 263
6.10.3. Notification Configuration. . . . 265
6.11. Actions . . . 268
6.11.1. General Information . . . 268
6.11.2. Actions Overview. . . . 269
6.11.3. Action Operations . . . 277
6.11.4. Action Details . . . 279
6.11.5. Run an Action . . . 280
6.11.6. Remote Action. . . . 280
6.11.7. Server Action. . . . 281
6.12. Service Dashboard . . . 282
6.12.1. General Information . . . 282
6.12.2. Local/Global Services . . . 283
6.12.3. Add Dashboard Service . . . 284
6.13. Operator Dashboard . . . 287
6.14. SSH Terminal . . . 289
6.15. Server Jobs . . . 290
6.15.1. General Information . . . 290
6.15.2. Server Jobs Operations and Configuration . . . 295
6.16. Import/Export Management Plug-Ins. . . . 296
6.16.1. Import MPI . . . 297
6.16.2. Export MPI. . . . 301
6.17. Statistics . . . 302
6.17.1. General Server Information . . . 302

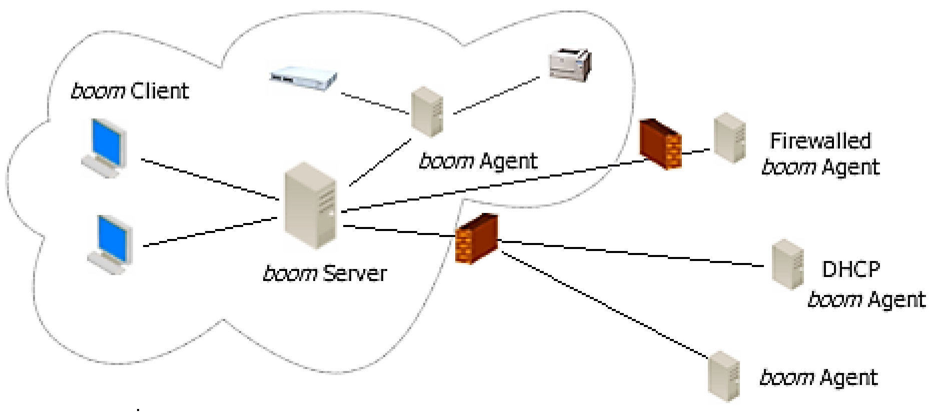
6.17.2. Agent based Statistics 303
6.17.3. Policy based Statistics 304
6.17.4. Performance Data 304
6.17.5. Indication Charts 305
6.18. History Chart 307
6.19. HotSpot 308
6.20. Pattern Validation 309
6.20.1. Pattern Validation Dialog (Java Pattern) 309
6.20.2. Pattern Validation Dialog (Simplified Pattern) 311
6.21. User Management 312
6.21.1. General Information 312
6.21.2. User Roles and Rights 313
6.21.3. User Accounts 314
6.21.4. User Details Dialog 315
6.21.5. Privileged Groups 317
6.21.6. User Filters 319
6.21.7. LDAP Authentication 324
6.21.8. Synchronize Users Job 327
6.22. Configuration View 336
6.22.1. SNMP v3 User Authentication Management 339
6.23. MIB Browser 340
6.23.1. General Information 340
6.23.2. MIB Browser Operations 341
6.24. Service Tree 345
6.24.1. General Information 345
6.24.2. Service Tree 345
6.24.3. Customize Service Tree 347
6.25. Maintenance Window Management 350
6.25.1. General Information 350
6.25.2. Scheduled Maintenance Details and Operations 351
6.25.3. Indication Maintenance 353
6.25.4. Scheduled Maintenance in a Master/Slave environment 354
6.26. AdHoc Maintenance 355
6.26.1. General Information 355
6.26.2. AdHoc Maintenance Details and Operations 355
6.26.3. Create a Manual AdHoc Maintenance 357
6.27. Server Policies 358
6.27.1. General Information 358
6.27.2. Server Policies Operations 360
6.27.3. Usage 362
6.27.4. Supported Variables 364
6.27.5. Server Policies - Listener 364
6.27.6. Server Policies in a Master/Slave environment 366
6.28. Browser 366
7. *boom* Server Mainpage 368
7.1. Web Browser Login 368
7.2. Server Status Page 369

7.3. Dashboards . . . 371
7.4. Deployment Packages . . . 377
8. *boom* Administration . . . 379
8.1. *boom* Server Administration. . . . 379
8.1.1. Starting and Stopping the *boom* Server . . . 379
8.1.2. Checking the *boom* Server Process . . . 379
8.1.3. *boom* Server Configuration Files . . . 380
8.1.4. *boom* Database Configuration Files . . . 383
8.1.5. *boom* Server Logfiles . . . 385
8.1.6. *boom* Agent Administration. . . . 385
8.1.6.1. Starting and Stopping the *boom* Agent . . . 385
8.1.6.2. Checking the *boom* Agent Process . . . 386
8.1.6.3. *boom* Agent Configuration File . . . 386
8.1.6.4. *boom* Agent Logfiles . . . 391
8.1.6.5. *boom* Agent Configuration on Cluster Nodes (Unix/Linux) . . . 391
8.1.6.6. *boom* Agents Connecting to a Clustered *boom* Server Environment . . . 392
8.1.6.7. *boom* Agent Configuration in a SUN Solaris Global Zone . . . 392
8.2. *boom* User Interface Administration . . . 392
8.2.1. User Interface Configuration Files . . . 392
8.2.2. XML Attributes . . . 393
8.2.3. XML File Example . . . 394
8.2.4. Setting Default Login Parameters . . . 394
8.3. Internal Indications / Operational States . . . 396
8.3.1. *boom* Agent Internal Indications . . . 396
8.3.2. *boom* Agent Operational States and Indications. . . . 398
8.4. Message Storm Detection . . . 398
8.5. Defining an Instruction URL . . . 401
8.6. Database Issues . . . 403
8.6.1. Database Insert of Indications . . . 403
8.7. SNMP Trap Listener . . . 403
8.8. Primary-Backup Connection . . . 404
8.8.1. Setting up a Primary/Backup Connection . . . 404
8.9. Master/Slave Connection . . . 406
8.9.1. Setting up a Master/Slave Connection. . . . 406
8.10. Configuring *boom* Auditing . . . 408
8.11. Configuring Outages . . . 408
8.11.1. Configuring Agent based Maintenances. . . . 408
8.11.2. Configuring Server based Maintenances . . . 410
8.12. Using custom certificates for built-in web server . . . 411
9. Command Line Interface. . . . 413
9.1. *boom* Server CLI. . . . 413
9.2. *boom* Agent Utilities . . . 415
10. Copyright & Trademarks . . . 419
11. Contact . . . 422

Chapter 1. Introduction

boom is a distributed client-server monitoring solution that helps IT departments to prevent, detect and solve problems on the managed systems, hosts, applications and services. The powerful filtering and correlation engine allows to increase the efficiency of the IT administrators by reducing the number of incidents and pin pointing of hot spots in the environment.

The agent based architecture provides local intelligence, reliable operation and minimize the network traffic.



Each agent can take the role of a monitoring station that provides agent-less monitoring for remote applications, services and hosts that don't have an agent installed. Firewalled and DHCP environments are handled by the secure communication layer. The role based configuration simplifies the configuration and rollout for complex environments and keeps the administrators in control.

Chapter 2. Installation

All *boom* components are packaged in the server package zip file. The first installation step therefore is always the server installation. The installation of all other components can only be performed after the server is installed and running.

2.1. Installation Prerequisites

Before starting the installation, be sure to verify that

- verify you are using the latest installation package - please contact Worldline's *boom* support (Email: support@bes-worldline.com, Phone: +49 69 256 552 201).
- the Java Runtime is installed on all systems
- the MySQL or Oracle database server is installed and running
- the names or IP addresses of the systems can be resolved by DNS or host files

2.1.1. Resolving Hostnames

Generally the hostname resolution mechanism should be configured to allow the *boom* agent and the *boom* server to resolve their own and each other's hostname to the correct IP address.

In environments with simple network paths and static addresses the hostname resolution is not mandatory. Communication between the server and the agent works without hostname resolution, but under some circumstances this might cause agents to show up with their IP address instead of their name. This can be avoided by setting the label attribute of the agent or using the server's aliasing functionality.

However in case of a complex server/agent setup e.g. restricted routing, multi-homed, address-allocation via DHCP, firewalled etc. hostname resolution on the agent and the server side is required so that the *boom* server and *boom* agents are able to resolve each other's hostnames.

Hostnames can be resolved using different methods e.g. files (`/etc/hosts` or `%windir%\system32\drivers\etc\hosts` for Windows systems), DNS … The hostname resolution should be configured in such a way that hostnames are resolved as fully qualified names.

2.1.2. Server Requirements

The *boom* server requires the **software versions** as listed in the table below:

Software	Version
Java Runtime	Java 17 Check the Java version: `# <java_17_repository_path>/bin/java -version`
MySQL Database	A local or remotely accessible MySQL/MariaDB/PerconaDB Version 5.x Check the mysql installation: i.e. `# /etc/init.d/mysqld status` or `#systemctl status mariadb`
Oracle Database	A local or remotely accessible Oracle Database Version 10g or higher

Software	Version
Operating System	Any that fulfills above requirements Tested are: - CentOS (x86_32, x86_64) - Debian (x86_32, x86_64) - Fedora (x86_32, x86_64) - Microsoft Windows 2016, 2012(R2), 2008(R2) - Novell SuSE Server/OpenSuSE (x86_32, x86_64, ia64) - RedHat Enterprise Linux (x86_32, x86_64, ia64) Deprecated: - HP-UX (PARISC, ia64) - Microsoft Windows 2003(R2), XP, Vista (x86_32, x86_64) - SUN SunOS/Open Solaris (x86_64, SPARC)

Hardware requirements:

	Recommended Minimum
Disc Space - Server	1 GB on the target disc or file system (e.g. `C:\Programs\boom\server` or `/opt/boom/server`) for the server files, policies, actions, and the deployables for agents, monitors and user interface packages.
Disc Space - Database	0.5 GB on the data disc or file system (e.g. `C:\Programs\mysql\data` or `/var/lib/mysql/data`) for the database files. The initial database size will be less than 15MB, but depending on the number of stored indications and performance values it can grow substantially.
Memory - Server	The server is configured by default to operate with a maximum Java heap size of 1,2GB. This setting allows the server to handle around 300000 indications. The value can be adjusted in the startup script.
Memory - Database	By default the *boom* database will use innodb to store the data. The buffer pool size (innodb_buffer_pool_size) should be set to at least 60MB. Logging should be enabled and set according to the recommendations.

2.1.3. Agent Requirements

The *boom* Agent requires the **software versions** as listed in the table below:

Software	Version
Java Runtime	Java 1.8/8 Java 17 Java 21 Check the Java version: `# java -version` Note: The Java minor release 1.8.0.242 contains a known bug and should not be used.

Software	Version
Operating System	Any that fulfills above requirements Tested are: - CentOS (x86_32, x86_64) - Debian (x86_32, x86_64) - Debian ARM (Raspberry PI, ARM) - Fedora (x86_32, x86_64) - FreeBSD (x86_32, x86_64) - HP-UX (PARISC, ia64) - Mac OS/X (x86) - Microsoft Windows 2016, 2012(R2), 2008(R2), 2003(R2), 10, 7, XP, Vista (x86_32, x86_64) - Novell SuSE Server (x86_32, x86_64, ia64) - OpenSuSE (x86_32, x86_64, ia64) - Oracle Linux (x86_64) - arm-raspi - RedHat Enterprise Linux (x86_32, x86_64, ia64) - SUN SunOS/Open Solaris (x86_64, SPARC) - Ubuntu (x86_32, x86_64, ARMv7) - VMWare ESX 3.x - zLinux (SLES, RedHat) - EulerOS
.NET Framework	Only for Windows systems: .NET Framework 2.0 or higher
optional: Standard C++ Library 32bit runtime package	It is recommended to use for 64bit Linux systems the according 64bit Java and Agent package. Additional libraries need to be available if a 32bit JRE and Agent package should be used i.e. for the more conservative memory management on a 64bit Linux system: The 32bit agent command line tools e.g. boomindi, boommon etc. will run only if the Standard C++ Library 32 bit runtime package is installed on the system. Depending on the system the libraries have different names. Examples: - Debian v6 64 bit: lib32stdc++6 - CentOS V6 64 bit: compat-libstdc++-33.i686

Hardware requirements:

	Recommended Minimum
Disc Space	20 MB on the target disc or file system (e.g. `C:\Programs\boom\agent` or `/opt/boom/agent`) for the program files, monitors, actions and temporary data.
Memory	The agent is configured to work with the default maximum Java heap size (plattform and java dependent, but typically between 64MB and 512MB).

2.1.4. User Interface Requirements

The *boom* User Interface requires the **software versions** as listed in the table below:

Software	Version
Java Runtime	Java 1.8/8 Check the Java version: `# java -version` Note: The Java minor release 1.8.0.242 contains a known bug and should not be used.
Operating System	- Linux (x86_32, x86_64) (Tested are RedHat and Novell distributions) - Mac OS/X (x86_64, cocoa) - Microsoft Windows 2012(R2), 2008(R2), 2003(R2), 7, XP, Vista (x86_32, x86_64)

Hardware requirements:

	Recommended Minimum
Disk Space	300 MB on the target disc or file system (e.g. `C:\Programs\boom\boom_gui` or `/opt/boom/boom_gui`) for the program files and profile information.
Memory	The user interface is configured by default to operate with a maximum Java heap size of 512MB.

2.1.5. Platform Notes

2.1.5.1. Mac OS X

Agent

The *boom* agent was tested on Mac OS X Server 10.5, 10.6, 10.8. The installation script of the agent will register the *boom* agent in the 'launchd' configuration. For more information about launchd see the Apple man page. Launchd registered services require to have "root:wheel" ownership and executable permissions.

The recommended installation path for the *boom* agent is `/opt/boom/agent`.

Please change the ownership and permissions of the extracted files before running the install script:

```
sudo chown -R root:wheel /opt/boom/agent
sudo chmod -R 755 /opt/boom/agent
```

Server

The *boom* server installation script does not fully support Mac OS X yet.

The supplied boom_server.plist file contains an example configuration for launchd. You can perform the installation manually and skip the step that registers 'rc' scripts. Instead of that, please register the *boom* server by using the supplied boom_server.plist configuration file for launchd.

The boom_server.plist file expects that the *boom* server is unpacked in `/opt/boom/boom_server/`. If you have the server installed in another directory, please change the ProgramArguments parameters in the plist file.

```
sudo chown -R root:wheel /opt/boom/server
sudo chmod -R 755 /opt/boom/server
sudo cp boom_server.plist  /Library/LaunchDaemons/
sudo launchctl load /Library/LaunchDaemons/boom_server.plist
```

You can stop and start the *boom* server by using:

```
sudo launchdctl stop com.blixx.boom.server
sudo launchdctl start com.blixx.boom.server
```

2.1.5.2. Linux

UI Client

The UI tries to integrate the default browser in the statistics and browser view with a component named xulrunner. Firefox introduced a newer version of the integration code that is not compatible with the Eclipse RCP version of the UI. If the statistics and browser view show that the browser can not be started internally it is necessary to download and install the xulrunner version 1.9.x from http://ftp.mozilla.org/pub/mozilla.org/xulrunner/releases/.

Extract the xulrunner files e.g. to `/opt/boom/boom_gui/xulrunner` and add in the `<boom/boom_gui>/boomgui.ini` file after the `-vmargs` argument the new line:

```
-Dorg.eclipse.swt.browser.XULRunnerPath=/opt/boom/boom_gui/xulrunner/
```

Some of the latest versions of the cairo library on Linux are no longer fully compatible. The UI might crash on some operation and cairo library errors are displayed in the log files or as console output. In this case please add a following java vm attribute into the boomgui.ini

file as last line:

```
-Dorg.eclipse.swt.internal.gtk.cairoGraphics=false
```

2.2. Installation Steps

All *boom* components are packaged in the server package zip file. The first installation step therefore is always the server installation. The installation of all other components can only be performed after the server is installed and running.

2.2.1. Server Installation

Worldline's latest *boom* can be downloaded from https://bes-worldline.com/en-us/downloads-main-menu/8-downloads.html or contact boom support (Email: support@bes-worldline.com, Phone: +49 69 256 552 201). Unzip the software package to the folder in which the server should be installed (e.g. `C:\Program Files\boom\server` or `/opt/boom/server`).

The installer is a Java based program. Java 17 needs to be available at *boom* Server.

Check availability of the Java 17 version:

```
# <java_17_repository_path>/bin/java -version
```

Make sure that your database is up and running.

2.2.1.1. MySQL:

For MySQL create a database user(s) that has rights to create databases or create the databases *boom* and BOOM_PERF manually and give the database user full access rights on these two databases.

Check the mysql installation:

```
# /etc/init.d/mysql status
```

After the installation of the mysql server the user "root" has no password set, so this should be configured first. For the example steps below we used highly insecure passwords like mysql which should not be used on production machines!

2.2.1.1.1. For MySQL Version 8

The initial root account may or may not have a password after installation. Please refer to MySQL documentation about default root password: https://dev.mysql.com/doc/refman/8.0/en/default-privileges.html

1. Open a terminal/shell
2. Run:

```
mysql -u root -p
Enter password: (enter root password here)

CREATE DATABASE BOOM;
CREATE DATABASE BOOM_PERF;
CREATE USER 'boom'@'%' IDENTIFIED BY 'secret';
GRANT ALL PRIVILEGES ON BOOM.* TO 'boom'@'%';
CREATE USER 'boom_perf'@'%' IDENTIFIED BY 'secret';
GRANT ALL PRIVILEGES ON BOOM_PERF.* TO 'boom_perf'@'%';
FLUSH PRIVILEGES;
exit
```

2.2.1.1.2. For MySQL before Version 8

1. Open a terminal/shell
2. Run:

```
mysql
grant all on *.* to 'root'@'%' identified by 'mysql';
grant all on *.* to 'root'@'localhost' identified by 'mysql';
grant all on *.* to 'root'@'127.0.0.1' identified by 'mysql';
flush privileges;
exit
```

3. Run:

```
mysql -u root --password=mysql
create database BOOM;
create database BOOM_PERF;
grant all on BOOM.* to 'boom'@'%' identified by 'secret';
grant all on BOOM_PERF.* to 'boom_perf'@'%' identified by 'secret';
flush privileges;
exit
```

Check once again if users and passwords are correct. If the login works correctly you will see the greeting message from mysql.

Run:

```
mysql -u boom --password=secret
exit
```

Run:

```
mysql -u boom_perf --password=secret
exit
```

2.2.1.2. Oracle

For Oracle create two users with identical rights as described in `/opt/boom/server/ORACLE_SQL/step1_create_boom_db.sql` to have distinct schemas for *boom* and BOOM_PERF. It is recommended to use separate tablespaces for *boom* and BOOM_PERF to be able to manage the server and performance data independently.

The steps to correctly create and configure an Oracle database with the right sizes and user access rights are more complicated than for MySQL. Therefore these steps should be done by a person with experience in the Oracle database administration.

2.2.1.3. Installing on Linux/Unix Systems

Invoking the installation script

The installation needs to be performed by an user that has administrative rights (i.e. root) to be able to add files to the system startup directories.

1. Copy the server zip file to `/tmp`
   Choose an install directory (e.g. `/opt/boom/server`)
   Unzip the zip File in the install directory

   ```
   # mkdir -p /opt/boom/server
   # cd /opt/boom
   # unzip /tmp/server_x_x.zip
   # cd server
   ```

2. Adapt "JAVA_BIN" entry (Java 17) in configuration file boom_srv.cfg

   ```
   # vi boom_srv.cfg
   JAVA_BIN=<java_17_repository_path>/bin
   ```

3. Change the permissions of the install script to be able to execute it

   ```
   # chmod 750 install.sh
   ```

4. Run the installation script "install.sh" by entering

   ```
   # ./install.sh
   ```

The script will trigger the installer which queries the configuration settings, updates the configuration files and performs all steps that are necessary to install the server. The installer can handle most Linux distributions, HP-UX, SUN Solaris and Windows. In most cases the default values that are proposed by the installer can be used without change, so only the database type and authentication information needs to be entered. If necessary the configuration can always be changed by running the installer again or by editing the property files as described below.

The installation script will ask some questions:

```
OS type: UNIX
Stop server? (yes|no|cancel)? [Y] Y
JAVA_BIN=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.0.x86_64/jre/bin
boom properties:
#AGENT_PORT=23021
#AUDITENABLE=false
#AUDITLOGCOUNT=5
#AUDITLOGDIR=.
#AUDITLOGSIZE=10
#AUTO_APPROVAL=false
#AUTOCLOSE_FINISHED_ALERTS=false
#AUTODETECT_EXTERNAL_HOSTS_FROM_SLAVES=false
#CLUSTER_NODES=
#GUI_PORT=23022
#HOSTNAMES_LOWERCASE=false
#HTTP_PORT=8888
#IGNORE_PERF_DATA=false
#INSTRUCTION_SERVER=http://wiki.local/
#LOGDIR=.
#MAIN_SERVER_IP=
#MAIN_SERVER_NAME=
#MAIN_SERVER_PORT=23020
#SHORT_LABELS=false
#SSL_KEYPASS=
#SSL_KEYSTORE=
#SSL_PASSWORD=
AUTO_ARCHIVE_DAYS=60
AUTO_CLOSE_DAYS=30
AUTO_DELETE_DAYS=90
AUTODETECT_EXTERNAL_HOSTS=true
HB_CTIMEOUT=5000
HB_INTERVAL=10
HB_RTIMEOUT=10000
LOGLEVEL=1

Properties are correct? press 'y' to continue... (yes|no|cancel)? [Y]

Checking database...

DB properties: ./db.props
DbName=BOOM
Host=localhost
Driver=com.mysql.jdbc.Driver
Password=
Login=root
Port=3306

Please check connection properties:

DB type (mysql|oracle) [mysql]:
DB Host [localhost]:
DB Port [3306]:
DB Name [BOOM]:
DB User [root]:
```

```
DB Password []: <insert a password>
....

Database is online...

Create database (existing data will be destroyed!)? (yes|no|cancel)? [Y]

... <a lot of DB output> ...

Main Database creation done.

Checking CREATE TABLE rights
CREATE TABLE EVENTS_TEST AS SELECT ID FROM EVENTS WHERE STATE='Z'
OK.

Checking DROP TABLE rights
DROP TABLE EVENTS_TEST
OK.

PERF DB properties: ./db_perf.props

DbName=BOOM_PERF
Host=localhost
Driver=com.mysql.jdbc.Driver
Password=
Login=boom_perf
Port=3306

Check Only? Y- check only,N - change settings (yes|no|cancel)? [Y] N

Please check connection properties:

DB type (mysql|oracle) [mysql]:
DB Host [localhost]:
DB Port [3306]:
DB Name [BOOM_PERF]:
DB User [boom_perf]:
DB Password []: <insert a password>
Checking connection
PERF DB is OK
chmod 755 boom_srv
chmod -R 755 ux-daemon.cfg
chmod -R 755 srv/packages
chmod 660 boom.props
chmod 660 db.props
chmod 660 db_perf.props
File is up to date.
File has been updated.
Install boot-time init scripts? (yes|no|cancel)? [Y]
who -r
runlevel=3
RC_DIR=/etc/rc
INIT_DIR=/etc/init.d
RC_DIR=/etc/rc
sh -c "init_s=`ps -ef | grep init | grep -v grep | awk '{print $2}' | grep -w \"1\"`; echo ${init_s}"
SYS_INIT: initd
File is up to date.
cp -fp ux-daemon.cfg/boom_srv /etc/init.d/
rm -f /etc/rc0.d/K*boom_srv
rm -f /etc/rc0.d/S*boom_srv
ln -s /etc/init.d/boom_srv /etc/rc0.d/K01boom_srv
```

```
rm -f /etc/rc1.d/K*boom_srv
rm -f /etc/rc1.d/S*boom_srv
ln -s /etc/init.d/boom_srv /etc/rc1.d/K01boom_srv
rm -f /etc/rc2.d/K*boom_srv
rm -f /etc/rc2.d/S*boom_srv
ln -s /etc/init.d/boom_srv /etc/rc2.d/K01boom_srv
rm -f /etc/rc3.d/K*boom_srv
rm -f /etc/rc3.d/S*boom_srv
ln -s /etc/init.d/boom_srv /etc/rc3.d/S99boom_srv
rm -f /etc/rc4.d/K*boom_srv
rm -f /etc/rc4.d/S*boom_srv
ln -s /etc/init.d/boom_srv /etc/rc4.d/S99boom_srv
rm -f /etc/rc5.d/K*boom_srv
rm -f /etc/rc5.d/S*boom_srv
ln -s /etc/init.d/boom_srv /etc/rc5.d/S99boom_srv
rm -f /etc/rc6.d/K*boom_srv
ln -s /etc/init.d/boom_srv /etc/rc6.d/K01boom_srv
insserv boom_srv
chkconfig --add boom_srv
chkconfig boom_srv on
start boom server? (yes|no|cancel)? [Y]y
INIT_DIR=/etc/init.d
sh -c "init_s=`ps -ef | grep init | grep -v grep | awk '{print $2}' | grep -w \"1\"`; echo ${init_s}"
SYS_INIT: initd

starting Server
/etc/init.d/boom_srv start
BOOM_SERVER=/opt/boom/boom_server
Starting boom Server
```

Manual Installation on Unix

For platforms that are not fully supported by the installer or if for some reasons the install program should not be used, the following steps need to be performed manually to install the server:

- Create the databases and initial table data by executing either the MySQL or the Oracle SqlPlus command

  ```
  cd /opt/boom/server/MYSQL_SQL
  mysql -u root -p <./step1_create_boom_db.sql
  mysql -u root -p <./step2_createOutageTables.sql
  mysql -u root -p <./step3_createPGTables.sql
  ```

  or

  ```
  cd /opt/boom/server/ORACLE_SQL
  sqlplus <user>/<password>@<tnsname> <./step1_create_boom_db.sql
  sqlplus <user>/<password>@<tnsname> <./step2_createOutageTables.sql
  sqlplus <user>/<password>@<tnsname> <./step3_createPGTables.sql
  ```

  Existing tables will be deleted and re-created!

- For MySQL edit the database configuration files <server directory>:db.props, db_perf.props and update them with the MySQL instance and logon information.
  For Oracle move the database configuration files from `/opt/boom/server/ORACLE_SQL/db.props` to `/opt/boom/server/db.props` and the same for `db_perf.props`.
  Update the files with the Oracle instance and logon information.

- Edit the server management script boom_srv. Set the BOOM_SERVER variable to the installation directory and

PATH information as described in the file.

- If required edit the boom_srv file to change java startup parameters or hardcode a PATH to a specifc JRE as described in the chapter *boom* Server Configuration Files.
- Edit the server rc scripts `ux_daemon.cfg/boom_srv` exactly with the same information as above.
- Copy the server rc script from `ux_daemon.cfg/boom_srv` to the init.d directory of the system. The exact location varies depending on the platform.
- Create the symbolic links in the rc level directories as required and use the insserv command if available. The location and the meaning of these vary for the different platforms.
- Verify that all configurations are done correctly by starting the server with e.g.

  ```
  /etc/init.d/boom_srv start
  ```

  Check if any errors showed up or are logged in the server's log file in `<install_dir>/BOOMServer_<date>.log`

2.2.1.4. Installing on Microsoft Windows systems

Invoking the installation script

On 64bit Windows systems make sure that you have a 64bit Java runtime installed. Verify that Java is in the System Search Path.

The installation needs to be performed by a user that has administrative rights (i.e. administrator) to be able to add the Windows Service. On platforms with enabled UAC like Windows Vista, Server 2008 and later the commands must be executed with advanced privileges.

Open a command window by selecting Start→Run... or Start→Search and enter cmd, press OK (or right click and choose "Run as Administrator).

In the command window change to the installation directory and adapt "JAVA_BIN" entry (Java 17) in configuration file boom_srv.cfg by entering

```
> cd <install dir> (e.g. cd c:\Program Files\boom\server)
> notepad boom_srv.cfg
JAVA_BIN="<java_17_repository_path>\bin"
```

and run the installer by entering

```
> install.cmd
```

The script will trigger the installer which queries the configuration settings, updates the configuration files and performs all steps that are necessary to install the server.

In most cases the default values that are proposed by the installer can be used without change, so only the database type and authentication information needs to be entered. If necessary the configuration can always be changed by running the installer again or by editing the property files as described below.

Manual Installation on Windows

On 64bit Windows systems make sure that you have a 64bit Java runtime installed. Verify that Java is in the System Search Path.

- Create the databases and the initial table data by executing either the MySQL or the Oracle SqlPlus command

```
> cd C:\Program Files\boom\server\MYSQL_SQL
> mysql -u root -p <.\step1_create_boom_db.sql
> mysql -u root -p <.\step2_createOutageTables.sql
> mysql -u root -p <.\step3_createPGTables.sql
```

or

```
> cd C:\Program Files\boom\server\ORACLE_SQL
> sqlplus <user>/<password>@<tnsname> <.\step1_create_boom_db.sql
> sqlplus <user>/<password>@<tnsname> <.\step2_createOutageTables.sql
> sqlplus <user>/<password>@<tnsname> <.\step3_createPGTables.sql
```

Existing tables will be deleted and re-created!

- For MySQL edit the database configuration files `C:\Program Files\boom\server\db.props`, `db_perf.props` and update them with the MySQL instance and logon information.
  For Oracle move the database configuration files from `C:\Program Files\boom\server\ORACLE_SQL\db.props` to `C:\Program Files\boom\server\db.props` and the same for `db_perf.props`. Update the file with the Oracle instance and logon information.
- Open a command window by selecting Start → Run... and enter cmd, press OK.
  In the command window change to the installation directory by entering
  `cd <install dir>` (e.g. `cd c:\Program Files\boom\server`)
  Depending on the installed Java version (32 or 64bit) register the *boom* Server as Windows Service with the command
  32bit:

```
> booms_service_32.exe -i
```

64bit:

```
> booms_service_64.exe -i
```

On platforms with UAC like Windows Vista, Server 2008 and later the command must be executed with advanced privileges.

- Verify that all configurations are done correctly by starting the server with the Windows Service Control Panel. Check if any errors showed up or are logged in the server's log file in `<install_dir>/BOOMServer_<date>.log`

2.2.2. Server running as "non-root"

The server process normally runs under user root on Unix systems and under the System account on Windows systems. In UNIX environments that are highly security sensitive it may be necessary to limit the number of processes that have full root permissions. The installation always has to be performed as root.

Server can be configured to run under user that does not have root permissions. However, the user needs specific "sudoers" (sudo) rights in order to start/stop the *boom* service. The configuration actions have to be performed as root.

- **Configure the "non-root" usage after installation of the server (Here: also group of <server_install_directory> will be changed)**
  - Create a "non-root" group (for example "boomerang")
  - Create a "non-root" user (for example "boomerang") referring to "non-root" group

- Adapt the "sudoers" configuration (password less) according to the OS related settings. For CentOS and user "boomerang" the configuration looks as follows:

  ```
  boomerang NOPASSWD /bin/systemctl * boom_srv
  ```

- Change the ownership (user and group) of the *boom* Server home directory:

  ```
  chown -R boomerang:boomerang /opt/boom/server/
  ```

- Modify "HTTP_PORT" entry in *boom* Server property file to a "non-root" usable port (for example 8443):

  vi /opt/boom/server/boom.props

  ```
  HTTP_PORT=8443
  ```

- Insert "User" entry to service file:

  vi /etc/systemd/system/boom_srv.service

  ```
  [Service]
  User=boomerang
  Type=forking
  PIDFile=/opt/boom/server/boom_server.pid
  ...
  ```

- Reload modified service file:

  ```
  systemctl daemon-reload
  ```

- Start the server as "non-root" user on UNIX systems

  by restarting the *boom* Server:

  ```
  <server_install_directory>/boom_srv -stop
  <server_install_directory>/boom_srv -start
  ```

This procedure works from a general standpoint and most use-cases are covered, however if the boom server itself is required to perform any action which need elevated rights, it won't be able to perform so. Agent running under the root user on their systems are not effected. For agents running under non-root user, please refer to the chapter Agent running as "non-root".

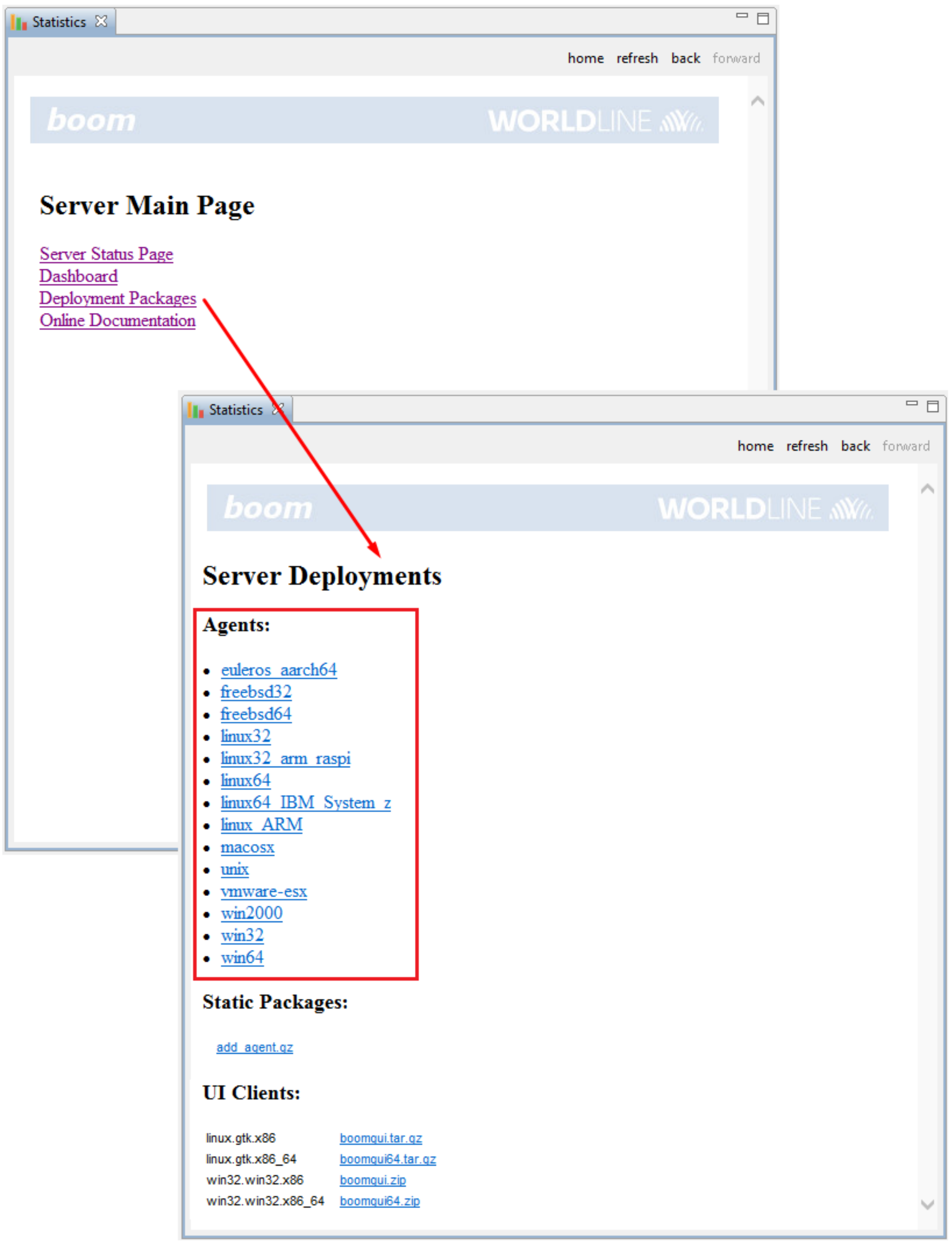
Due to UNIX's restriction root only can bind to low ports (<1024) the boom's "HTTP_PORT" port setting must be chosen from high port range (>1024).

2.2.3. Agent Installation

The agent packages are placed on the server and can be downloaded from its built-in web server. The server will prepare the agent configuration files and generate the packages when a download request is received. Therefore it is mandatory to once download an agent from this web server to have all packages build. Afterwards the packages can also be copied by other means to the target systems.

To download the agent point your browser to the following address:

https://your.boom.server



Server Main Page

Server Status Page
Dashboard
Deployment Packages
Online Documentation

Server Deployments

Agents:

- euleros_aarch64
- freebsd32
- freebsd64
- linux32
- linux32_arm_raspi
- linux64
- linux64_IBM_System_z
- linux_ARM
- macosx
- unix
- vmware-esx
- win2000
- win32
- win64

Static Packages:

add_agent.gz

UI Clients:

linux.gtk.x86	boomgui.tar.gz
linux.gtk.x86_64	boomgui64.tar.gz
win32.win32.x86	boomgui.zip
win32.win32.x86_64	boomgui64.zip

Unzip the *boom* agent package to the folder in which the agent should be installed (e.g. `C:\Program Files\boom\agent` or `/opt/boom/agent`).

2.2.3.1. Installing on Linux/Unix Systems

Invoking the installation script

The recommended installation path for the *boom* Agent is `/opt/boom/agent`, we also recommend performing the installation as root user. In case the boom Agent should run under a different system user (non-root user) please refer to Agent running as "non-root".

1. Copy the agent zip file to /tmp and change to the install directory

```
e.g.
# cd /tmp
# wget https://your.boom.server/deploy/agent/<OS system>_boom_agent.zip
# mkdir /opt/boom/agent
# cd /opt/boom/agent
```

2. Unzip the zip file in the install directory

```
# pwd
/opt/boom/agent
# unzip /tmp/<OS system>_boom_agent.zip
```

3. Change the permissions of the install script to be able to execute it.

```
# chmod 750 install
```

4. Run the installation script "install". It will query the configuration settings, update the configuration files and perform all steps that are necessary to install the agent. The script can handle most Linux distributions, HP-UX, SUN Solaris and VMWare ESX.
   For others some manual steps might be required e.g. to integrate the server with the system start-up and shutdown routines.

```
# ./install
Set BOOM_ROOT=/opt/boom/agent
Install Agent to /opt/boom/agent?
(yes|no|cancel) [Y]? Y
Information: Checking if Agent is running and stop it...
Please specify the agent port that should be used.
[23021]?                --> common for all agents / use the default if available
Please specify the server port that should be used (common for all agents).
[23020]?                --> use the default if available
Please enter the full qualified name of the Management Server.
[monserver]?
DNS Lookup of monserver failed. Continue anyway?
(yes|no|cancel) [Y]?
The path to Java can be hardcoded in the startup scripts (i.e. the rc start/stop files).
If this is not selected, ensure that the path to Java is in the PATH variable of the init/systemd
process.
Should the path be hardcoded?
(yes|no|cancel) [Y]?
Please enter the full path to the java executable.
[/usr/bin/]?/usr/bin/
Should the agent be started under the default [root] user?
(yes|no|cancel) [Y]?yes
Should the script add the according entries in the run level directories to start and stop the
agent automatically at system boot time?
(yes|no|cancel) [Y]?
At which runlevel should the agent be started or stopped (2-5)?
[3]?

----------------------------
Collection of information finished.

Selection Summary:
Installation directory: /opt/boom/agent
Server:            monserver
Server Port:       23020
Server IP:         Server_IP
Agent Port:        23021
NAT/Shared IP:     false
Start at Runlevel: 3
Log Level:         1
Size per Logfile:  10

The path to java (/usr/bin/) will be hardcoded in the script files.
The agent start/stop entries will be entered in the runlevel directories to startup at level 3

---------------------------
Perform this installation
(yes|no|cancel) [Y]?
Installation finished.

Start the agent?
(yes|no|cancel) [Y]?

Starting Agent
```

5. Approve the agent in the *boom* GUI

```
Select the "Hosts" view
Open the "Not approved Agents" Group
Select the new agent
right-click --> Approve Agent
```

2.2.3.2. Silent Installation on Linux/Unix Systems

Instead of parsing the configuration parameters through the dialog during installation, it is also possible to install an agent silently. Therefore, the steps for a convenient installation should be followed until the install script is called.

In order to see the details of silent installation, the help of the install script can be called.

```
./install --help
./install -s
./install -s -us <non-root user> -j <path2java>
```

When using only the -s flag the default settings of the installation script will be used. For other settings such as path to java or a different system user to run the process, the appropriate flags should be used (see below). The installation process works equally and the output of the settings which where used are also displayed to the console.

Flag	Parameter	Default
-s	silent install	
-u	uninstall	
-a	CLUSTER_NODES	-
-ap	AGENT_PORT	23021
-c	IS_CLUSTER	false
-h	AGENT_HOST	
-i	AGENT_IP	
-ll	LOGLEVEL	1
-lc	LOGCOUNT	5
-ls	LOGSIZE	10
-j	JAVA_PATH	(done automatically)
-n	SHARED_IP	false
-rl	RUN_LEVEL	-
-si	MAIN_SERVER_IP	-
-sh	MAIN_SERVER_NAME	-
-sp	MAIN_SERVER_PORT	-
-us	AGT_USER	root

2.2.3.3. Agent running as "non-root"

The agent process normally runs under user root on Unix systems and under the System account on Windows systems. In UNIX environments that are highly security sensitive it may be necessary to limit the number of processes that have full root permissions.

Agent can be configured to run under user that does not have root permissions. However, the user needs specific "sudoers" (sudo) rights in order to start/stop the *boom* service. There are two possibilities to configure the Agent to run as "non-root" but the actions have to be performed as root:

- **Configure the "non-root" user directly during agent installation (Here: group of <agent_install_directory> remains unchanged)**
  - Create a user, we will name it "boomerang", before starting the agent installation. Run the installer as usual under root and enter the username here:

```
Should the agent be started under the default [root] user?
(yes|no|cancel) [Y]? no
Please enter the user.
[]? boomerang
Should the script add the according entries in the run level directories to start and stop the
agent automatically at system boot time?
(yes|no|cancel) [Y]?
At which runlevel should the agent be started or stopped (2-5)?
[3]?
Should the script add the according entries in the run level directories to start and stop the
agent automatically at system boot time?
(yes|no|cancel) [Y]?
At which runlevel should the agent be started or stopped (2-5)?
[3]?

----------------------------
Collection of information finished.

Selection Summary:
Installation directory: /opt/boom/agent
Server:             monserver
Server Port:        23020
Server IP:          Server_IP
Agent Port:         23021
NAT/Shared IP:      false
Start at Runlevel: 3
Log Level:          1
Size per Logfile:  10
Start the Agent as User: boomerang
The path to java (/usr/bin/) will be hardcoded in the script files.
The agent start/stop entries will be entered in the runlevel directories to startup at level 3

---------------------------
```

  - Adapt the "sudoers" configuration (password less) according to the OS related settings. For CentOS and user "boomerang" the configuration looks as follows:

```
boomerang NOPASSWD /bin/systemctl * boom_agt
```

- **Configure the "non-root" usage after installation of the agent (Here: also group of <agent_install_directory> will be changed)**

  In case you want to create the "non-root" user after the agent installation, please proceed as follows:

  - Create a "non-root" group (for example "boomerang")

- Create a "non-root" user (for example "boomerang") referring to "non-root" group
- Adapt the "sudoers" configuration (password less) according to the OS related settings. For CentOS and user "boomerang" the configuration looks as follows:

  ```
  boomerang NOPASSWD /bin/systemctl * boom_agt
  ```

- Change the ownership (user and group) of the *boom* Agent home directory:

  ```
  chown -R boomerang:boomerang /opt/boom/agent/
  ```

- Insert "User" entry to service file:

  vi /etc/systemd/system/boom_agt.service

  ```
  [Service]
  User=boomerang
  Type=forking
  PIDFile=/opt/boom/agent/boom_agent.pid
  ...
  ```

- Reload modified service file:

  ```
  systemctl daemon-reload
  ```

- Start the agent as "non-root" user on UNIX systems

  by setting the variable USER in the boom_agt.cfg file in the agent installation directory

  ```
  vi <agent_install_directory>/boom_agt.cfg

  # agent user
  USER=boomerang
  ```

  and restart the *boom* Agent:

  ```
  <agent_install_directory>/boom_agt -stop
  <agent_install_directory>/boom_agt -start
  ```

The "non-root" agent is **only supported on UNIX systems**. The agent for Windows must run under the System account or an Account which belongs to the Administrators group!

Some of the monitoring policies and packages have to be adopted to "non-root" restrictions (default SNMP trap receiver port cannot be used, no access to certain log files or executables, …). Hint: Due to UNIX’s restriction root only can bind to low ports (lower 1024) the SNMP trap receiver port must be chosen from high port range (greater 1024) and a port rerouting has to be arranged.

2.2.3.4. Installing on Windows systems

Invoking the installation procedure

On the Windows platform perform the following steps to install the agent:

- **Login as Administrator**
  To install the agent an Administrator account is needed. After the installation the BoomAgent windows service has to run either under a local system account (default) or under a technical user, who belongs to the Administrators group.
- **Install the *boom* Agent**
  - Download the agent package fitting your Windows and Java architecture (32 or 64bit) from i.e.
    https://your.boom.server/deploy/agent/win32_boom_agent.zip
    or
    https://your.boom.server/deploy/agent/win64_boom_agent.zip
  - unzip to a folder of your choice (i.e. `C:\Program Files\boom\agent`)
  - Verify that the *boom* server name and port settings are correctly entered in the configuration file conf/agent.conf. All other values will be filled automatically by the agent.

    Do not change this file while the agent is running, since the agent will overwrite it.

  - Open a command window by selecting Startmenu → Run..., enter cmd, press OK.
  - In the command window change to the installation directory by entering
    `cd <install dir>` (e.g. `cd C:\Program Files\boom\agent`)
  - Register the *boom* agent as Windows service with the command:

    ```
    > booma_service_32.exe -i
    ```

    or

    ```
    > booma_service_64.exe -i
    ```

    On platforms with UAC like Windows Vista, Server 2008 and later the command must be executed with advanced privileges.

  - Start the *boom* Agent Windows service

    ```
    > net start BoomAgent
    ```

    or change to the Windows service and start the BoomAgent service manually.
  - Approve the Agent in the *boom* GUI

    ```
    Select the "Hosts" view
    Open the "Not approved Agents" Group
    Select the new agent
    right-click --> Approve Agent
    ```

- Check if any errors showed up or are logged in the agent's log file in `<install_dir>/Agent_<date>.log`

2.2.3.5. Postinstallation Tasks

Assignment Group "boom-Basic"

For full agent functionality, deploy the assignment group "boom-Basic" to the agent. This mandatory group contains the policy "boom_Messages" for internal messages, the package "boomJavaMonitors" for agent specific Java monitoring classes, and Inventory for the discovery of inventory information.

2.2.4. Agent Uninstallation

2.2.4.1. Uninstalling the Agent on Unix Systems

Execute the following steps to uninstall an agent:

1. Stop the agent

   ```
   #/etc/init.d/boom_agt stop
   ```

2. Delete the installation folder, init.d and rc scripts

   ```
   # rm -rf /opt/boom/agent
   # rm -f /etc/init.d/boom_agt
   # rm -f /etc/rc?.d/*boom_agt
   ```

3. If necessary delete the entry for the management server from `/etc/hosts`
4. Delete the agent in the *boom* GUI

2.2.4.2. Uninstalling the Agent on Microsoft Windows Systems

Execute the following steps to uninstall an agent:

1. Stop the agent

   ```
   > net stop BoomAgent
   ```

2. Execute the setup file from a command window to uninstall BoomAgent service
   a. Change to the installation folder
   b. Execute the setup file with -u flag (uninstall) to delete the windows service

      ```
      > booma_service_32.exe -u
      ```

      or

      ```
      > booma_service_64.exe -u
      ```

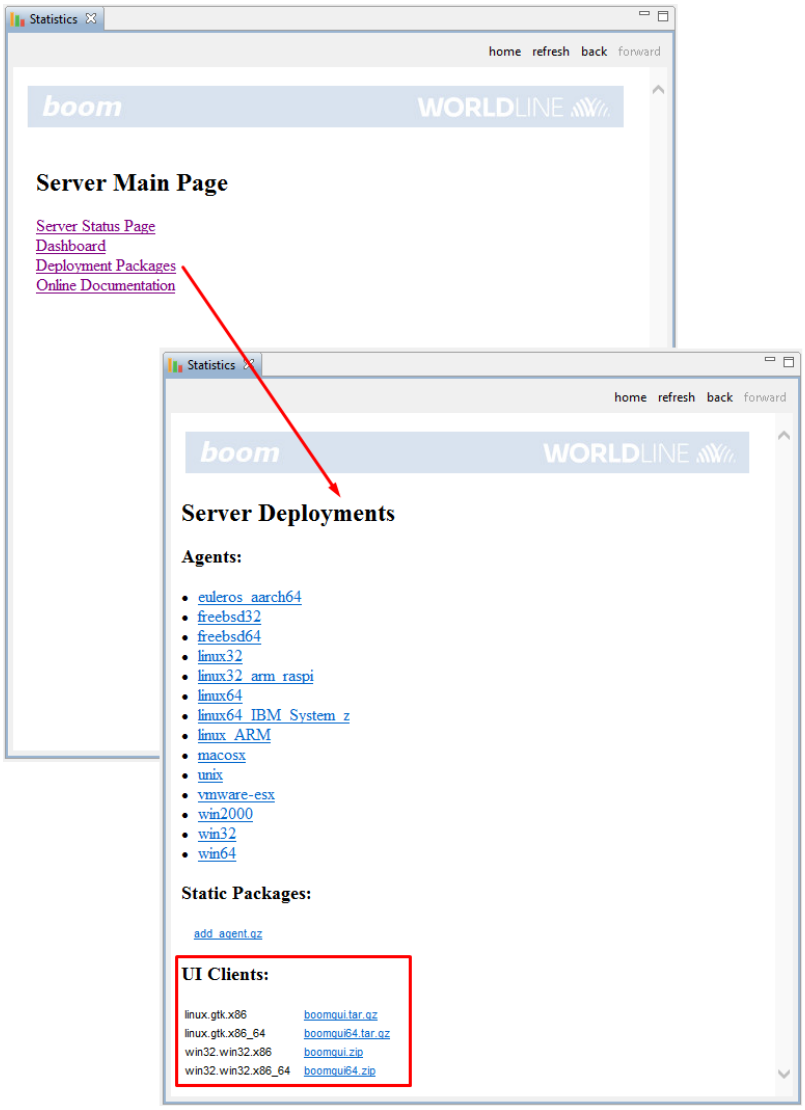
3. Delete the installation folder
   i.e. `C:\Program Files\boom\agent`
4. If necessary delete the entry for the management server from `%windir%\system32\drivers\etc\hosts`
5. Delete the agent in the *boom* GUI

2.2.5. *boom* Client (User Interface) installation

Download the boomgui package that suits your platform from the server’s web page as described for the agent.

To download the client package point your browser to the following address:

https://your.boom.server



Server Main Page

Server Status Page
Dashboard
Deployment Packages
Online Documentation

Server Deployments

Agents:

- euleros_aarch64
- freebsd32
- freebsd64
- linux32
- linux32_arm_raspi
- linux64
- linux64_IBM_System_z
- linux_ARM
- macosx
- unix
- vmware-esx
- win2000
- win32
- win64

Static Packages:

add_agent.gz

UI Clients:

linux.gtk.x86	boomgui.tar.gz
linux.gtk.x86_64	boomgui64.tar.gz
win32.win32.x86	boomgui.zip
win32.win32.x86_64	boomgui64.zip

1. Select the link **Deployment packages** and download the Client package that suits your platform.
2. Unzip the package to the folder in which the user interface should be installed (e.g. `C:\boom\boom_gui` or `/opt/boom/boom_gui`). No further installation steps are necessary.
3. Java needs to be in the current user's search path in order to startup the GUI.
4. Start the graphical user interface by executing the boomgui program

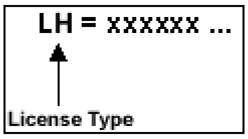
2.3. Licensing

In order to add new licenses you need to run the *boom* user interface. If you do not have installed the *boom* UI, you need to download the latest boomgui package from the server by pointing the browser to the following address: https://your.boom.server.

For more information please contact *boom* support (Email: support@bes-worldline.com).

With version 4.0 the license counting and checking has been changed.

License Types

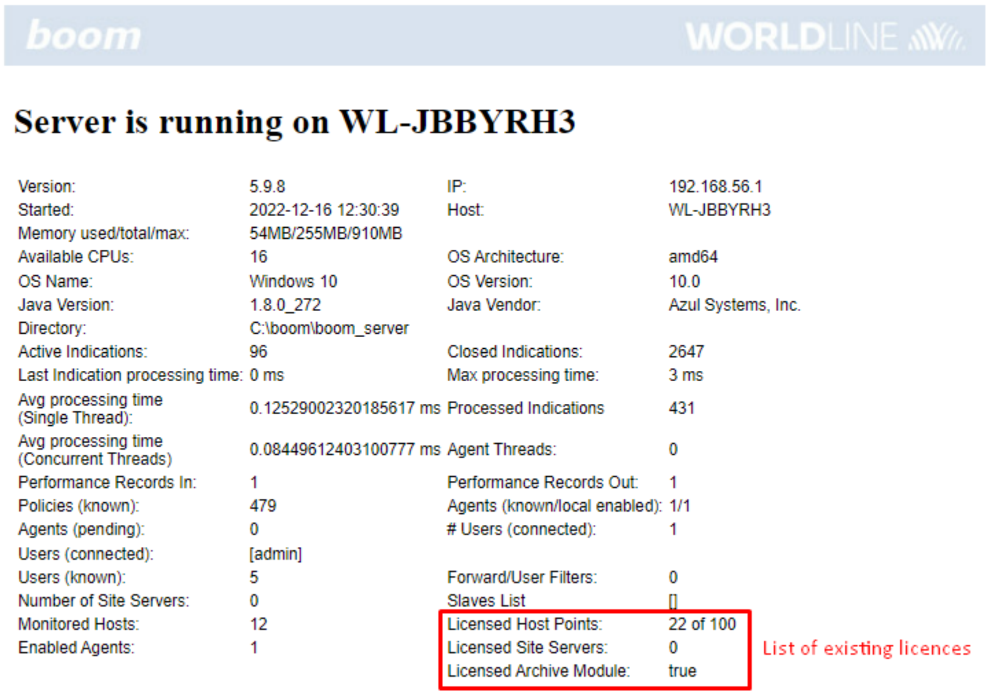


LH	Licensed Host Points:	each Agent requires 10 points, each agentless monitored system or device requires 1 point
BB	Licensed Site Servers:	the number of integrated servers like boom2boom slaves, om2b slaves, mirror servers and similar

Check existing Licenses

A list of existing licenses can be found in the Statistics View in the *boom* User Interface or by pointing your web browser to the servers status page at:

https://your.boom.server

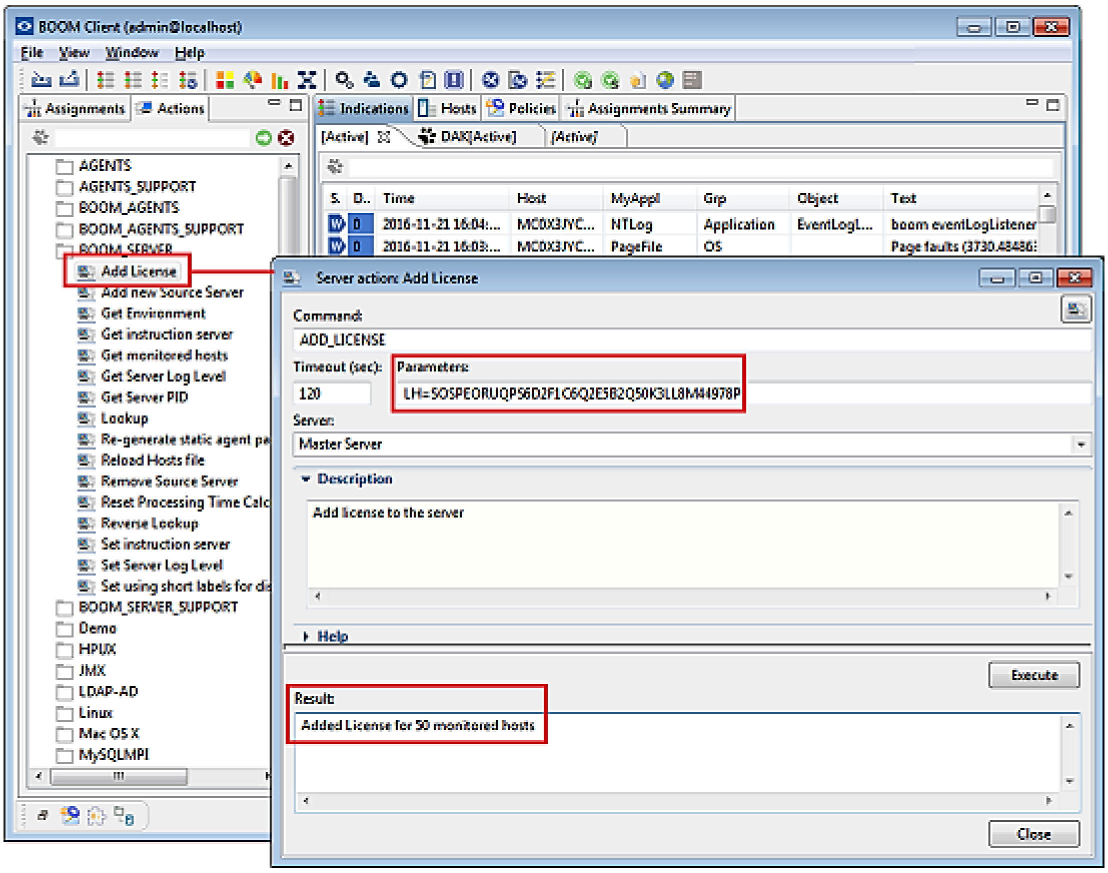


Server is running on WL-JBBYRH3

Version:	5.9.8	IP:	192.168.56.1
Started:	2022-12-16 12:30:39	Host:	WL-JBBYRH3
Memory used/total/max:	54MB/255MB/910MB		
Available CPUs:	16	OS Architecture:	amd64
OS Name:	Windows 10	OS Version:	10.0
Java Version:	1.8.0_272	Java Vendor:	Azul Systems, Inc.
Directory:	C:\boom\boom_server		
Active Indications:	96	Closed Indications:	2647
Last Indication processing time:	0 ms	Max processing time:	3 ms
Avg processing time (Single Thread):	0.12529002320185617 ms	Processed Indications	431
Avg processing time (Concurrent Threads)	0.08449612403100777 ms	Agent Threads:	0
Performance Records In:	1	Performance Records Out:	1
Policies (known):	479	Agents (known/local enabled):	1/1
Agents (pending):	0	# Users (connected):	1
Users (connected):	[admin]		
Users (known):	5	Forward/User Filters:	0
Number of Site Servers:	0	Slaves List	[]
Monitored Hosts:	12	Licensed Host Points:	22 of 100
Enabled Agents:	1	Licensed Site Servers:	0
		Licensed Archive Module:	true

Add New License

To add one or more new licenses you have to perform the following *boom* server action:



Command: ADD_LICENSE

Timeout (sec): 120

Parameters: LH=5OSPEORUQPS6D2F1C6Q2E5B2Q50K3LL8M44978P

Server: Master Server

Description: Add license to the server

Result: Added License for 50 monitored hosts

After the 'Add License' Action has finished, go to the *boom* statistics view to see the list of all existing licenses (see paragraph Check existing Licenses above).

2.4. *boom* Upgrade

Upgrade from Version 5.x to 5.10.0

Worldline’s latest *boom* can be downloaded from https://bes-worldline.com/en-us/downloads-main-menu/8-downloads.html or contact boom support (Email: support@bes-worldline.com, Phone: +49 69 256 552 201).

All required components to upgrade the *boom* software are part of the server installation package available on download contact mentioned above. The upgrade of the *boom* components Server/Agent/GUI is being performed by the following steps:

1. Upgrade the server as described in the following chapters. In Primary-Backup/Master-Slave Environments always upgrade all **Backup/Slave Servers first**, before the Primary/Master Server is upgraded.

   The GUI based Import MPI functionality offers features like the content compare for policies and binaries, reducing greatly the efforts that are necessary to re-check custom changes compared to the import functionality that was formerly used in the upgrade process. Therefore the upgrade of actions, policies, assignment groups and packages has been removed from the server upgrade script and should be performed manually as step 3.

2. Upgrade the GUI. With this version the GUI comes with updated libraries that require to upgrade by re-installation of the new GUI package as described in the chapter *boom* Client (User Interface) installation. The use of the automatic Upgrade by simply logging into the new server with the old GUI version and then follow the steps described in the chapter *boom* Client / UI Upgrade will be possible for later versions again.
3. Use now the new GUI to upload the latest versions of actions, assignment groups, packages, and policies with the Import MPI functionality using the temporary upgrade directory `<boom_server_temp>/srv`.
4. As last step upgrade the *boom* agent via the GUI as described in chapter *boom* Agent Upgrade and redeploy the

packages that were upgraded (i.e. boomJavaMonitors, Inventory, SNMP)

2.4.1. *boom* Server Upgrade

After downloading the latest software package, unzip the package into a temporary folder (temporary upgrade directory). This folder MUST be outside the current server installation folder!

If you are running the *boom* Server in a Primary-Backup/Master-Slave environment, please make sure you are upgrading the *boom* Servers in the following order:
1. **Backup/Slave Server**
2. **Primary/Master Server**

Linux/Unix

The upgrade process must be performed **inside the temporary upgrade directory** by a user root or a non-root user that has the ownership of the current server installation directory.
As a first step you have to adapt entry JAVA_BIN="<java_17_repository_path>/bin" (Java 17) in configuration file boom_srv.cfg inside the temporary folder:

```
cd /<unpacked_temp_dir>/server
vi boom_srv.cfg
    JAVA_BIN="<java_17_repository_path>/bin"
```

And then change the permissions of the upgrade script in oder to be able to execute the script by entering the following command: `chmod 750 upgrade.sh`
Run the upgrade script `upgrade.sh` by entering `./upgrade.sh`

After the script has been started it will guide you through the complete server upgrade process. All information that is displayed in square brackets [] will be used by the script as default value if no other data is entered by the user.

For more detailed information about the Linux upgrade process see the following chapter *boom* Server Upgrade Example Unix/Linux.

Upgrade Steps:

- Note: Please check/adapt "JAVA_BIN" entry (Java 17) in configuration file boom_srv.cfg before starting upgrade script.
- The script will first ask for the current server installation directory. Please enter the directory where the *boom* server that you want to upgrade is running.
- If the *boom* Server is running, it should be stopped before the upgrading process will start.
- You have the possibility to backup the current installation.
- updating (optional) rc scripts (Unix only) and the booms_service.exe (Windows only).
- updating (optional) the database.
- updating (optional) the boom_srv startup scripts (boom_srv, boom_srv.cmd).
- updating (optional) main files (main directory with boom_server.jar, install, upgrade, and subdirectories refs, jsi)
- updating (optional) the libraries (subdirectory srv/lib)
- updating (optional) the Oracle and MySQL specific sql scripts (subdirectories ORACLE_SQL and MYSQL_SQL)
- updating (optional) the Unix template file for service/daemon configuration (subdirectory ux-daemon.cfg)
- updating (optional) the SNMP MIB files (subdirectory srv/mibs)
- updating (optional) the HTML files (subdirectory srv/html used for the server webpages like statistics and the dashboards)

- updating (optional) the Agent and UI deployment bundles (subdirectory srv/deployment containing the agent and UI bundles)
- updating (optional) the mandatory binary Packages (boomJavaMonitors, Inventory, SNMP)
- Start the *boom* Server

Windows

The installation needs to be performed by a user that has **administrative rights (i.e. administrator)**. To start the upgrade process on a Windows system, you have to run the upgrade.cmd from a Windows command line interface. Open the cmd.exe console, switch to the directory where the upgrade package is located. Adapt entry JAVA_BIN="<java_17_repository_path>\bin" (Java 17) in configuration file boom_srv.cfg and then run the upgrade script:

```
C:\temp\boom_server> notepad boom_srv.cfg
    JAVA_BIN="<java_17_repository_path>\bin"
C:\temp\boom_server> upgrade.cmd
```

Upgrade Steps:

All steps that will be performed during the upgrade are identically to the Unix upgrade steps as described above.

For more detailed information about the Windows upgrade process see chapter *boom* Server Upgrade Example Windows.

2.4.1.1. *boom* Server Upgrade Example Unix/Linux

Upgrade from Version 5.9.8 to 5.10.0 on Linux

Initial situation: *boom* Server installation directory is "/opt/boom/server"; temporary upgrade directory is "/tmp/server"

```
[root@boomS ~]# cd /tmp/server
[root@boomS server]# cat boom_srv.cfg
JAVA_OPTS="-Xmx1300M -Dfile.encoding=UTF8 -Dsun.net.inetaddr.ttl=300 -Djdk.tls.ephemeralDHKeySize=2048"
JAVA_BIN=/opt/boom/java17/bin

[root@boomS server]# chmod 750 upgrade.sh
[root@boomS server]# ./upgrade.sh
Checking java: /opt/boom/java17/bin/java
/opt/boom/java17/bin/java -version
NAME="Red Hat Enterprise Linux"
VERSION="8.7 (Ootpa)"
ID="rhel"
ID_LIKE="fedora"
VERSION_ID="8.7"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Red Hat Enterprise Linux 8.7 (Ootpa)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:redhat:enterprise_linux:8::baseos"
HOME_URL="https://www.redhat.com/"
DOCUMENTATION_URL="https://access.redhat.com/documentation/red_hat_enterprise_linux/8/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"

REDHAT_BUGZILLA_PRODUCT="Red Hat Enterprise Linux 8"
REDHAT_BUGZILLA_PRODUCT_VERSION=8.7
REDHAT_SUPPORT_PRODUCT="Red Hat Enterprise Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="8.7"
```

```
openjdk version "17.0.9" 2023-10-17 LTS
Java is OK. Major version: 17
OS: Linux
isUnix: true
whoami
Current User: root
stat -c %U /opt/boom/server
User /opt/boom/server: boomerang
stat -c %G /opt/boom/server
Group /opt/boom/server: boomerang
Target boom server installation: /opt/boom/server
/opt/boom/zulu17.46.19-ca-jdk17.0.9-linux_x64/bin/java -jar /opt/boom/server/boom_server.jar -version
boom version: [5.9.8]
Current location: /tmp/server
/opt/boom/zulu17.46.19-ca-jdk17.0.9-linux_x64/bin/java -p /tmp/server/srv/libs -m
boom.server/com.blixx.server.ServerEngine -version
boom version: [5.10.0]

Stop server? (yes|no|cancel)? [Y]
INIT_DIR=/etc/init.d
sh -c "init_s=ps -ef | grep init | grep -v grep | awk '{print $2}' | grep -w \"1\"; echo ${init_s}"
sh -c "init_s=ps -ef | grep systemd | grep -v grep | awk '{print $2}' | grep -w \"1\"; echo ${init_s}"
SYS_INIT: systemd
stopping Server
systemctl stop boom_srv

Backup existing installation? (yes|no|cancel)? [Y]
Backup file:  [/tmp/server/boom_server_bkp.zip]:
chmod 755 boom_srv
chmod -R 755 ux-daemon.cfg
chmod -R 755 srv/packages
chmod 660 boom.props
chmod 660 db.props
chmod 660 db_perf.props
upgrade rc scripts? (yes|no|cancel)? [N]
Runtime OpenJDK: /opt/boom/java17/bin
Set JAVA_BIN in /opt/boom/server/boom_srv.cfg: JAVA_BIN=/opt/boom/java17/bin
Upgrade database? (yes|no|cancel)? [N]

PERF DB properties: /opt/boom/server/db_perf.props

Port=1521
EnPass=AAAACEGTavJYM4+E
Driver=oracle.jdbc.OracleDriver
DbName=BOOM_PERF
Host=jean.bes-intern.com
Login=boom_perf
Password=
SID=openview

Check Only? Y- check only,N - change settings (yes|no|cancel)? [Y]
PERF DB is OK
Update main files? (yes|no|cancel)? [Y]
/opt/boom/server/refs/APACHE-2_0.txt done.
/opt/boom/server/refs/copyrights.txt done.
/opt/boom/server/refs/epl-v10.html done.
/opt/boom/server/refs/lgpl-2.1.txt done.
/opt/boom/server/jsi/upgrade.jsi.log done.
/opt/boom/server/jsi/install.jsi done.
/opt/boom/server/jsi/upgrade.jsi done.
/opt/boom/server/jsi/testJavaVersion.jsi done.
```

```
/opt/boom/server/jsi/installDB.jsi done.
/opt/boom/server/jsi/upgradeFiles.jsi done.
/opt/boom/server/jsi/checkDatabase.jsi done.
/opt/boom/server/jsi/checkDatabasePerf.jsi done.
/opt/boom/server/jsi/installSystemD.jsi done.
/opt/boom/server/jsi/upgradeDB.jsi done.
/opt/boom/server/jsi/startServer.jsi done.
/opt/boom/server/jsi/stopServer.jsi done.
/opt/boom/server/jsi/getRcDir.jsi done.
/opt/boom/server/jsi/installInitRC.jsi done.
/opt/boom/server/jsi/upgradeRC.jsi done.
/opt/boom/server/jsi/getInitSystem.jsi done.
/opt/boom/server/jsi/getInitDir.jsi done.
/opt/boom/server/jsi/patch.jsi done.
found /opt/boom/server/srv/policies/fwd/errTOdnEntw.fwd.xml
found /opt/boom/server/srv/policies/fwd/errTOdtTST.fwd.xml
found /opt/boom/server/srv/policies/fwd/evTOdt.fwd.xml
found /opt/boom/server/srv/policies/fwd/UserFilter_1704899225705.fwd.xml
boom.props file is already upgraded.

Update srv/libs folder? (yes|no|cancel)? [Y]
updating libraries...
/opt/boom/server/srv/libs/activation-1.1.jar done.
/opt/boom/server/srv/libs/boom.server-5.10.0.jar done.
/opt/boom/server/srv/libs/commons-email-1.5.jar done.
/opt/boom/server/srv/libs/gson-2.10.1.jar done.
/opt/boom/server/srv/libs/javax.mail-1.5.6.jar done.
/opt/boom/server/srv/libs/jetty-http-11.0.18.jar done.
/opt/boom/server/srv/libs/jetty-io-11.0.18.jar done.
/opt/boom/server/srv/libs/jetty-jakarta-servlet-api-5.0.2.jar done.
/opt/boom/server/srv/libs/jetty-security-11.0.18.jar done.
/opt/boom/server/srv/libs/jetty-server-11.0.18.jar done.
/opt/boom/server/srv/libs/jetty-servlet-11.0.18.jar done.
/opt/boom/server/srv/libs/jetty-util-11.0.18.jar done.
/opt/boom/server/srv/libs/jfreechart-1.5.4.jar done.
/opt/boom/server/srv/libs/jsch-0.1.55.jar done.
/opt/boom/server/srv/libs/log4j-api-2.21.1.jar done.
/opt/boom/server/srv/libs/log4j-core-2.21.1.jar done.
/opt/boom/server/srv/libs/log4j-slf4j2-impl-2.21.1.jar done.
/opt/boom/server/srv/libs/mysql-connector-j-8.2.0.jar done.
/opt/boom/server/srv/libs/ojdbc10-19.21.0.0.jar done.
/opt/boom/server/srv/libs/protobuf-java-3.21.9.jar done.
/opt/boom/server/srv/libs/slf4j-api-2.0.9.jar done.
/opt/boom/server/srv/libs/boom-jsi.jar done.
updating libraries done.
Update ORACLE_SQL folder? (yes|no|cancel)? [Y]
updating ORACLE_SQL ...
/opt/boom/server/ORACLE_SQL/db.props done.
/opt/boom/server/ORACLE_SQL/step1_create_boom_db.sql done.
/opt/boom/server/ORACLE_SQL/step2_createOutageTables.sql done.
/opt/boom/server/ORACLE_SQL/step3_createPGTables.sql done.
/opt/boom/server/ORACLE_SQL/step4_createAgentExtTables.sql done.
/opt/boom/server/ORACLE_SQL/upgrade_oracledb.sql done.
Update MYSQL_SQL folder? (yes|no|cancel)? [Y]
updating MYSQL_SQL ...
/opt/boom/server/MYSQL_SQL/createDBsAndUsers_sample.sql done.
/opt/boom/server/MYSQL_SQL/db.props done.
/opt/boom/server/MYSQL_SQL/step1_create_boom_db.sql done.
/opt/boom/server/MYSQL_SQL/step2_createOutageTables.sql done.
/opt/boom/server/MYSQL_SQL/step3_createPGTables.sql done.
/opt/boom/server/MYSQL_SQL/step4_createAgentExtTables.sql done.
```

```
/opt/boom/server/MYSQL_SQL/upgrade_mysqldb.sql done.
Update/Replace srv/mibs folder? (yes|no|cancel)? [Y]
/opt/boom/server/srv/mibs/A3COM-SWITCHING-SYSTEMS-MIB done.
...
/opt/boom/server/srv/mibs/VRRP-MIB done.
/opt/boom/server/srv/mibs/WWP-LEOS-BLADE-MIB done.
/opt/boom/server/srv/mibs/WWP-PRODUCTS-MIB done.
/opt/boom/server/srv/mibs/WWP-SMI done.
/opt/boom/server/srv/mibs/WWW-MIB done.
/opt/boom/server/srv/mibs/XYLAN-BASE-MIB done.
Update srv/html folder? (yes|no|cancel)? [Y]
updating html files...
/opt/boom/server/srv/html/acknMsgPie.png done.
...
/opt/boom/server/srv/html/warning_f.png done.
updating html done.
Update srv/deploy folder? (yes|no|cancel)? [Y]
updating deployments...
agent.conf remains the same
updating deployments done.
Update srv/packages/BoomJavaMonitors? (yes|no|cancel)? [Y]
updating BoomJavaMonitors...
updating BoomJavaMonitors done.
Update srv/packages/SNMP? (yes|no|cancel)? [N]
Update srv/packages/Inventory? (yes|no|cancel)? [N]
Update srv/jobs folder? (Config changes will be replaced) (yes|no|cancel)? [N]
Add new jobs? (yes|no|cancel)? [Y]
File /opt/boom/server/srv/jobs/AgentExport.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/AutoArchiveFiltered.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/AutoCloseFiltered.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/Auto_Archive_Duplicates.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/Certificate_expiration_check.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/cleanBoomPerfDB.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/CMDB_Export.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/Find_Lost_Policies.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/importVAgents.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/InventoryClean.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/sampleExec.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/SyncAgents.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/Synchronize_PROXY_Slaves.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/SyncUsers.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/UserExport.job.xml exists. Skipped.
File /opt/boom/server/srv/jobs/vagentsimport exists. Skipped.
done.

updating srv/cli/LinuxUtilsSamples...
/opt/boom/server/srv/cli/LinuxUtilsSamples/boomAGTSTAT done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/boomCMD done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/boomDNLD done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/boomIIDGET done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/boomINDIGET done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/boomINDISEXP done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/boomSRVSTAT done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/boomUPLD done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/boom_env_ssl.cnf done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/boom_prov_enc.sh done.
/opt/boom/server/srv/cli/LinuxUtilsSamples/README.txt done.
creating/updating srv/cli/LinuxUtilsSamples folder done.
Upgrade finished.
If you had HTTPS activated - you need to convert to p12 format and move your keystore to a new location
```

```
i.e. keytool -importkeystore -srckeystore /opt/boom/server/keystore.jks -destkeystore
/opt/boom/server/srv/etc/keystore.p12 -srcstoretype JKS -deststoretype PKCS12 -srcstorepass <storePass>
-deststorepass changeit -srcalias <oldAlias> -destalias server -srckeypass <keyPass> -destkeypass changeit
-noprompt
Please consider to review and import policies, binary packages, etc with boom UI -> Import MPI View.
chown -R boomerang:boomerang /opt/boom/server
/opt/boom/server: change owner boomerang:boomerang recursively done

Start server? (yes|no|cancel)? [Y]
INIT_DIR=/etc/init.d
sh -c "init_s=ps -ef | grep init | grep -v grep | awk '{print $2}' | grep -w \"1\"; echo ${init_s}"
sh -c "init_s=ps -ef | grep systemd | grep -v grep | awk '{print $2}' | grep -w \"1\"; echo ${init_s}"
SYS_INIT: systemd
starting Server
systemctl restart boom_srv
```

2.4.1.2. *boom* Server Upgrade Example Windows

Upgrade from Version 5.9.8 to 5.10.0 on Windows

Initial situation: *boom* Server installation directory is "C:\boom\server"; temporary upgrade directory is "C:\Users\Administrator\Downloads\server\server"

Please open a console window with administrative rights.

```
C:\>cd C:\Users\Administrator\Downloads\server\server

C:\Users\Administrator\Downloads\server\server>type boom_srv.cfg
JAVA_OPTS="-Xmx1300M -Dfile.encoding=UTF8 -Dsun.net.inetaddr.ttl=300 -Djdk.tls.ephemeralDHKeySize=512"
JAVA_BIN=C:\boom\zulu17.48.15-ca-jdk17.0.10-win_x64\bin

C:\Users\Administrator\Downloads\server\server>upgrade.cmd

C:\Users\Administrator\Downloads\server\server>FOR /F "delims=" %L IN ('findstr /v /c:"#" boom_srv.cfg')
DO set %L

C:\Users\Administrator\Downloads\server\server>set JAVA_OPTS="-Xmx1300M -Dfile.encoding=UTF8
-Dsun.net.inetaddr.ttl=300 -Djdk.tls.ephemeralDHKeySize=512"

C:\Users\Administrator\Downloads\server\server>set JAVA_BIN=C:\boom\zulu17.48.15-ca-jdk17.0.10-win_x64\bin

C:\Users\Administrator\Downloads\server\server>C:\boom\zulu17.48.15-ca-jdk17.0.10-win_x64\bin\java.exe -cp
"srv/libs/*" com.blixx.jsi.Run jsi/upgrade.jsi
Checking java: C:\boom\zulu17.48.15-ca-jdk17.0.10-win_x64\bin/java
C:\boom\zulu17.48.15-ca-jdk17.0.10-win_x64\bin/java -version
openjdk version "17.0.10" 2024-01-16 LTS
Java is OK. Major version: 17
Please enter the path to the boom server installation that will be updated: [/opt/boom/server]:
c:\boom\server
OS: Windows Server 2019
OS type: Windows
isUnix: false
Target boom server installation: c:\boom\server
"C:\boom\zulu17.48.15-ca-jdk17.0.10-win_x64\bin\java" -jar "c:\boom\server"/boom_server.jar -version
boom version: [5.9.8]
Current location: C:\Users\Administrator\Downloads\server\server
"C:\boom\zulu17.48.15-ca-jdk17.0.10-win_x64\bin\java" -p
"C:\Users\Administrator\Downloads\server\server"/srv/libs -m boom.server/com.blixx.server.ServerEngine
-version
boom version: [5.10.0]
```

```
Stop server? (yes|no|cancel)? [Y]
net stop BOOMServer
BOOMServer was stopped.

Backup existing installation? (yes|no|cancel)? [Y]
Backup file: [./boom_server_bkp.zip]:
c:\boom\server\booms_service_64.exe -i
Runtime OpenJDK: C:\boom\zulu17.48.15-ca-jdk17.0.10-win_x64\bin
Set JAVA_BIN in c:\boom\server/boom_srv.cfg: JAVA_BIN=C:\boom\zulu17.48.15-ca-jdk17.0.10-win_x64\bin
Upgrade database? (yes|no|cancel)? [N]

PERF DB properties: c:\boom\server/db_perf.props

Port=3306
EnPass=AAAAEOOzLMi0yQbjehT/r0/khZ0=
Driver=com.mysql.jdbc.Driver
DbName=boomav3_boom_perf
Host=boomdb.bes-intern.com
Login=boomav3
Password=

Check Only? Y- check only,N - change settings (yes|no|cancel)? [Y]
PERF DB is OK
Update main files? (yes|no|cancel)? [Y]
C:\boom\server\refs\APACHE-2_0.txt done.
C:\boom\server\refs\copyrights.txt done.
C:\boom\server\refs\epl-v10.html done.
C:\boom\server\refs\lgpl-2.1.txt done.
C:\boom\server\jsi\checkDatabase.jsi done.
C:\boom\server\jsi\checkDatabasePerf.jsi done.
C:\boom\server\jsi\getInitDir.jsi done.
C:\boom\server\jsi\getInitSystem.jsi done.
C:\boom\server\jsi\getRcDir.jsi done.
C:\boom\server\jsi\install.jsi done.
C:\boom\server\jsi\installDB.jsi done.
C:\boom\server\jsi\installInitRC.jsi done.
C:\boom\server\jsi\installSystemD.jsi done.
C:\boom\server\jsi\patch.jsi done.
C:\boom\server\jsi\startServer.jsi done.
C:\boom\server\jsi\stopServer.jsi done.
C:\boom\server\jsi\testJavaVersion.jsi done.
C:\boom\server\jsi\upgrade.jsi done.
C:\boom\server\jsi\upgrade.jsi.log done.
C:\boom\server\jsi\upgradeDB.jsi done.
C:\boom\server\jsi\upgradeFiles.jsi done.
C:\boom\server\jsi\upgradeRC.jsi done.
boom.props file is already upgraded.

Update srv/libs folder? (yes|no|cancel)? [Y]
updating libraries...
C:\boom\server\srv\libs\activation-1.1.jar done.
C:\boom\server\srv\libs\boom-jsi.jar done.
C:\boom\server\srv\libs\boom.server-5.10.0.jar done.
C:\boom\server\srv\libs\commons-email-1.5.jar done.
C:\boom\server\srv\libs\gson-2.10.1.jar done.
C:\boom\server\srv\libs\javax.mail-1.5.6.jar done.
C:\boom\server\srv\libs\jetty-http-11.0.18.jar done.
C:\boom\server\srv\libs\jetty-io-11.0.18.jar done.
C:\boom\server\srv\libs\jetty-jakarta-servlet-api-5.0.2.jar done.
C:\boom\server\srv\libs\jetty-security-11.0.18.jar done.
```

```
C:\boom\server\srv\libs\jetty-server-11.0.18.jar done.
C:\boom\server\srv\libs\jetty-servlet-11.0.18.jar done.
C:\boom\server\srv\libs\jetty-util-11.0.18.jar done.
C:\boom\server\srv\libs\jfreechart-1.5.4.jar done.
C:\boom\server\srv\libs\jsch-0.1.55.jar done.
C:\boom\server\srv\libs\log4j-api-2.21.1.jar done.
C:\boom\server\srv\libs\log4j-core-2.21.1.jar done.
C:\boom\server\srv\libs\log4j-slf4j2-impl-2.21.1.jar done.
C:\boom\server\srv\libs\mysql-connector-j-8.2.0.jar done.
C:\boom\server\srv\libs\ojdbc10-19.21.0.0.jar done.
C:\boom\server\srv\libs\protobuf-java-3.21.9.jar done.
C:\boom\server\srv\libs\slf4j-api-2.0.9.jar done.
updating libraries done.
Update ORACLE_SQL folder? (yes|no|cancel)? [Y]
updating ORACLE_SQL ...
C:\boom\server\ORACLE_SQL\db.props done.
C:\boom\server\ORACLE_SQL\step1_create_boom_db.sql done.
C:\boom\server\ORACLE_SQL\step2_createOutageTables.sql done.
C:\boom\server\ORACLE_SQL\step3_createPGTables.sql done.
C:\boom\server\ORACLE_SQL\step4_createAgentExtTables.sql done.
C:\boom\server\ORACLE_SQL\upgrade_oracledb.sql done.
Update MYSQL_SQL folder? (yes|no|cancel)? [Y]
updating MYSQL_SQL ...
C:\boom\server\MYSQL_SQL\createDBsAndUsers_sample.sql done.
C:\boom\server\MYSQL_SQL\db.props done.
C:\boom\server\MYSQL_SQL\step1_create_boom_db.sql done.
C:\boom\server\MYSQL_SQL\step2_createOutageTables.sql done.
C:\boom\server\MYSQL_SQL\step3_createPGTables.sql done.
C:\boom\server\MYSQL_SQL\step4_createAgentExtTables.sql done.
C:\boom\server\MYSQL_SQL\upgrade_mysqldb.sql done.
Update/Replace srv/mibs folder? (yes|no|cancel)? [Y]
C:\boom\server\srv\mibs\A3COM-SWITCHING-SYSTEMS-MIB done.
...
C:\boom\server\srv\mibs\WWP-SMI done.
C:\boom\server\srv\mibs\WWW-MIB done.
C:\boom\server\srv\mibs\XYLAN-BASE-MIB done.
Update srv/html folder? (yes|no|cancel)? [Y]
updating html files...
C:\boom\server\srv\html\acknMsgPie.png done.
...
C:\boom\server\srv\html\warning.png done.
C:\boom\server\srv\html\warning_f.png done.
updating html done.
Update srv/deploy folder? (yes|no|cancel)? [Y]
updating deployments...
agent.conf remains the same
updating deployments done.
Update srv/packages/BoomJavaMonitors? (yes|no|cancel)? [Y]
updating BoomJavaMonitors...
updating BoomJavaMonitors done.
Update srv/packages/SNMP? (yes|no|cancel)? [N]
Update srv/packages/Inventory? (yes|no|cancel)? [N]
Update srv/jobs folder? (Config changes will be replaced) (yes|no|cancel)? [N]
Add new jobs? (yes|no|cancel)? [Y]
File c:\boom\server\srv\jobs\AgentExport.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\AutoArchiveFiltered.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\AutoCloseFiltered.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\Auto_Archive_Duplicates.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\Certificate_expiration_check.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\cleanBoomPerfDB.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\CMDB_Export.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\Find_Lost_Policies.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\importVAgents.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\InventoryClean.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\sampleExec.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\SyncAgents.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\Synchronize_PROXY_Slaves.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\SyncUsers.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\UserExport.job.xml exists. Skipped.
File c:\boom\server\srv\jobs\vagentsimport exists. Skipped.
done.

updating srv/cli/LinuxUtilsSamples...
C:\boom\server\srv\cli\LinuxUtilsSamples\boomAGTSTAT done.
C:\boom\server\srv\cli\LinuxUtilsSamples\boomCMD done.
C:\boom\server\srv\cli\LinuxUtilsSamples\boomDNLD done.
C:\boom\server\srv\cli\LinuxUtilsSamples\boomIIDGET done.
C:\boom\server\srv\cli\LinuxUtilsSamples\boomINDIGET done.
C:\boom\server\srv\cli\LinuxUtilsSamples\boomINDISEXP done.
C:\boom\server\srv\cli\LinuxUtilsSamples\boomSRVSTAT done.
C:\boom\server\srv\cli\LinuxUtilsSamples\boomUPLD done.
C:\boom\server\srv\cli\LinuxUtilsSamples\boom_env_ssl.cnf done.
C:\boom\server\srv\cli\LinuxUtilsSamples\boom_prov_enc.sh done.
C:\boom\server\srv\cli\LinuxUtilsSamples\README.txt done.
creating/updating srv/cli/LinuxUtilsSamples folder done.
Upgrade finished.
If you had HTTPS activated - you need to convert to p12 format and move your keystore to a new location

i.e. keytool -importkeystore -srckeystore c:\boom\server/keystore.jks -destkeystore
c:\boom\server/srv/etc/keystore.p12 -srcstoretype JKS -deststoretype PKCS12 -srcstorepass <storePass>
-deststorepass changeit -srcalias <oldAlias> -destalias server -srckeypass <keyPass> -destkeypass changeit
-noprompt
Please consider to review and import policies, binary packages, etc with boom UI -> Import MPI View.
Start server? (yes|no|cancel)? [Y] Y
net start BOOMServer
The BOOMServer service is starting.
The BOOMServer service was started successfully.
```

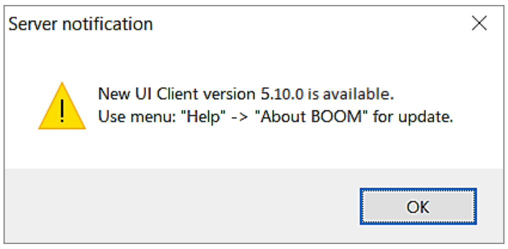
2.4.2. *boom* Client / UI Upgrade

The upgrade of the UI can be performed by using the automatic update functionality or manually.

Please be aware that for upgrades from UI versions below 5.5 the update must be performed by re-installation of the UI.

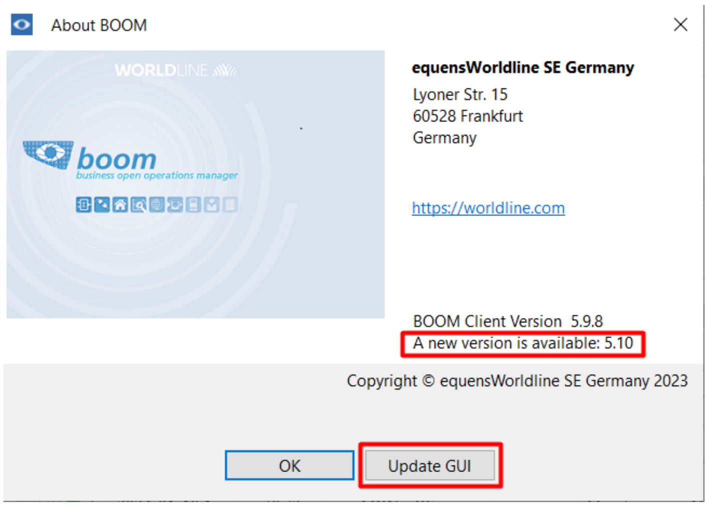
Automatic Update

To use the automatic update, login with your old UI version to the updated *boom* Server.
During startup you get the following information:


Server notification

New UI Client version 5.10.0 is available.
Use menu: "Help" -> "About BOOM" for update.

In order to find out if there is a new version available on the server, open the **About BOOM** dialog in the **Help Menu**.



About BOOM

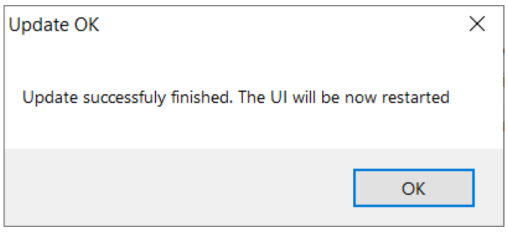
equensWorldline SE Germany
Lyoner Str. 15
60528 Frankfurt
Germany

https://worldline.com

BOOM Client Version 5.9.8
A new version is available: 5.10

Copyright © equensWorldline SE Germany 2023

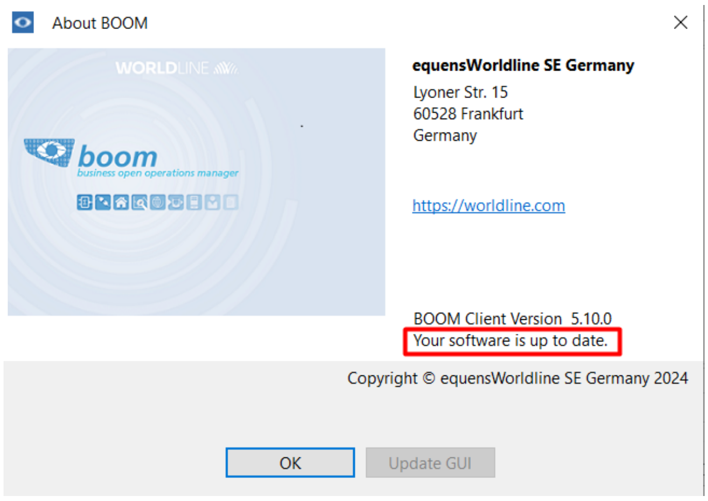
If there is a new version available, select **Update GUI** and the new version will be automatically installed from the *boom* Server. The GUI will be restarted.



Update OK

Update successfuly finished. The UI will be now restarted

After successfully updated the *boom* UI, your "About BOOM" dialog will look like this:



About BOOM

equensWorldline SE Germany
Lyoner Str. 15
60528 Frankfurt
Germany

https://worldline.com

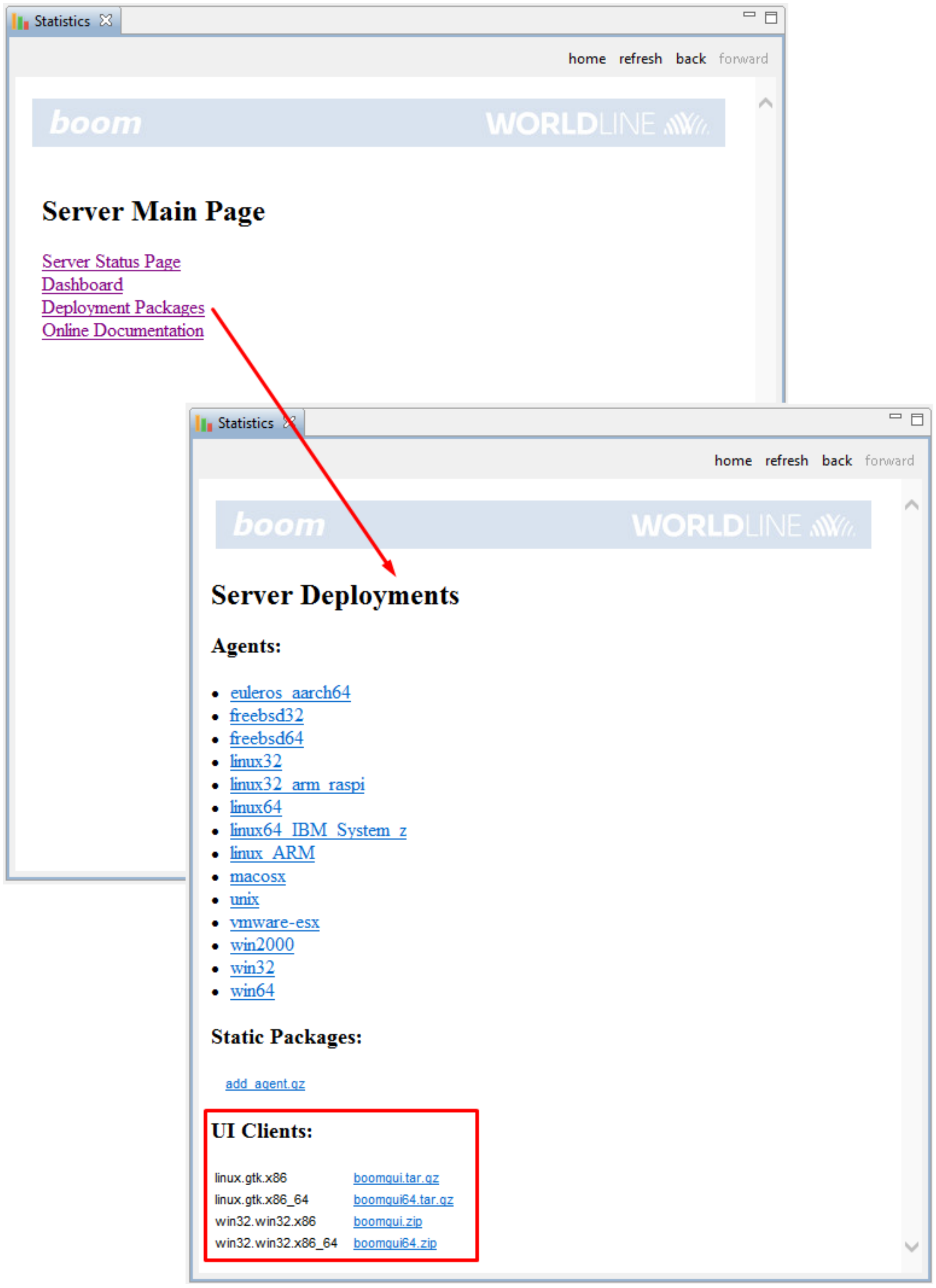
BOOM Client Version 5.10.0
Your software is up to date.

Copyright © equensWorldline SE Germany 2024

Manual Update

To manually update the UI, make sure the UI is stopped. The UI packages are placed on the server and can be downloaded from its built-in web server. To download the UI packages point your browser to the following address:

https://your.boom.server



Statistics

home refresh back forward

boom

Server Main Page

Server Status Page
Dashboard
Deployment Packages
Online Documentation

Statistics

home refresh back forward

boom

Server Deployments

Agents:

- euleros_aarch64
- freebsd32
- freebsd64
- linux32
- linux32_arm_raspi
- linux64
- linux64_IBM_System_z
- linux_ARM
- macosx
- unix
- vmware-esx
- win2000
- win32
- win64

Static Packages:

add_agent.gz

UI Clients:

linux.gtk.x86	boomgui.tar.gz
linux.gtk.x86_64	boomgui64.tar.gz
win32.win32.x86	boomgui.zip
win32.win32.x86_64	boomgui64.zip

Upgrade Steps:

- Download the UI package that suits you platform and unzip the package.
- Replace the old User Interface with the new boomgui package.
- Start the User Interface by executing the boomgui program.

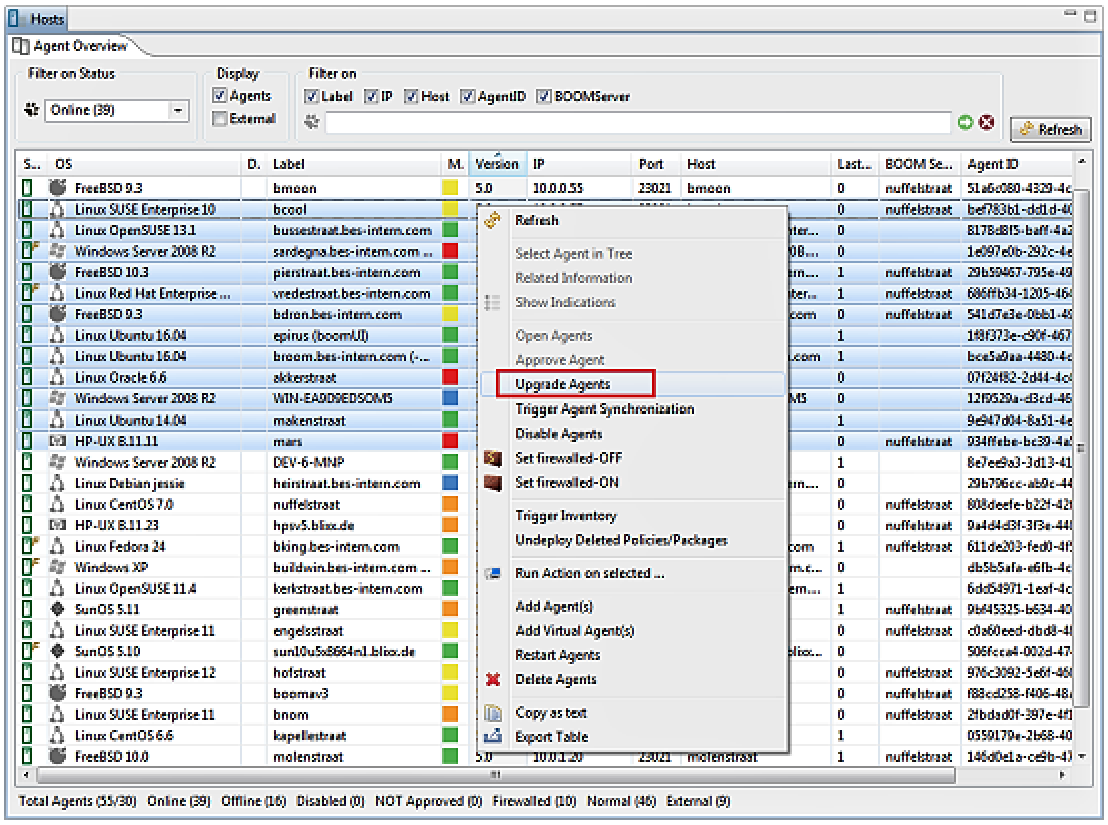
Reusing the old directory allows to keep the locally stored preferences and UI settings. To get a clean installation please remove all directories in the UI installation directory and <profile> directory before placing the new files.

No further steps are necessary. Java needs to be in the current user’s search path in order to startup the GUI.

2.4.3. *boom* Agent Upgrade

The *boom* Agents should only be updated if the upgrade process of the server has been successfully finished. All agent information belonging to the upgrade process is placed inside the server. The agent upgrade can be performed by simple re-deploying the agent from the server via the *boom* user interface.

For new agent installations always use the latest agent packages. These contain support for new system start technologies and service registrations for the *boom* agent.

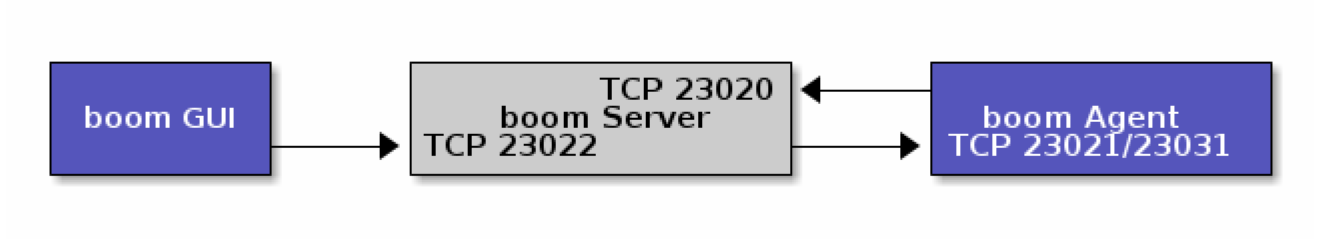


Online (39) — ☑ Agents ☐ External — ☑ Label ☑ IP ☑ Host ☑ AgentID ☑ BOOMServer

S..	OS	D.	Label	M.	Version	IP	Port	Host	Last..	BOOM Se...	Agent ID
	FreeBSD 9.3		bmoon	🟨	5.0	10.0.0.55	23021	bmoon	0	nuffelstraat	51a6c080-4329-4c
	Linux SUSE Enterprise 10		bcool	🟨					0	nuffelstraat	bef783b1-dd1d-40
	Linux OpenSUSE 13.1		bussestraat.bes-intern.com	🟩				ter...	0		8178d815-baff-4a2
F	Windows Server 2008 R2		sardegna.bes-intern.com ...	🟥				08...	0		1e097e0b-292c-4e
	FreeBSD 10.3		pierstraat.bes-intern.com	🟩				ern...	1	nuffelstraat	29b59467-795e-49
F	Linux Red Hat Enterprise ...		vredestraat.bes-intern.com	🟩				ter...	1	nuffelstraat	686ffb34-1205-464
	FreeBSD 9.3		bdron.bes-intern.com	🟨				com	0	nuffelstraat	541d7e3e-0bb1-49
	Linux Ubuntu 16.04		epirus (boomUI)	🟩					1		1f8f373e-c90f-467
	Linux Ubuntu 16.04		broom.bes-intern.com (-...	🟩				.com	1		bce5a0aa-4480-4c
	Linux Oracle 6.6		akkerstraat	🟥					0		07f24f82-2d44-4c
	Windows Server 2008 R2		WIN-EA9D9EDSOMS	🟦				MS	0		12f9529a-d3cd-46
	Linux Ubuntu 14.04		makenstraat	🟩					1		9e947d04-8a51-4e
	HP-UX B.11.11		mars	🟥					0	nuffelstraat	934ffebe-bc39-4a
	Windows Server 2008 R2		DEV-6-MNP	🟩					1		8e7ee9a3-3d13-41
	Linux Debian jessie		heinstraat.bes-intern.com	🟦				ern...	0		29b796cc-ab0c-44
	Linux CentOS 7.0		nuffelstraat	🟧					0	nuffelstraat	808deefe-b22f-42
	HP-UX B.11.23		hpsv5.blixx.de	🟧					0	nuffelstraat	9a4d4d3f-3f3e-44
F	Linux Fedora 24		bking.bes-intern.com	🟩				com	1	nuffelstraat	611de203-fed0-4f
F	Windows XP		buildwin.bes-intern.com ...	🟧				m.c...	0		db5b5afa-e6fb-4c
	Linux OpenSUSE 11.4		kerkstraat.bes-intern.com	🟩				em...	1		6dd54971-1eaf-4c
	SunOS 5.11		greenstraat	🟧					1	nuffelstraat	9b645325-b634-40
	Linux SUSE Enterprise 11		engelsstraat	🟨					0	nuffelstraat	c0a60eed-dbd8-4
F	SunOS 5.10		sun10u5x8664nl.blixx.de	🟩				blixx...	0		506fcca4-002d-47
	Linux SUSE Enterprise 12		hofstraat	🟨					0	nuffelstraat	976c3092-5e6f-46
	FreeBSD 9.3		boomav3	🟨					0	nuffelstraat	f88cd258-f406-48
	Linux SUSE Enterprise 11		bnom	🟧					0	nuffelstraat	2fbdad0f-397e-4f1
	Linux CentOS 6.6		kapellestraat	🟩					1		0559179e-2b68-40
	FreeBSD 10.0		molenstraat	🟩	5.0	10.0.1.20	23021	molenstraat	1	nuffelstraat	146d0e1a-ce9b-47

Total Agents (55/30) Online (39) Offline (16) Disabled (0) NOT Approved (0) Firewalled (10) Normal (46) External (9)

2.5. Firewalled Environment



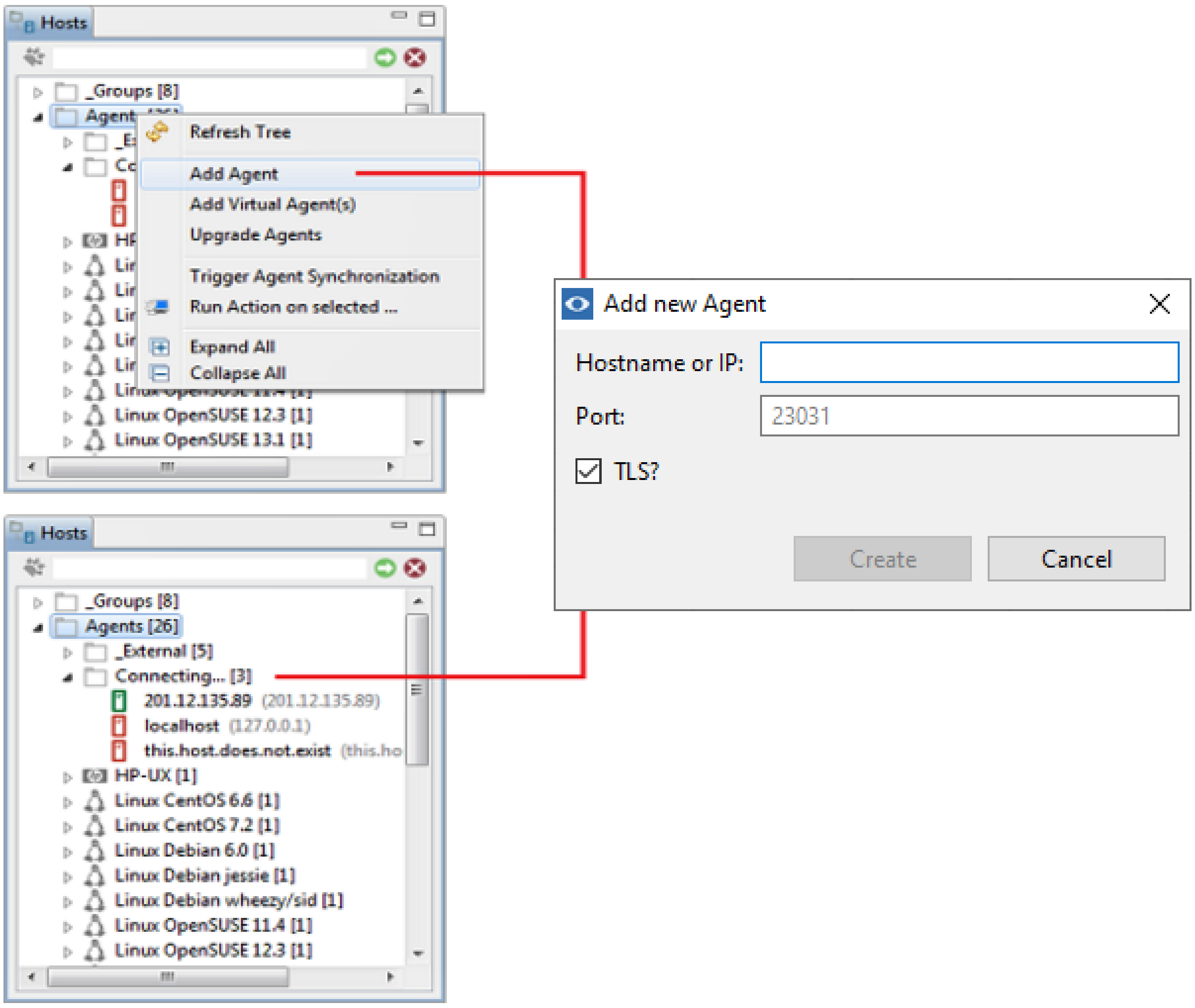
boom will work in environments with Inbound or Outbound only firewall settings for the server - agent connection.

Firewalled *boom* Server (Outbound)

If a firewall blocks the incoming traffic on the server port 23020, the *boom* agents can not actively submit any information or data to the server. The server will switch to a polling mode and fetch the data from the agent as long as agent will be configured with "Mode7" (see also chapter Agent Heartbeats). Mode7 communication means that server opens TCP port 23021 or 23031 (TLS) to agent for polling indication/heartbeat data.

Fresh installed *boom* Agents can not announce their presence to the *boom* server. Therefore the administrator has to add such agents manually in the GUI with the correct IP address. The remote agent must be running. After entering the agent's IP address the *boom* Server will initiate a request to this *boom* Agent to retrieve all necessary information.

If the agent resides on a system with changing IP address (e.g. DHCP) it must be ensured that the agent's address can be resolved by DNS.


Add new Agent

Hostname or IP:

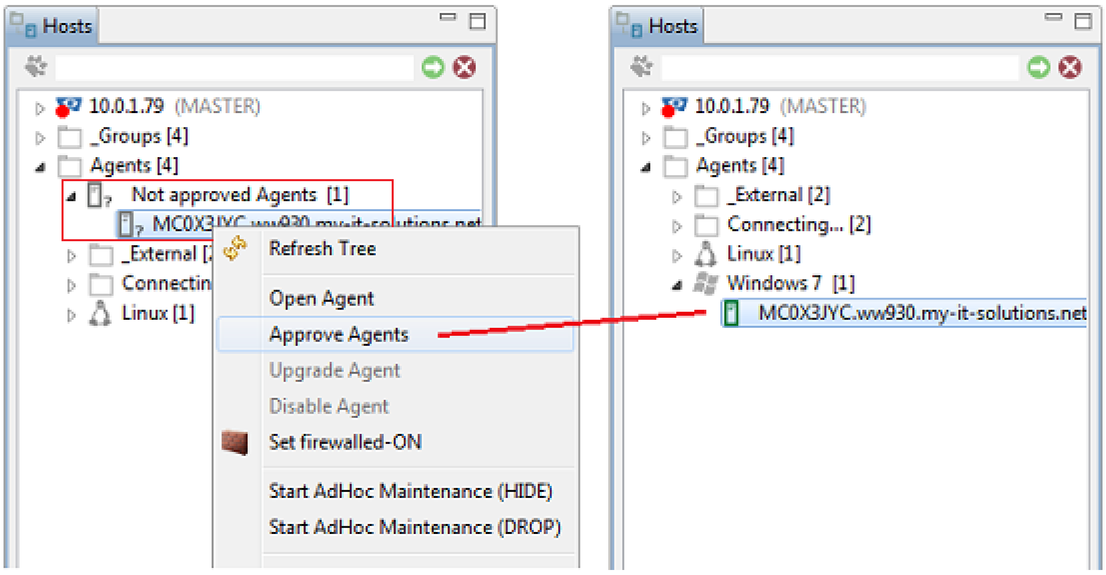
Port: 23031

☑ TLS?

The successfully connected agent will be automatically approved and placed in the appropriate OS type folder with resolved hostname and other attributes.

Firewalled *boom* Agent (Inbound)

If a firewall blocks the incoming traffic on the agent port 23021, the server can not actively send any request to the agent. Every fresh installed *boom* agent will send a request for approval to its configured *boom* server.



After the approval the *boom* Server will try to reach the agent and might need up to 20 seconds to recognize that this is a firewalled agent and change the icon in the 'Hosts View'.
As soon as the *boom* Server marked an agent as firewalled the communication protocol will be switched to a listening mode and the agent will poll data and requests from the server. In this mode the server will continue to try to reach the agent.

To avoid that the server tries to actively establish a connection to the agent, open the 'Agent Details' by double clicking on the agent. Clear the 'Firewalled' checkbox to reset this auto-detected firewalled state and press 'Save'. Then set the 'Firewalled' checkbox and press Save. This sets the agent to the manual firewalled state. No more heartbeats or other kind of connections will be initiated from the *boom* Server.

Remote actions in this firewall mode will have delays up to 5 seconds, before the agent polls them from the server.

2.6. TLS Encryption

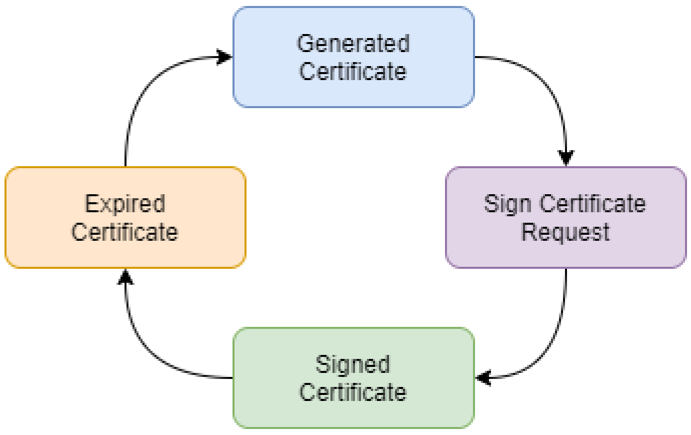
From version 5.9 on Boom does not only provide its own encrypted communication between Server, Agent and UI but does also support SSL/TLS encryption.

General Concept
boom Server supports single port TLS/non-TLS socket communication with Agents and UI. TLS/non-TLS Agents can be used simultaneously.

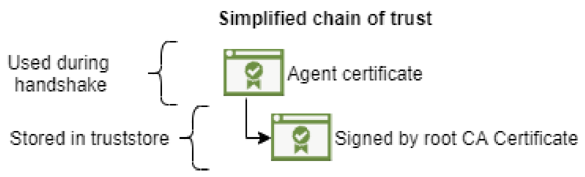
Support of TLS v1.2 is required from the underlying JRE, so ensure using that the latest java 8 version is running on server, agent and UI.

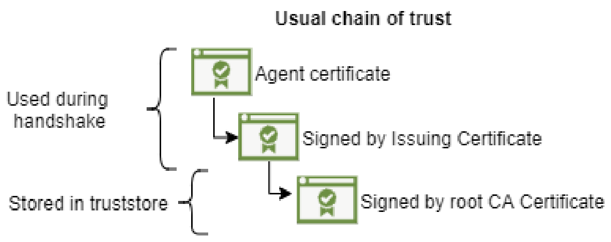
boom provides two modes of certificate handling. By default, self-signing is activated, meaning that a CA is auto-created on the server and used to sign all agent and server certificates.
As alternative, file-based PKI can be activated, to have an external PKI authority integration.

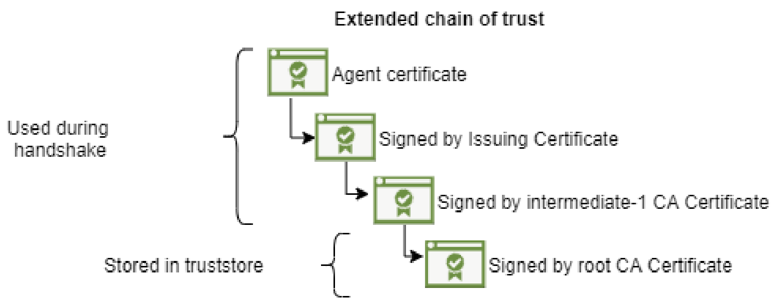
Generally, healthy TLS communication is based on trusted X.509 digital certificates.



Signing is normally overtaken by a trusted third party instance, known as Certificate Authority (CA). Certificates are validated through a chain-of-trust originating from a CA root certificate. With asymmetric cryptography it is possible to use the private key of the CA root certificate to sign other certificates, which can then be validated using only the public key of the root certificate.






Extended chain of trust
Used during handshake
Stored in truststore
Agent certificate
Signed by Issuing Certificate
Signed by intermediate-1 CA Certificate
Signed by root CA Certificate

During handshake, the TLS client will validate the incoming server certificate chain with help of the root CA certificates stored in the client's truststore.

2.6.1. Enabling TLS on the Agent

In order to use TLS encryption for a specific Agent the <BOOM_Agent_Home>/conf/agent.cfg parameters have to be set as in example:

```
...
AGENT_TLS_PORT=23031
COMM_TYPE=TLS
KS_FILE=conf/keystore.p12
KS_PASS=changeit
TS_FILE=conf/truststore.p12
TS_PASS=changeit
```

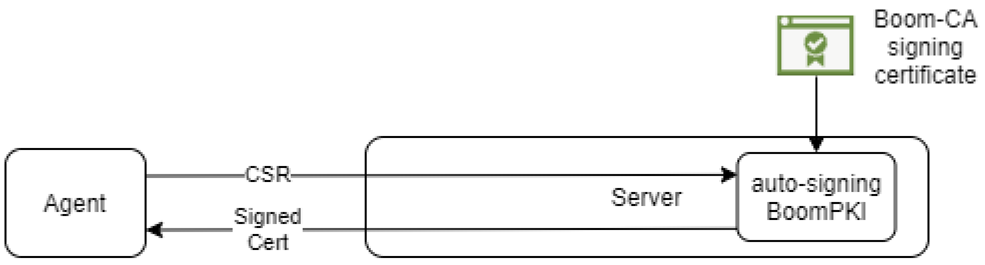
the agent version has to be at least 5.9 in order to use TLS.

The agent package, downloadable from the web page of the boom server, contains the CA root certificate in its conf/truststore.p12 if the version of the Agent is 5.9 or greater. Upon first start of the Agent, a keypair is generated in the Agents keystore. The Agent is going to raise a certificate signing request (CSR) on the Server site. When the user clicks on "Approve Agent", the server starts processing the CSR either by sending the CSR to the PKI authority configured or by self signing the Agents certificates with the CA stored on the server.

After processing, the signed certificate will be automatically transferred to the Agent and activated. On the server side the Agent certificate will be stored for monitoring of the expiration time and additional validation during communication.

2.6.2. Boom PKI

By default, the *boom* Server is configured to use the build-in PKI implementation named "BoomPKI".
The BoomPKI is responsible to sign all incoming CSRs from Agent and Server. The Boom-CA private key and certificate will be automatically generated during first start of the server. It is possible to use the BoomPKI functionality to sign certificates with a CA from an official certificate provider as well.

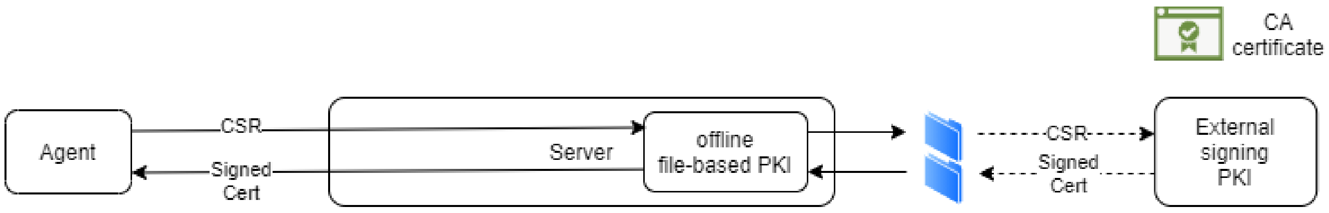


Sample boom_pki.props file:

```
ca_ks_file=srv/etc/cakeystore.p12
ca_distinguished_name=CN\=boomCA, O\=Worldline, C\=DE
ca_password=changeit
```

2.6.3. File Based PKI

"FileBasedPKI" is based on file exchange and allows asynchronous integration with any external PKI authority. The integration of a PKI authority has to be scripted individually on the server. When an Agent or the server raises a certificate signing request (CSR), the server will put the CSRs to the directory specified in file_pki.props (<csr_dir>) using the AgentID as a name. The Agent status will switch from "not approved" to "CSR pending" in the UI. The PKI authority has to be configured to scan this folder, pick up the CSR and delete the request file. When the PKI authority signs a request, it has to place the signed certificate chain in PEM format into the directory specified in file_pki.props (<cert_dir>). The boom server regularly scans this directory for incoming signed certificate chains, processes and pushes them to the corresponding Agent in order to start TLS communication. In the UI the Agent then switches to the "normal" state with a blue TLS sign after its hostname.



2.6.3.1. Setting up a *boom* Server to run with File Based PKI

To run a *boom* Server with file-based PKI some parameters have to be changed prior to the first start of the server. PKI_CLASS in the boom.props has to be changed from "com.blixx.server.pki.BoomPKI" to "com.blixx.server.pki.FileBasedPKI"and PKI_CONFIG_FILE from boom_pki.props to file_pki.props.

```
PKI_CLASS=com.blixx.server.pki.FileBasedPKI
PKI_CONFIG_FILE=file_pki.props
```

Sample file_pki.props file:

```
csr_dir=srv/pki/csr
cert_dir=srv/pki/cert
scan_interval_ms=10000
```

Save your CA root certificate under /srv/pki/cert/ca.pem and start your server for the first time. The server is going to raise a csr inside srv/pki/csr for his own certificate which has to be processed by your pki service. When your pki service provides you with the server.pem save it to srv/pki/cert/ and wait for the server to process the certificate.

2.6.3.2. Switching a Server from Boom PKI to File Based PKI

To switch a boom server from using self-signed certificates to File Based PKI the following steps are necessary to perform:

We assume here that file_pki.props has the default entries.

```
csr_dir=srv/pki/csr
cert_dir=srv/pki/cert
scan_interval_ms=10000
```

1. Stop the *boom* Server
2. Change "PKI" parameters in boom.props to

```
PKI_CLASS=com.blixx.server.pki.FileBasedPKI
PKI_CONFIG_FILE=file_pki.props
```

3. Save the ca.pem (your CA root certificate) to → /opt/boom/server/srv/pki/cert/
4. Start the *boom* Server
5. Get /opt/boom/server/srv/pki/csr/server.csr processed by your pki service.

If you have an intermediate signing certificate, your "server.pem" should contain 2 x certificates (server cert and intermediate signing cert)

6. Save server.pem to "srv/pki/cert/" and wait, until the server processed the incoming certificate.
7. If you have agents already connected with old self-signed certificates - WAIT, until:
   a. New CA will be distributed to the agents
   b. Agents will send new CSRs to the server, that needs to be processed.
8. Wait until agent csr is signed and distributed to all agents
9. Stop *boom* Server
10. Backup /srv/etc/keystore.p12
11. Replace keystore.p12 with /srv/etc/keystore_renewal.p12 which has been created after the server certificate has been signed.
12. Start *boom* Server.

Agent certificate expiration and update
Agents monitor their certificate expiration and will create a new key pair and issue a new certificate signing request before expiration. This certificate signing request will be sent to the Server and processed similarly to the initial CSR processing.

Server certificate expiration and update
In order to update a server certificate one needs to execute the UPDATE_SERVER_CERT boom server action via the Boom UI. This will create a new key pair in keystore_renewal.p12 keystore and issue a certificate signing request, which will be processed by the configured PKI. The signed certificate will be stored in keystore_renewal.p12. In order for the Server to use the new certificate one needs to replace keystore.p12 with keystore_renewal.p12 manually and restart the Boom Server. In case of FileBasedPKI the signed server certificate need to be put into <cert_dir> as server.pem file.

CA certificate update
CA certificate update is a complex procedure involving CA certificate update itself and consecutive Agent certificate and Server certificate updates.

In order to initiate this procedure for *BoomPKI* one needs to remove the cakeystore.p12 on the Server side and restart the Server. This will create a new key pair and certificate for boom server builtin CA and initiate Agent and Server certificate updates.

In case of *FileBasedPKI* one needs to put the new root certificate in PEM format as *ca.pem* file into <cert_dir>. This will also initiate the Agent and Server certificate updates.

2.7. Example: Running *boom* as part of a Pacemaker Cluster

In some environments it might be necessary that applications are available all the time. One method to provide this high-availability of a service is by using a server cluster to run the service. The RedHat HA suite consisting of Pacemaker & Corosync provides such a functionality. Here we show an example configuration of a high-availablilty cluster for BOOM on CentOS 8 systems. Please note that this is only an example, every cluster must be configured individually. Please refer to the official High-availability Cluster Guide for general information on high-availability clusters.

The concept behind a high-availability cluster is that if a server is not available, another server overtakes its services automatically providing a seamless access to the services.

2.7.1. Requirements

- 2 Servers with CentOS 8 installed
- 3 file systems which are mountable for both servers

Firewall and SELinux have to be adjusted or turned off in order to work. These adjustments are not part of this example.

2.7.2. Folder structure

On both servers:

```
mkdir -p /opt/boom/server/srv
mkdir -p /opt/boom/mysql
mkdir -p /opt/boom/mysqlperf
```

2.7.3. MySQL Installation and Configuration

As a database server Percona is used in this example.

```
rpm -ivh Percona-Server-server-57-5.7.31-34.1.el8.x86_64.rpm Percona-Server-client-57-5.7.31-
34.1.el8.x86_64.rpm Percona-Server-shared-57-5.7.31-34.1.el8.x86_64.rpm Percona-Server-shared-compat-57-
5.7.31-34.1.el8.x86_64.rpm
```

The configuration file is modified in a way that 2 individual instances of MySQL are running.

```
vi /etc/percona-server.conf.d/mysqld.cnf
```

mysqld.cnf

```
[mysqld@mysql_boom]
datadir=/opt/boom/mysql
socket=/opt/boom/mysql/mysql.sock
log-error=/var/log/mysqld_boom.log
pid-file=/var/run/mysqld/mysqld_boom.pid
```

```
[client]
port=3306
socket=/opt/boom/mysql/mysql.sock
```

```
[mysqld@mysql_boom_perf]
datadir=/opt/boom/mysqlperf
socket=/opt/boom/mysqlperf/mysql.sock
port=3307
log-error=/var/log/mysqld_boomperf.log
pid-file=/var/run/mysqld/mysqld_boomperf.pid
```

Mount the file systems on one server for each of the three created directories. Than start the SQL Servers on this server.

```
systemctl start mysqld@mysql_boom
systemctl start mysqld@mysql_boom_perf
```

Do not enable the services as process start-up management should be done by the cluster later.

```
grep 'temporary password' /var/log/mysqld_boom.log
grep 'temporary password' /var/log/mysqld_boomperf.log
```

```
mysql -u root -S /opt/boom/mysql/mysql.sock -p
```

```
Alter user 'root'@'localhost' identified by 'secret'; or SET Password for 'root'@'localhost' =
PASSWORD('secret');
grant all on *.* to 'root'@'%' identified by 'secret';
grant all on *.* to 'root'@'localhost' identified by 'secret';
grant all on *.* to 'root'@'127.0.0.1' identified by 'secret';
flush privileges;
exit
```

```
mysql -u root -S /opt/boom/mysqlperf/mysql.sock -p
```

```
Alter user 'root'@'localhost' identified by 'secret';
grant all on *.* to 'root'@'%' identified by 'secret';
grant all on *.* to 'root'@'localhost' identified by 'secret';
grant all on *.* to 'root'@'127.0.0.1' identified by 'secret';
flush privileges;
exit
```

```
mysql -u root -S /opt/boom/mysql/mysql.sock -p
```

```
create database BOOM;
grant all on BOOM.* to 'boom'@'%' identified by 'secret';
flush privileges;
exit
```

```
mysql -u root -S /opt/boom/mysqlperf/mysql.sock -p
create database BOOM_PERF;
grant all on BOOM_PERF.* to 'boom_perf'@'%' identified by 'secret';
flush privileges;
exit
```

2.7.4. *boom* Installation

boom must be installed on both nodes of a cluster. /server/srv needs to be deleted after installation on the second node. The home folder of BOOM is outside the moveable filesystem and resides on each server individually.

```
cd /opt
unzip OpenJDK8U-jdk_x64_linux_hotspot_8u265b01.tar.gz
cp -s /opt/jdk8u265-b01/bin/java /usr/bin/
cd /opt/boom/server
unzip /tmp/BOOM-Install/boom_server_vX.zip
rsync -ar /opt/boom/server/boom_srv/* /opt/boom/server
rm -rf /opt/boom/server/boom_srv/
chmod 750 install.sh
./install.sh
```

instead of localhost the virtual IPs of the databases have to be entered.

2.7.5. Cluster Installation

on both servers:

```
dnf config-manager --set-enabled HighAvailability
dnf install -y pcs fence-agents-all pcp-zeroconf
passwd hacluster
    hacluster
```

on first server:

```
systemctl start pcsd
systemctl enable pcsd
```

from the second server:

```
pcs host auth <first server>
pcs cluster setup boom_cluster --start <second server>
pcs cluster start --all
pcs cluster enable --all
pcs property set stonith-enabled=false
```

Disabling Stonith is only recommended in test environments. For production please refer to the the official High-availability Cluster Guide

on first server:

```
pcs resource create mysql_fs Filesystem device="/dev/mapper/abc1" directory="/opt/boom/mysql/"
fstype="xfs" --group mysql
pcs resource create mysqlperf_fs Filesystem device="/dev/mapper/abc2" directory="/opt/boom/mysqlperf/"
fstype="xfs" --group mysql
pcs resource create boom_fs Filesystem device="/dev/mapper/abc3" directory="/opt/boom/server/srv"
fstype="xfs" --group boom
pcs resource create mysql_srv systemd:mysqld@mysql_boom.service --group mysql
pcs resource create mysqlperf_srv systemd:mysqld@mysql_boom_perf.service --group mysql
pcs resource create mysql_vip IPaddr2 ip=<IP_adressMySQL> cidr_netmask=26 --group mysql nic=eth1
pcs resource create boom_srv systemd:boom_srv --group boom
pcs resource create boom_vip IPaddr2 ip=<IP_adressBOOM> cidr_netmask=26 nic=eth1 --group boom
pcs resource update mysql_vip op monitor interval=30s timeout=40s
pcs resource update boom_vip op monitor interval=30s timeout=40s
```

from second node:

```
dnf config-manager --set-enabled HighAvailability
dnf install -y pcs fence-agents-all pcp-zeroconf
passwd hacluster
     hacluster
systemctl start pcsd
systemctl enable pcsd
```

from first node:

```
pcs host auth <second server> +
pcs cluster node add <second server>
```

from second node:

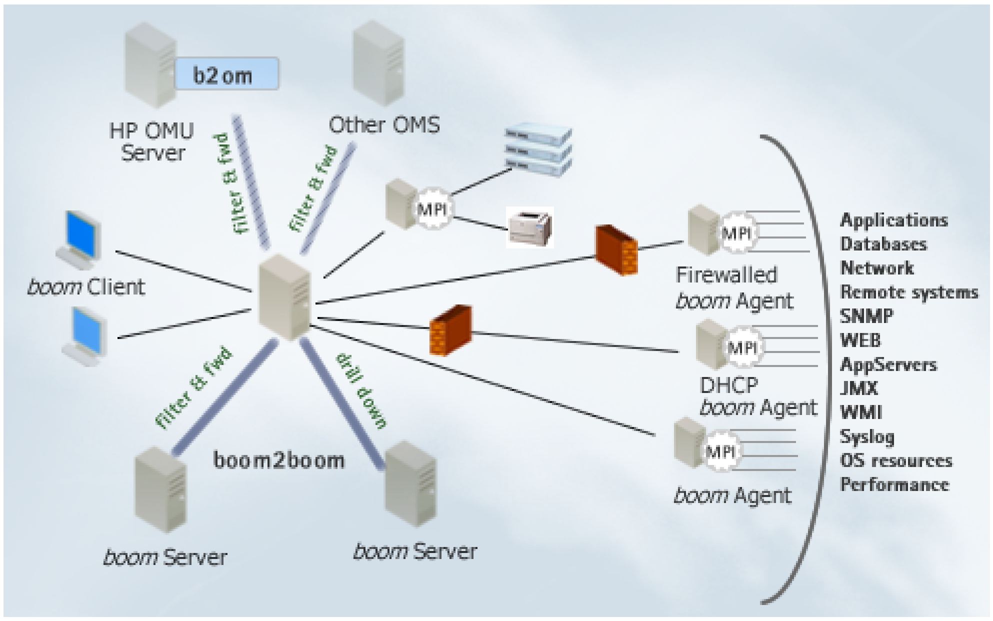
```
pcs cluster start
pcs cluster enable
pcs cluster standby <first node>
```

See if all services are migrated to the second node. If yes remove the standby mode of the first node and the cluster is done.

Chapter 3. Architecture

3.1. Architecture Overview

boom is a distributed client-server monitoring solution that helps IT departments to detect and solve problems on the managed nodes and hosts. The agent based architecture allows minimizing network traffic and enables to manage systems in distributed and firewalled environments.



Key features

- Runs on a wide range of platforms (Windows, Linux, HP-UX, SunOS, MacOS X, VMWare ESX and others).
- Works with firewalled hosts (plain, IN and OUT modes) and DHCP systems.
- Role based configurations for Users and Systems.
- Full featured stand-alone User Interface Client.
- Web UI for Service and Operator Views for SmartPhone, Tablet and PC.
- Integrated Hotspot and Monitor History Graphs.
- Remote actions and automatic actions.
- Full featured policy management with revision, version comparison and merging.
- Automated configuration checks and synchronization.
- Powerful correlation and suppression engine.
- Hot deployable Java monitors and actions.
- Central performance data collection and history data view.
- Remote deployments of packages and policies.
- Centralized deployment inventory.
- High performance encrypted communication layer.
- Includes performance graphing engine.
- Open interfaces for integration of self developed monitors and element managers.
- Even default system tools can be used as monitors without scripting by using OPM.
- Hierarchical *boom* servers (proxy, full-control, read-only).
- On-fly agent re-assignments.

- Maintenance Window management.
- Integration with Change Management, Workflow and Ticketing Tools.
- State-based or exception-based type of message view possible.
- Availability and KPI metrics.
- Investment protection and easy migration path.

If you have already developed your own monitoring functionality based on Nagios® or HP Operations Manager for Unix, there is no need to start all over again. You can easily reuse or convert these for the use with *boom*.

Features		
Agent-based monitoring	Yes	main feature
Agent-less monitoring	Yes	ssh, smtp, snmp, jmx, wmi, http, https, etc
SYSLOG listener	Yes	
SNMP Trapd	Yes	v1,v2,v2c,v3
SNMP Walk	Yes	v1,v2,v2c,v3
Logfile monitoring (single&multi-line)	Yes	
Logfile monitoring (transactions)	Yes	
External monitors	Yes	
Threshold monitors	Yes	
VMWare ESX monitors	Yes	
Database monitors	Yes	MySQL, Oracle
JMX monitoring	Yes	
OS monitoring	Yes	HPUX, Linux, MacOS/X, Sun, VMWare ESX, Windows, …
Nagios plugins	Supported	
OVO plugins	Supported	

3.2. *boom* Primary / Backup Concept (Single-Tier)

General Information:

The *boom* server is developed as a modular, primary-backup relation. On one level 2 boom server coexist assuming their roles of Primary and Backup server. The Primary server will push Indications, Agents and configuration to the Backup server. Backup server acts as a hot standby server to overtake indication processing in case Primary server is offline. Agent needs to be configured [BACKUP_SERVER_*] to allow Backup server communication. The Primary server uses a **boom user** to log on Backup server. When the Backup server has been verified, Primary data will be transferred.

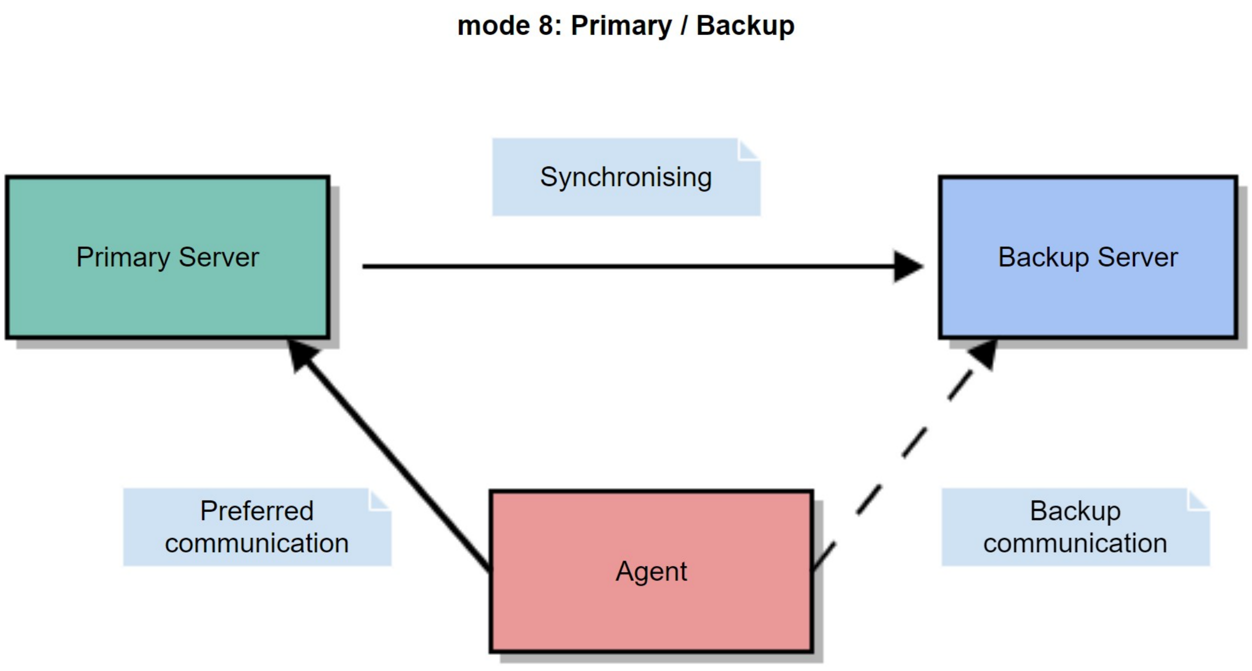
From version 5.9.5 on boom provides the high-availability mode described in the following.

3.2.1. *boom* Primary-Backup Server Scenario

The *boom* server operates in the mode below-mentioned.

Primary-Backup (Mode 8):

Agents are able to switch their communication to a second boom server in case the first one is not available anymore or shutdown. The Primary server synchronizes all necessary data to its Backup server. This enables the Backup server at all time to overtake the Primaries workload also known as "hot-standby" mode.



Agent is switching to the Backup server after more than 3 consecutive failed heartbeats to the Primary server.

3.2.2. Synchronization Primary / Backup

Master-Slave Synchronization for Primary / Backup (Mode 8)

The Primary / Backup concept is designed to provide redundancy of productive servers. This is a "Hot-Standby" configuration which can be used, when other HA (High Availability) implementations are not possible due to virtualization or other reasons. The goal of such a setup is to keep the Backup server ready to overtake the processing quickly. In case of an outage of the Primary server a quick switch reduces service outage to a minimum. We recommend taking general consideration of High Availability into account when setting such an environment up.

Automatically synchronized boom entities:

- Agents
- Indications
- Policies
- Binary packages
- Assignment groups
- Assignments
- Scheduled Maintenances
- Server Policies

- Actions

Not synchronized:

- User and User Groups
- Server Jobs
- Server Configuration (e.g. srv/etc/hosts)
- Notification configs
- Server Filters
- Annotations

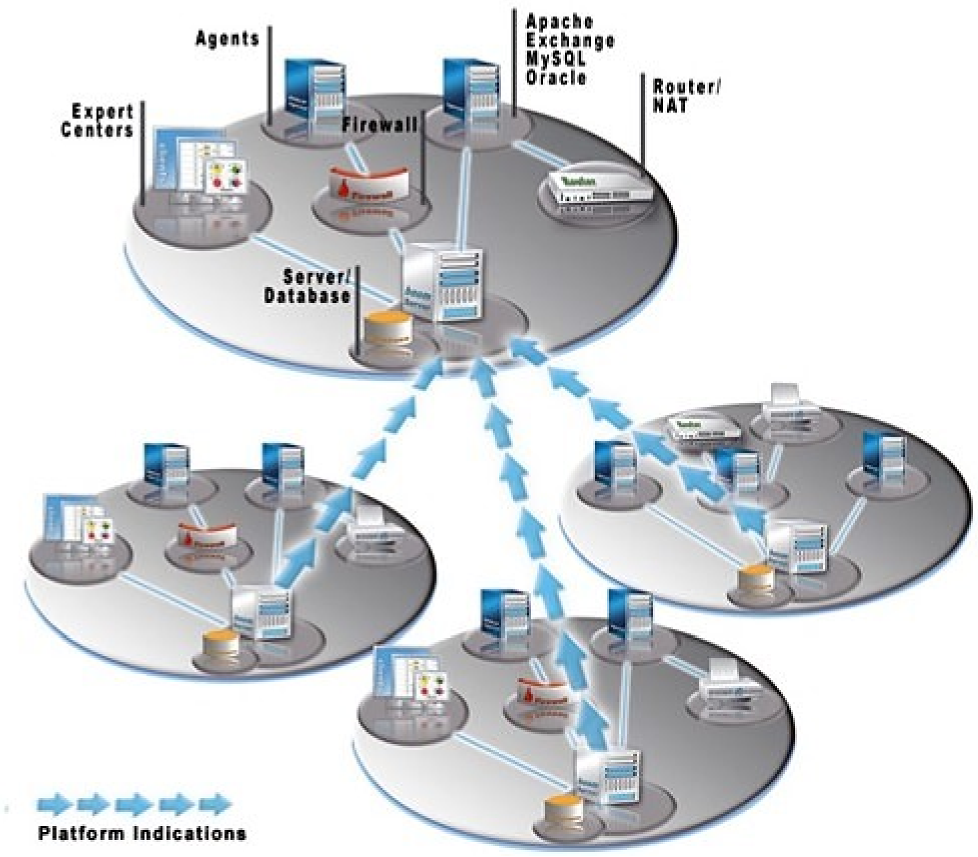
Agents
Agents and Node Groups will be synchronized from the Primary to the Backup automatically. This also includes Agent’s outages and Agent’s statuses like Enabled/Disabled or Firewalled/Not-Firewalled. The Backup server will assume incoming Agents as "local" and will start to communicate with them.

Indications
The incoming messages (Indications) on the Primary server are routed directly to the Backup server, but not vice-versa! On both servers, all messages are archived according to a time schedule.

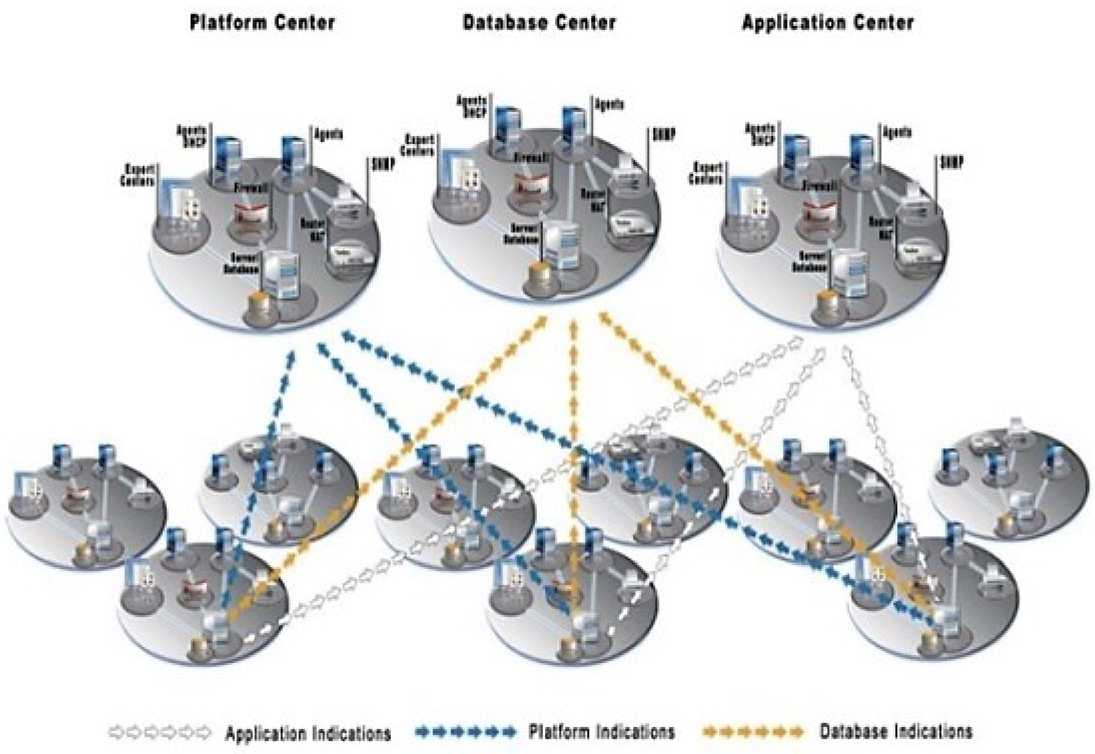
3.3. *boom* Master / Slave Concept (Multi-Tier)

General Information:

The *boom* server is developed as a modular, hierarchical, master-slave relation. The Master server uses a ***boom* user** to log on to a slave server. After the slave servers have been verified, indications will be forwarded depending on the defined server – mode and any filter rules that might be in place for the used *boom* user.



Filtering and forwarding enable a variety of concepts: e.g. The support of distributed environments, departmental servers, time / shift based IT centers, application-specific servers, and more. From version 5.9.5 on *boom* does also support a high-availability mode.

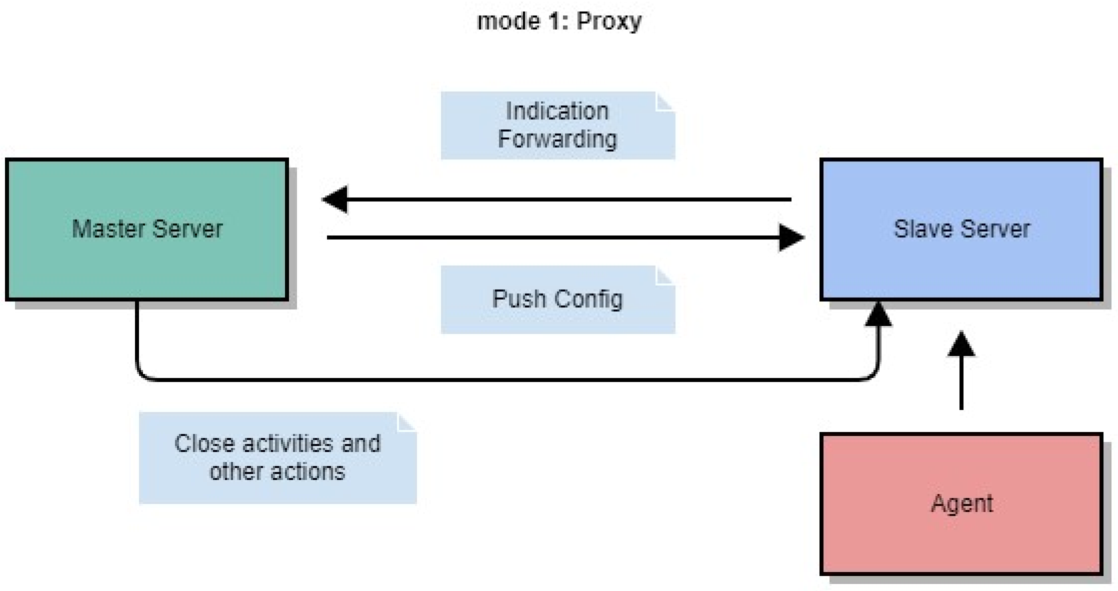


3.3.1. *boom* Master-Slave Server Scenarios

The *boom* server can operate in different modes:

Proxy Slave (Mode 1):

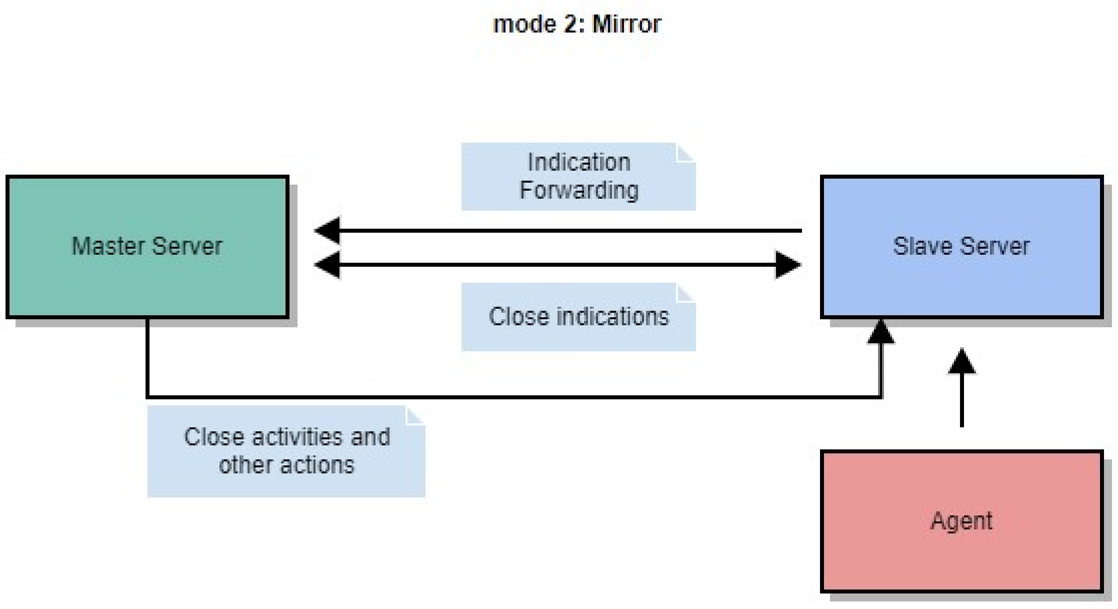
This scenario is always used when the indications have to be managed on one central *boom* master server. The incoming messages on the slave server (Indications) are routed directly to the master server. On the slave server, all messages are archived according to a time schedule.



Mirror (Mode 2):

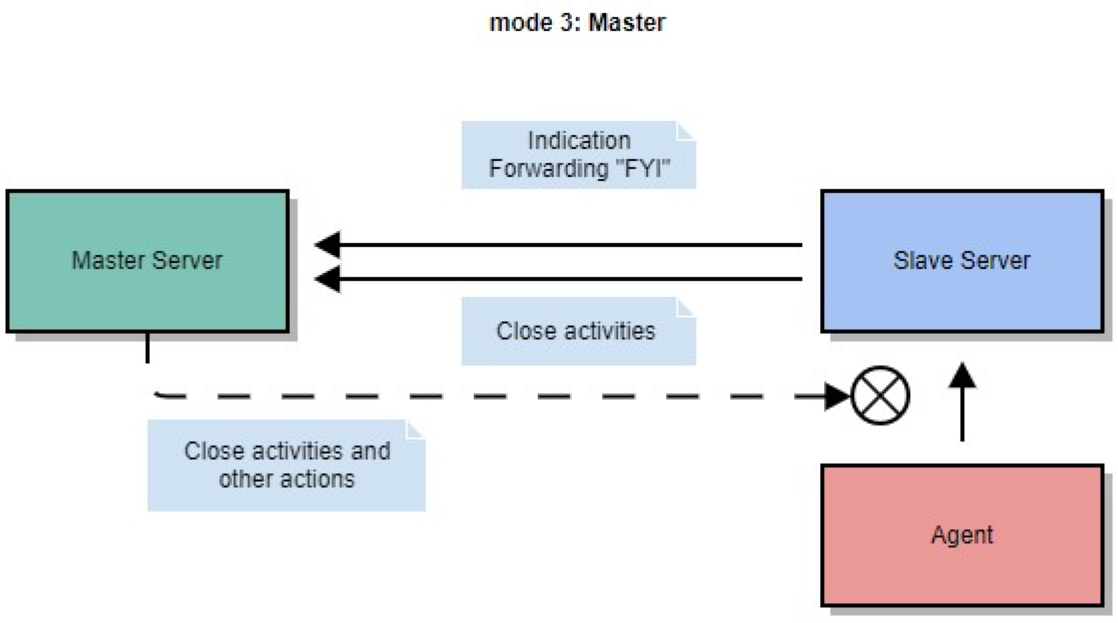
The Indications are forwarded with control to the Master Server. Close activities are synchronized between Master and Slave Server. This means that the messages exist on both servers and for example the Close of an Indication on the master, triggers a close on the slave server.

Activities such as close will be synchronized between the Slave (transmitter) and Master (receiver) server. Furthermore any server action such as "Deploy" is possible from Master to Slave.



Read-Only (Mode 3):

Indications are forwarded to the Master (receiver) as "FYI" (**F**or **Y**our **I**nformation), control remains at the Slave (transmitter). Activities e.g. a close will only be forwarded by the Slave to the Master. Activities on the Master will not be forwarded to the Slave.


mode 3: Master
Indication Forwarding "FYI"
Master Server
Slave Server
Close activities
Close activities and other actions
Agent

3.3.2. Synchronization Master-Proxy server

Master-Slave Synchronization for Proxymode (Mode 1)

The server job **Synchronize_Proxy_Slaves** automatically synchronizes binary packages, policies and assignment groups on PROXY *boom* servers if PROXY slaves are connected to this master server. Per default this job is scheduled every 15 min.

Every modification of policies and assignment groups is directly synchronised to the slaves. Each modification of packages via the user interface like file change, create file, delete file, create folder etc. is forwarded to the slaves immediately. Package modifications which were made on the file system (<boom_dir>/srv/packages) itself are synchronised either on demand (Push Package to all Proxy Server) or with help of the Synchronize_PROXY_SLAVES job. The schedule time for the job is configurable.

Every Deploy, Re-Deploy and Undeploy action initiated from the *boom* Master Server is cached internally. That means when the *boom* Slave Server is offline while acting the deployments, the Master Server provides all the lost deployment jobs in his deployment queue as soon as the slave is online again.

After that all the "queued" deployments at Master site can be inspected via "Assignments Summary Tab – Deployment Queue" and they have to be triggered /cancelled manually if necessary.

Packages
Before Packages can be deployed successfully to the agent they have to be available at the Slave site.

Packages are transferred from Master to Slave either on demand by starting Packages functionality "Push Packages to all Proxy Servers" manually or they are transferred automatically by Server Job "Synchronize_PROXY_Slaves" every 15 minute interval.

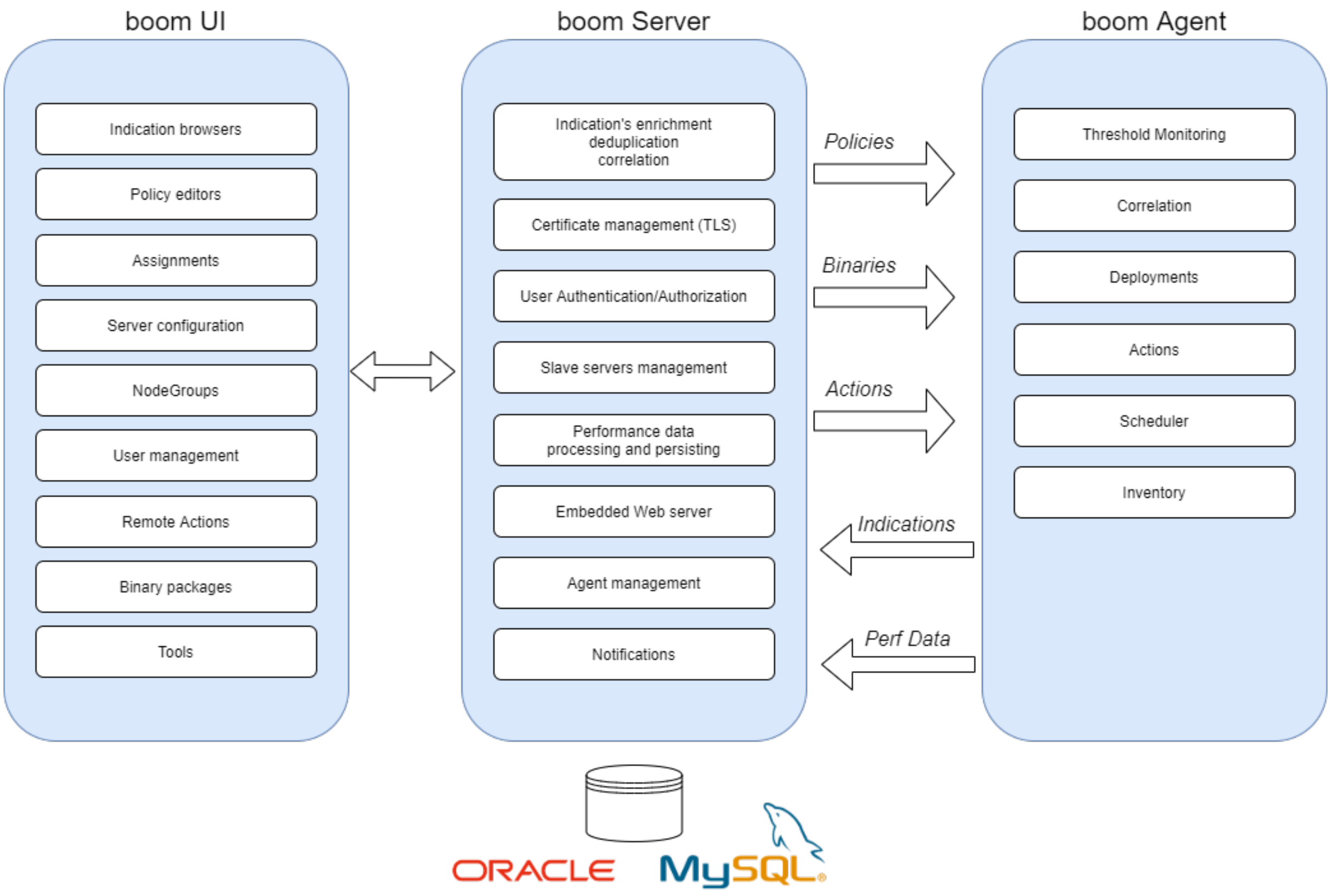
Policies
Any change of policies will be synchronized from the master to the slaves automatically.

Indications
The incoming messages (Indications) on the slave server are routed directly to the master server. On the source-server (slave), all messages are archived according to a time schedule.

Chapter 4. Concept

4.1. Concept Overview

The *boom* management solution consists of 3 main parts: *boom* agent, *boom* server and the *boom* GUI. The core server application requires a MySQL or Oracle database. All components can co-exist on one physical system or can be installed separately.



The *boom* agent is the main component that is responsible for performing monitoring tasks by using a combination of binary probes, scripts or JAVA Monitors with policies that define monitor parameters and threshold levels. In addition 'Filter Policies' allow filtering and creation of event based Indications submitted by external application and Logfile Monitors. And finally Remote Actions give administrators and operators the possibility to perform corrective actions or discover the detailed system status even on firewalled systems. Each agent can take the role of a collection station and perform remote monitoring tasks for agent less hosts (e.g. using SSH, SNMP, etc).

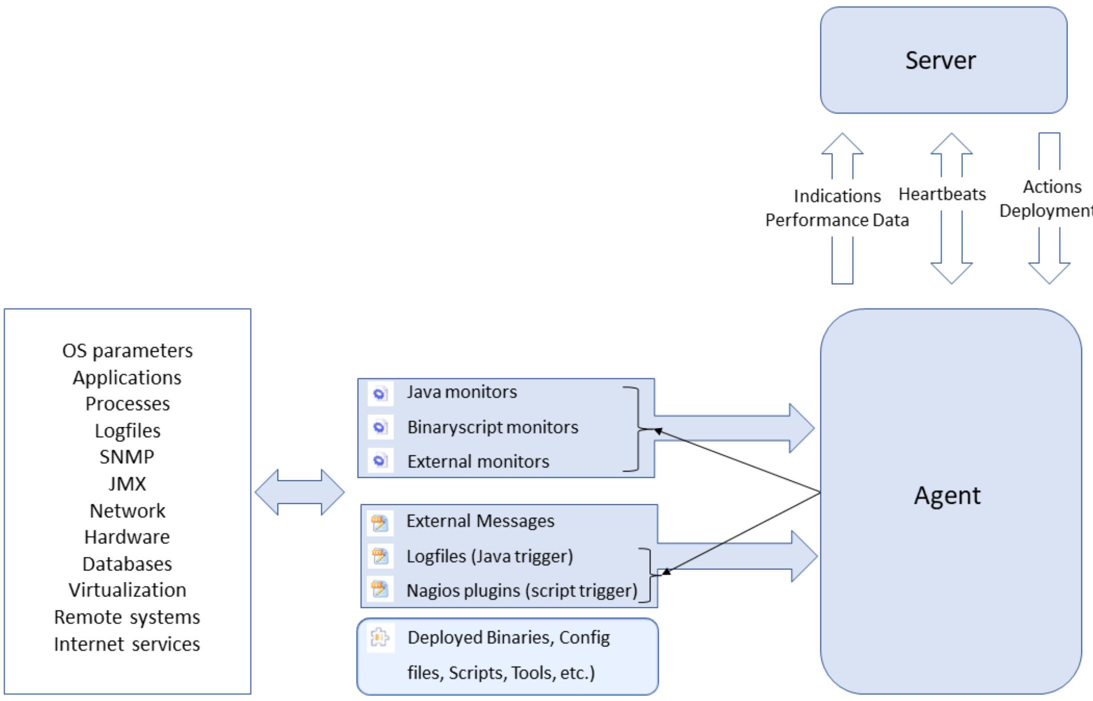
The *boom* server application is the core component that collects and stores all Indications, makes a second stage correlation, manage all sort of configurations and interacts with GUI clients and agents.

The *boom* GUI is not only an Indication browser but also a very powerful control and summarization console. A big number of helper functions and tools allow the administrators and operators to get a clear status overview as well as simplified configuration tasks.

The agent based architecture together with the full featured GUI Client allows the user to get the optimal load balancing of systems, reduced network traffic and improved operator and administrator productivity.

4.1.1. *boom* Agent

The *boom* agent is a distributed part of the *boom* solution, that needs to be installed on the managed node. The agent receives the information what should be monitored and how this is monitored (packages and policies) from the server. After this step the agent is performing the monitoring tasks autonomously. It is responsible to perform monitor calls, collect information, calculations, threshold evaluation, filtering of data and first level of de-duplication and suppression of unnecessary data. Only necessary information like state changes and selected performance data is delivered to the central *boom* server. Furthermore the *boom* agent can perform remote actions (triggered from the *boom* server) and forward performance data.



The *boom* agent can be installed on any system that has a Java Runtime Environment (jre) 1.5 or later installed. *boom* supports such old java versions to allow management of out-dated systems and platforms as they are still active and playing an important role in various customer solutions. Whenever possible an actual version of the jre should be used.

At first start the agent generates an ID which will be send together with an approval request to the configured *boom* server. Only after the approval action (or auto-approval) the agent is able to communicate with the central server.

If the *boom* agent can't access the central server because of a firewall rule, the manual action "Add a new Agent" can be used to add such host to the *boom* server. Generally, the communications between agent and server is possible if at least one component can establish a connection with another. The *boom* agent can be used in firewalled or NAT environments without reducing functionality.

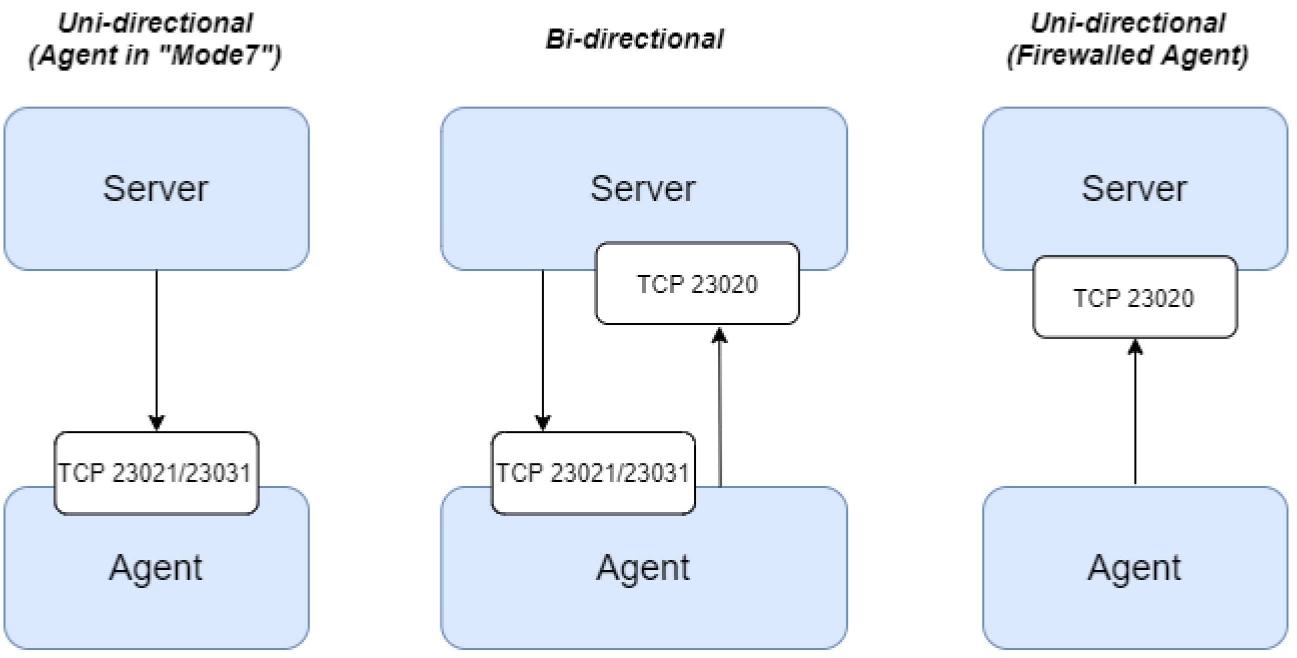
After installation and registration of the agent, usually, one of the first actions is to deploy a set of policies and binaries. On the server side, policies and binaries are grouped in logical groups called "Assignments" that are typically used to manage sets of monitoring configurations for certain system roles and are deployed to the agents.

All monitoring, calculation and filtering made by the *boom* agent are defined in deployed policies. A policy is a sort of configuration file that informs the agent what has to be monitored, when, how and what kind of results should be delivered to the central *boom* server. A single message that needs to be delivered to the central *boom* server is named "Indication".

4.1.2. *boom* Server

The *boom* server is the central manager of all indications that are coming from the managed nodes. The server is responsible for the final correlations, de-duplication and the filtering of incoming Indications. Furthermore the *boom* server manages all sort of configurations and interacts with all GUI clients and agents.

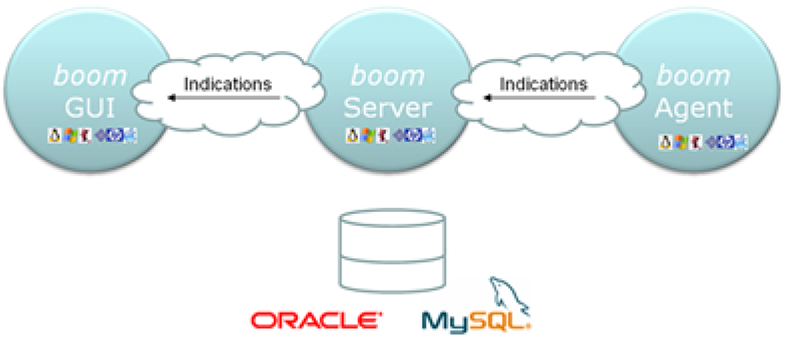
4.2. Communication matrix



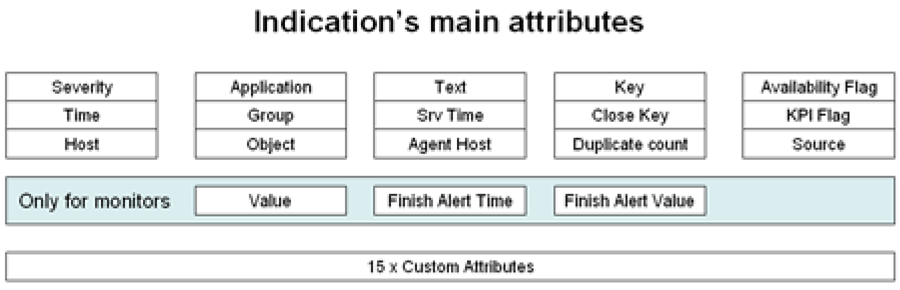
Component	Direction	Port	Description
Server	Incoming	TCP/23020	Processing incoming agent connections
Server	Incoming	TCP/23022	UI and Server to Server communication
Server	Incoming	TCP/443	Embedded web server
Agent	Incoming	TCP/23021	Server to Agent communication and local commands
Agent TLS	Incoming	TCP/23031	Server to Agent communication using TLS
Agent	Incoming	UDP/162	SNMP traps receiver (optional)

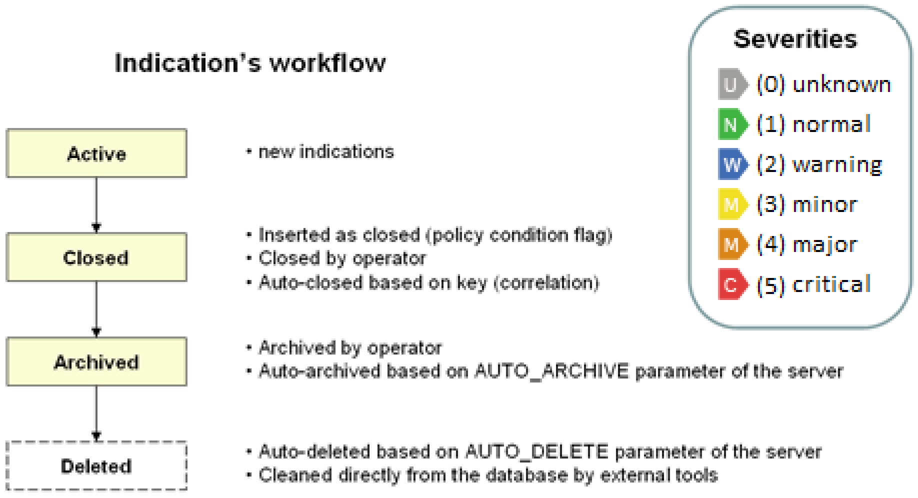
4.3. Indications

All Indications coming from the *boom* agents will be processed by central *boom* server and delivered to an Operator UI.



Each Indication has a severity, text, and other attributes that helps to cover various complex processing or integration scenarios.

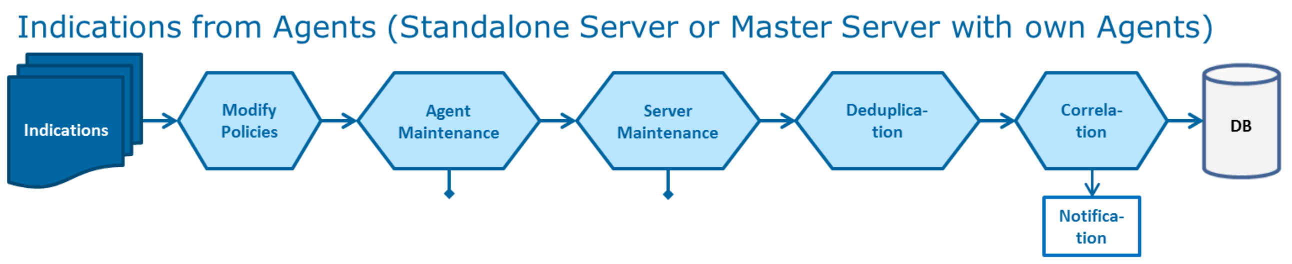


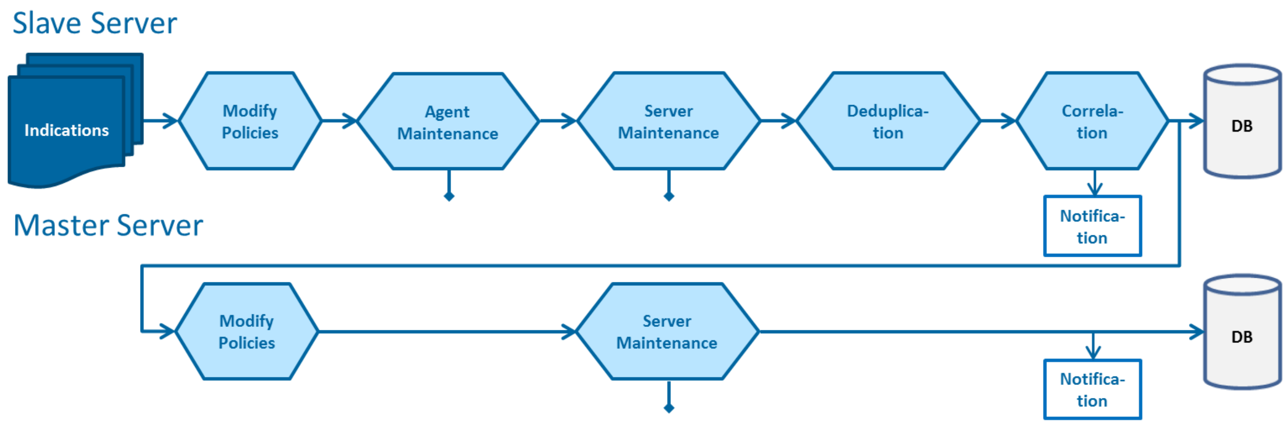

Indication's workflow
Active
• new indications
Closed
• Inserted as closed (policy condition flag)
• Closed by operator
• Auto-closed based on key (correlation)
Archived
• Archived by operator
• Auto-archived based on AUTO_ARCHIVE parameter of the server
Deleted
• Auto-deleted based on AUTO_DELETE parameter of the server
• Cleaned directly from the database by external tools
Severities
U (0) unknown
N (1) normal
W (2) warning
M (3) minor
M (4) major
C (5) critical

> The *boom* server and agent truncates most indication attributes to a maximum size of characters. For detailed information see chapter Indication Attribute Sizes.

4.3.1. Indication Processing on the *boom* Server

After an agent finished the processing of it's monitored data sources and submitted an indication to the server a next round of processing is taking place on the server.
Each incoming indication is processed by the server in multiple steps before it gets displayed to the operator.





Each step is described in more detail in the according chapter in this manual.

Just as short overview:

Modify Policies: this step allows to change attributes of incoming indications.

AdHoc Maintenance: active maintenances, in this case initiated by the agent, can suppress indications or move them to the "outage" indication stream.

Scheduled Maintenance: active maintenances, in this case initiated by the server, can suppress indications or move them to the "outage" indication stream.

Deduplication: the server checks based on indication attributes or the indication key if this indication already exists and just increments the duplicate counter of the existing indication instead of creating a new one.

Correlation: an indication might have a close mask. This close mask defines which other indications should be closed when this indication arrives. In this step the server performs the according close operations.

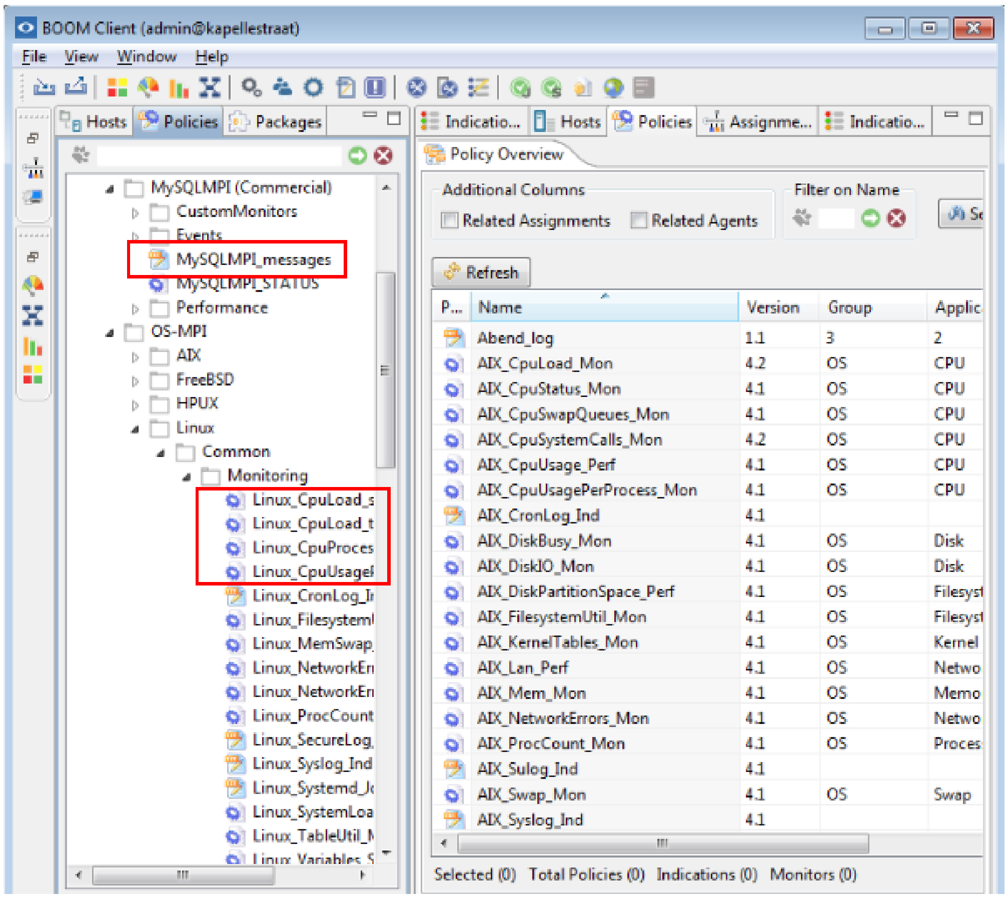
After the last step is performed the indication gets stored in the database, gets visible to the users and might trigger notifications.

Chapter 5. Monitoring

All monitoring, calculation and filtering made by the *boom* agent are defined in deployed policies. A policy is a sort of configuration file that informs the agent what has to be monitored, when, how and what kind of results should be delivered to the central *boom* server.

boom knows two types of policies:

Indication Policy:	This type of policy is used for all text related processing.
Monitor Policy:	This type of policy is used for all value based monitoring (thresholding).



- MySQLMPI (Commercial)
  - CustomMonitors
  - Events
  - MySQLMPI_messages
  - MySQLMPI_STATUS
  - Performance
- OS-MPI
  - AIX
  - FreeBSD
  - HPUX
  - Linux
    - Common
      - Monitoring
        - Linux_CpuLoad_s
        - Linux_CpuLoad_t
        - Linux_CpuProces
        - Linux_CpuUsageP
        - Linux_CronLog_Ir
        - Linux_Filesysteml
        - Linux_MemSwap
        - Linux_NetworkEr
        - Linux_NetworkEr
        - Linux_ProcCount
        - Linux_SecureLog
        - Linux_Syslog_Ind
        - Linux_Systemd_Ir
        - Linux_SystemLoa
        - Linux_TableUtil_M
        - Linux_Variables_S

P...	Name	Version	Group	Applic
	Abend_log	1.1	3	2
	AIX_CpuLoad_Mon	4.2	OS	CPU
	AIX_CpuStatus_Mon	4.1	OS	CPU
	AIX_CpuSwapQueues_Mon	4.1	OS	CPU
	AIX_CpuSystemCalls_Mon	4.2	OS	CPU
	AIX_CpuUsage_Perf	4.1	OS	CPU
	AIX_CpuUsagePerProcess_Mon	4.1	OS	CPU
	AIX_CronLog_Ind	4.1		
	AIX_DiskBusy_Mon	4.1	OS	Disk
	AIX_DiskIO_Mon	4.1	OS	Disk
	AIX_DiskPartitionSpace_Perf	4.1	OS	Filesyst
	AIX_FilesystemUtil_Mon	4.1	OS	Filesyst
	AIX_KernelTables_Mon	4.1	OS	Kernel
	AIX_Lan_Perf	4.1	OS	Netwo
	AIX_Mem_Mon	4.1	OS	Memo
	AIX_NetworkErrors_Mon	4.1	OS	Netwo
	AIX_ProcCount_Mon	4.1	OS	Proces
	AIX_Sulog_Ind	4.1		
	AIX_Swap_Mon	4.1	OS	Swap
	AIX_Syslog_Ind	4.1		

Selected (0) Total Policies (0) Indications (0) Monitors (0)

5.1. Monitor Policy (Threshold monitoring)

5.1.1. General Information

The concept of threshold monitoring is well-known in the IT administration community. Any threshold monitor consists of two major parts:

- An executable or a script that is able to submit a metric value from the monitored component.
- A Monitor Policy with configured threshold levels that defines different conditions for the values.

Every declared threshold defines some numeric (double) value. Generally the metric value becomes important for an operator when a value crosses a defined level. For example: The free disk space value is not important for productive environments while it stays above some minimum. To prevent an "out of disk space" situation an administrator has to specify the threshold for the free disk space monitor that will give enough time to investigate and take corrective actions before the system will crash or degrade.

Due to the nature of an IT environment the threshold levels are different from system to system and must therefore be configurable. Another aspect of the thresholds is that ONE threshold value can't reflect the dynamic of changes in an adequate way. Multiple thresholds with different severities will give a clearer picture and more time to prevent problems.

Another side of the monitors are the different objects of the same type that need to be monitored. Like each partition of a disk has a different size. In an ideal case one monitor should be able to deliver values for multiple objects and these objects can have different thresholds.

In the *boom* infrastructure every monitor requires to have a monitor policy. The monitor policy contains all necessary definitions to start the monitor binary (if required) and react on submitted values.

Configured and uploaded to the *boom* agent, the monitor policy is an instruction that declares what to trigger and how to process the data. All policies in the *boom* environment must have unique names. These names must be used for submitting monitor values. In other words, if a monitor policy named as "Monitor_A" exists, this monitor policy will be used for checking values that are submitted as Monitor_A="double_value". The monitor binary can submit values for multiple instances by setting the object attribute for each instance.

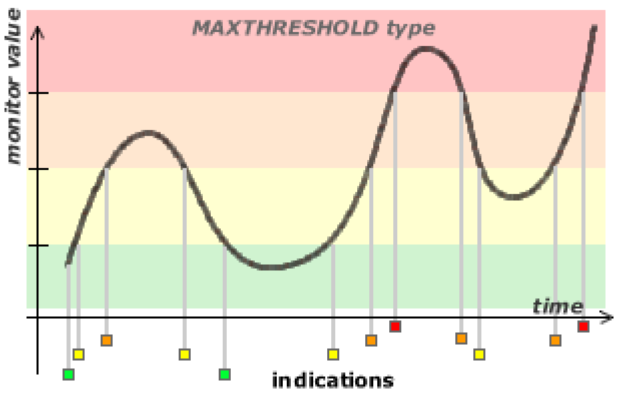
Monitor Types and Variations

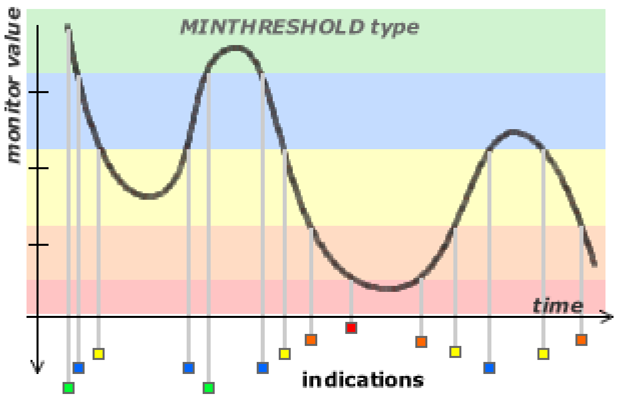
Threshold monitors have different types and variations. The two main categories are **MAXTHRESHOLD** and **MINTHRESHOLD**.

For example: A 'free disk space' monitor has a **MINTHRESHOLD** type, but CPU utilization has a MAXTHRESHOLD.
In perspective of call types - **EXEC**, **JAVA** and **EXTERNAL** types are supported.
Some more variations are policy **WITH RESET** or **WITHOUT RESET**.

Maximum and Minimum

The thresholds are one of the major aspects of the monitors. They allow to reduce the number of indications that are coming to an operator screen. The *boom* product operates with two types of thresholds:



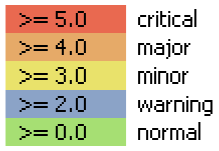


Since thresholds are part of a policy, it makes sense to generalize the threshold type to the policy. Even more - a threshold is not only one attribute that is used during processing, so a policy has a list of conditions and every condition has an attribute named 'threshold'. Of course, every condition contains a list of additional attributes that are used during the creation of an indication.

A monitor policy of the MAXTHRESHOLD type declares the number of conditions with maximum threshold levels in descending order. Opposite MINTHRESHOLD monitor policy expects ascending order of the thresholds.

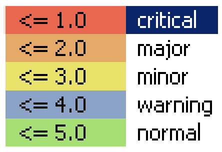
This requirement is important since all thresholds will be checked by the calculation engine one after another. The first matched condition notifies the engine to stop processing all following conditions.

MAXTHRESHOLD



>= 5.0	critical
>= 4.0	major
>= 3.0	minor
>= 2.0	warning
>= 0.0	normal

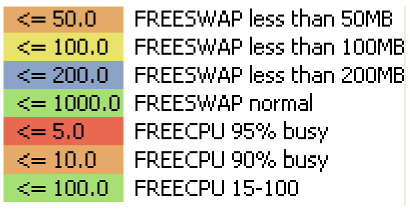
MINTHRESHOLD



<= 1.0	critical
<= 2.0	major
<= 3.0	minor
<= 4.0	warning
<= 5.0	normal

When a threshold value will be crossed the first time, the *boom* agent creates an indication and sends it to the *boom* server. Afterwards the agent keeps silent until one of the following values crosses a different threshold level. This suppression algorithm reduces the number of messages received by the operator as well as network traffic.

It is possible to have multiple conditions with the same severity, it is also possible to skip unnecessary severities. Supported object filters allow to combine multiple condition sets for different objects in one policy.



<= 50.0	FREESWAP less than 50MB
<= 100.0	FREESWAP less than 100MB
<= 200.0	FREESWAP less than 200MB
<= 1000.0	FREESWAP normal
<= 5.0	FREECPU 95% busy
<= 10.0	FREECPU 90% busy
<= 100.0	FREECPU 15-100

During the processing of the calculation engine, only conditions will be taken into account that matches with the object value which has been submitted together with the monitor value. If an object mask is not specified - all objects will be processed by the particular condition.

MAXONLYCHANGES and MINONLYCHANGES

Starting from v2.55 monitor policies supports two new types: MAXONLYCHANGE and MINONLYCHANGES. These types can be used for monitoring not frequently changed values. For such types a new indication will be created for every change detected. The policy conditions are used only to detect severity and to define an indication attribute but no thresholding is used.

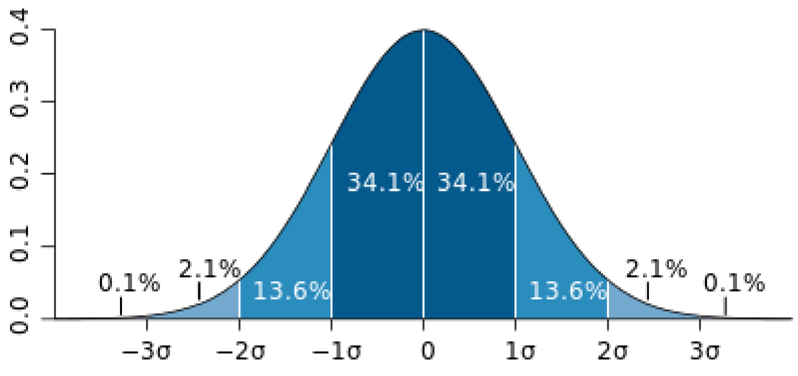
STDDEV

Starting from v5.5 Agent, monitor policies support type STDDEV. This is standard deviation monitor based on well known statistic method (https://en.wikipedia.org/wiki/Standard_deviation).

The monitor policy of STDEV type will be do following steps:

- Agent starts collecting values per policy and object into history series.
- Collection of values is not differ from any other Monitor policy.
- Policy will transform original observed value into "Distance" value.
- On first 10 observation, the "Distance" value is always 0 (zero).
- After first 10 observed values, the Agent starts calculation of real "Distance" value.

- The "Distance" will be checked agains specified in the policy threshold values.

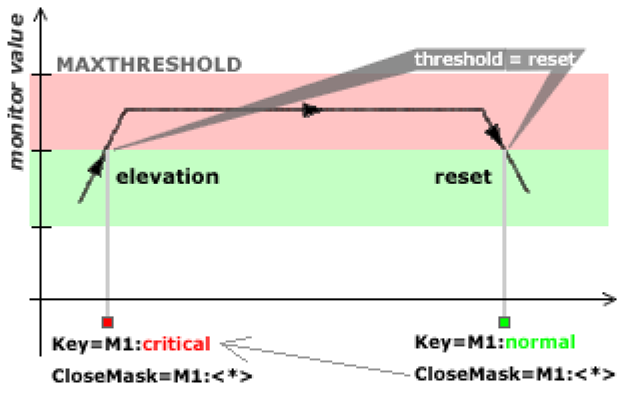
> Distance - (D) is an absolute distance between last observed value (V) and the Mean (M) of previous observations normalized by standard deviation (σ).
> D = abs((V - M) / σ)
>
> 
>
>
> Therefore if the policy has threshold >= 3 it will send an indications only for ~ 0,2% of values from far left and far right bands shown on the plot displayed above.

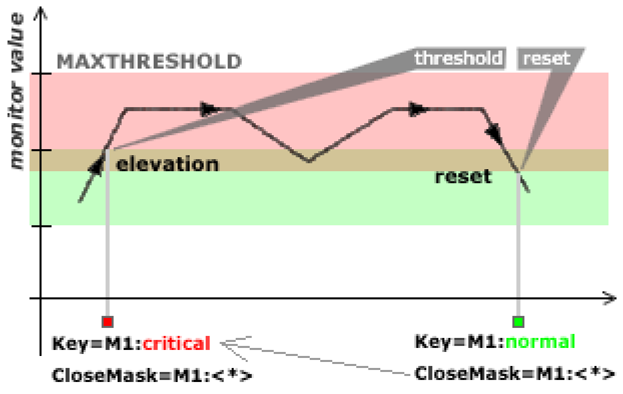
Reset Values

All monitor policies' thresholds are coming with reset values. The "Reset" concept is playing an important role in the calculation.

First of all, the monitor policy has a global flag called **"Policy with reset"**. If this flag is set to **'NO'**, the policy is ignoring all reset values as well as the silence periods and it will deliver values all the time when it is submitted to the agent. This type of monitors is also known as **'Continuous Monitors'**.
If the **'YES'** value is selected - it will be a threshold monitor with reset.

When a monitor value crosses a threshold in the defined direction (increasing for MAXTHRESHOLD and decreasing for MINTHRESHOLD) - it can be named "elevation". Backward direction is a "reset".
Monitors with a reset value different from the threshold value have some special handling in the "reset" direction. The reset value gives the possibility to ignore small value’s fluctuations and keeps the reached threshold unchanged.





The reset feature can be explained with the following example:

A process CPU utilization monitor has a critical threshold = 95% indicating process high CPU load. A normal threshold that indicates a normal state is above 0%. When the process reaches 95%, an indication will be sent to the server. Lets assume the critical condition has specified a reset value = 70%, this allows to keep the critical level unchanged until the process goes down below 70% CPU. So the deviations between 70 and 95 percent will not reset the severity to normal.

An other example is the MINTHRESHOLD of free disk space:

A critical condition has a threshold equal to 100MB. The reset value is 1024MB. As result of this a critical indication like "100 MB free space left on a disk…" can be kept active until an administrator cleans up disk space up to at least 1GB.

The minimum type threshold requires a reset value to be bigger than the threshold, the maximum type threshold requires the reset value to be less than the threshold.

Another optional possibility in the condition section of a monitor policy is the *"Ignore Reset"* flag. This flag is set to **"NO"** by default. In case of switching the flag's value to **"YES"** the threshold condition becomes a 'continuous' nature. That means that on this level any submitted monitor value generates an indication. This can be used for more precise monitoring of critical conditions.

The *"Silence Count"* parameter of a condition can be used when it is necessary to suppress a couple of first generated values. The *boom* agent will ignore the specified amount of submitted values that match with the condition before an indication will be sent to the *boom* server.

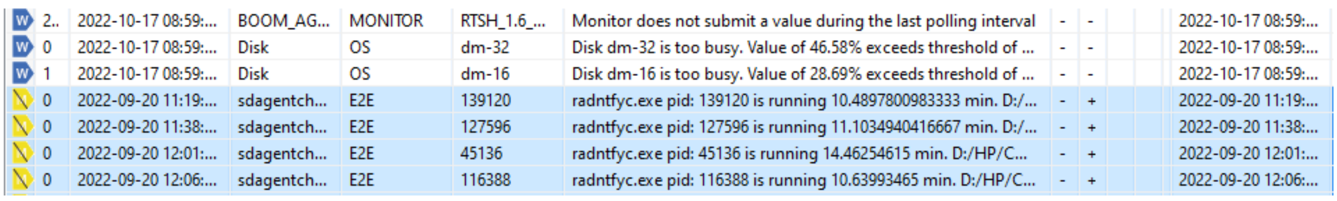
If an indication is delivered with *"Close Mask"* - the server is able to automatically close related previous indications.

The default working directory for the monitor's executable is "$BOOM_ROOT/spi/". All binaries and script calls must be specified relative to this directory. In case it is necessary to use a binary that is placed in a different location that is available in the PATH variable - use the '#' character as a prefix. i.e. #df, #top

Finished Alerts

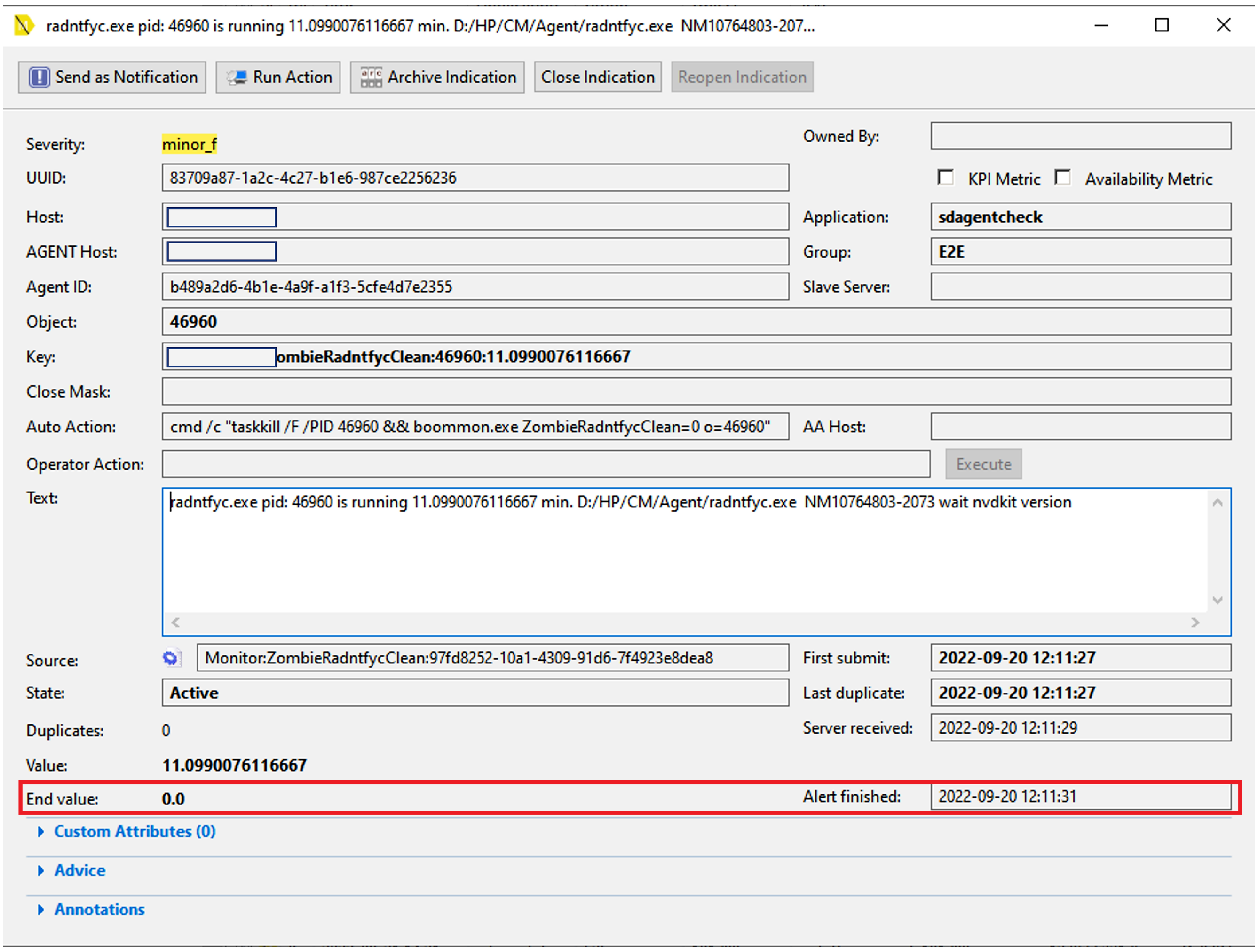
The *"Alert Finished"* state will be reached when the monitor submits a value outside the defined threshold borders. In case of MAXTHRESHOLD it's below the lowest threshold value and for MINTHRESHOLD - biggest one. This state indicates the end of the previous state and enables operators to identify if a problem is still ongoing. Beside this benefit for the exception based operations concept, it also avoids the sending of indications with normal severity.

The indication browser displays such indications with severities crossed by line:



W	2..	2022-10-17 08:59:...	BOOM_AG...	MONITOR	RTSH_1.6_...	Monitor does not submit a value during the last polling interval	-	-			2022-10-17 08:59:...
W	0	2022-10-17 08:59:...	Disk	OS	dm-32	Disk dm-32 is too busy. Value of 46.58% exceeds threshold of ...	-	-			2022-10-17 08:59:...
W	1	2022-10-17 08:59:...	Disk	OS	dm-16	Disk dm-16 is too busy. Value of 28.69% exceeds threshold of ...	-	-			2022-10-17 08:59:...
🟨	0	2022-09-20 11:19:...	sdagentch...	E2E	139120	radntfyc.exe pid: 139120 is running 10.4897800983333 min. D:/...	-	+			2022-09-20 11:19:...
🟨	0	2022-09-20 11:38:...	sdagentch...	E2E	127596	radntfyc.exe pid: 127596 is running 11.1034940416667 min. D:/...	-	+			2022-09-20 11:38:...
🟨	0	2022-09-20 12:01:...	sdagentch...	E2E	45136	radntfyc.exe pid: 45136 is running 14.46254615 min. D:/HP/C...	-	+			2022-09-20 12:01:...
🟨	0	2022-09-20 12:06:...	sdagentch...	E2E	116388	radntfyc.exe pid: 116388 is running 10.63993465 min. D:/HP/C...	-	+			2022-09-20 12:06:...

After an alert is finished you can see in the indication details the time stamp and last value that triggered the finished state.



radntfyc.exe pid: 46960 is running 11.0990076116667 min. D:/HP/CM/Agent/radntfyc.exe NM10764803-207...

Severity:	minor_f	Owned By:	
UUID:	83709a87-1a2c-4c27-b1e6-987ce2256236	☐ KPI Metric ☐ Availability Metric	
Host:	[redacted]	Application:	sdagentcheck
AGENT Host:	[redacted]	Group:	E2E
Agent ID:	b489a2d6-4b1e-4a9f-a1f3-5cfe4d7e2355	Slave Server:	
Object:	46960		
Key:	[redacted]ombieRadntfycClean:46960:11.0990076116667		
Close Mask:			
Auto Action:	cmd /c "taskkill /F /PID 46960 && boommon.exe ZombieRadntfycClean=0 o=46960"	AA Host:	
Operator Action:		Execute	
Text:	radntfyc.exe pid: 46960 is running 11.0990076116667 min. D:/HP/CM/Agent/radntfyc.exe NM10764803-2073 wait nvdkit version		
Source:	Monitor:ZombieRadntfycClean:97fd8252-10a1-4309-91d6-7f4923e8dea8	First submit:	2022-09-20 12:11:27
State:	Active	Last duplicate:	2022-09-20 12:11:27
Duplicates:	0	Server received:	2022-09-20 12:11:29
Value:	11.0990076116667		
End value:	0.0	Alert finished:	2022-09-20 12:11:31

5.1.2. Output Parser Monitor (OPM)

The Output Parser Monitor (OPM) bundled with *boom* allows you to create a monitor based on the output produced by system tools or other executables. OPM is a Java based monitor that is able to parse and summarize result text lines containing numeric data.

Overall processing by OPM monitors consists of following steps:

1. Execute command which output contains data to be monitored
2. Parse output using regular expressions
3. Calculate object name using values extracted from command output
4. Calculate monitor value using values extracted from command output
5. Submit monitor value

Most of the standard system tools like "top", "df" or "ps" are delivering results with useful numeric values for multiple objects. These values can be used for the system or process monitoring.

OPM also supports some Java based actions that can generate output for parsing as well:

SshAction
: can be used for monitoring remote systems over SSH.

JmxAction
can be used for monitoring remote JMX servers/Java Beans.

HttpAction
can be used for parsing remote HTTP pages.

StrAction
allows to split, to filter and to transform blocks contained in the output into one single line.

Syntax of OPM supported Java Actions:

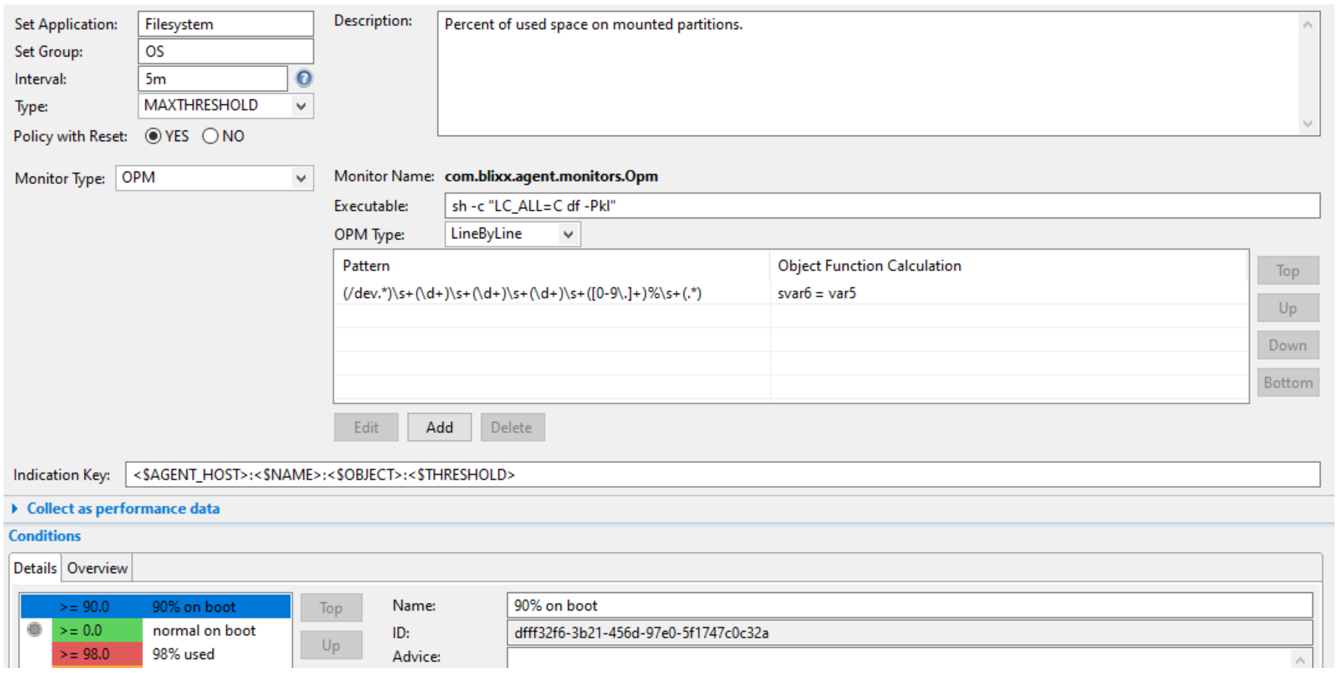
```
SshAction:  <host> <user> <pass> <command to execute on remote system>
JmxAction:  [options] -u <url>
               with options:
                -l <login>
                -p <password>
                -o <objecName> JMX canonical object name mask. Like java.*:type=Memory"
                -a <attribute>
                -v <valueMask> only masked attribute will be queried
                -D <environment variables>

HttpAction:   <URL>
StrAction:    —exec <command_or_script> -sbp <start_block_java_patter>
                      [-ebp <end_block_java_pattern>] [—alp <accept_line_java_pattern>]
                      [-sep <separator>]

                -sbp defines the start pattern of the beginning of the block to be converted
                -ebp defines the end pattern of the end of the block to be converted
                -alp defines patterns to filter only necessary strings,
                     multiple alp patterns can be specified: -alp "p1" "alp" "p2"
                -sep defines a separator to be used in the resulting/converted string
```

To better explain the function of the OPM type, let’s take an example of processing the `df` output of a Unix system. We will use the `sh —c "LC_ALL=C df —Pkl"` call to have a similar output for most of the Unix systems:

```
File System        1024-blocks Used    Available   Capacity    Mounted on
/dev/vg00/lvol5    24456       8472    15984       35%         /home
/dev/dsk/c0t0d0    3235348     3235348 0           100%        /iuxcdrom0
/dev/vg00/lvol6    5235064     4167056 1068008     80%         /opt
/dev/vg00/lvol4    715400      8632    706768      2%          /tmp
/dev/vg00/u01      7693337     213287  7480050     3%          /u01
/dev/vg00/lvol7    3289800     2853656 436144      87%         /usr
/dev/vg00/lvol8    6118216     1811520 4306696     30%         /var
/dev/vg00/lvol1    309928      130312  179616      43%         /stand
/dev/vg00/lvol3    409016      327264  81752       81%         /
                   174116      0       174116      0%          /dev/shm
```

Most of these lines are showing really important values for monitoring. But every line provides information for a different mounted partition. So, to be able to monitor mounted partitions, a monitor must submit a separate value for each object.

Let’s assume we want to monitor the percentage of used space on each volume. We need to create a new OPM with the MAXTHRESHOLD type and OPM type "LineByLine".



Set Application:	Filesystem
Set Group:	OS
Interval:	5m
Type:	MAXTHRESHOLD
Policy with Reset:	[✓ YES] NO
Description:	Percent of used space on mounted partitions.
Monitor Type:	OPM
Monitor Name:	**com.blixx.agent.monitors.Opm**
Executable:	sh -c "LC_ALL=C df -Pkl"
OPM Type:	LineByLine

Pattern	Object Function Calculation
(/dev.*)\s+(\d+)\s+(\d+)\s+(\d+)\s+([0-9\.]+)%\s+(.*)	svar6 = var5

Indication Key: <$AGENT_HOST>:<$NAME>:<$OBJECT>:<$THRESHOLD>

Collect as performance data

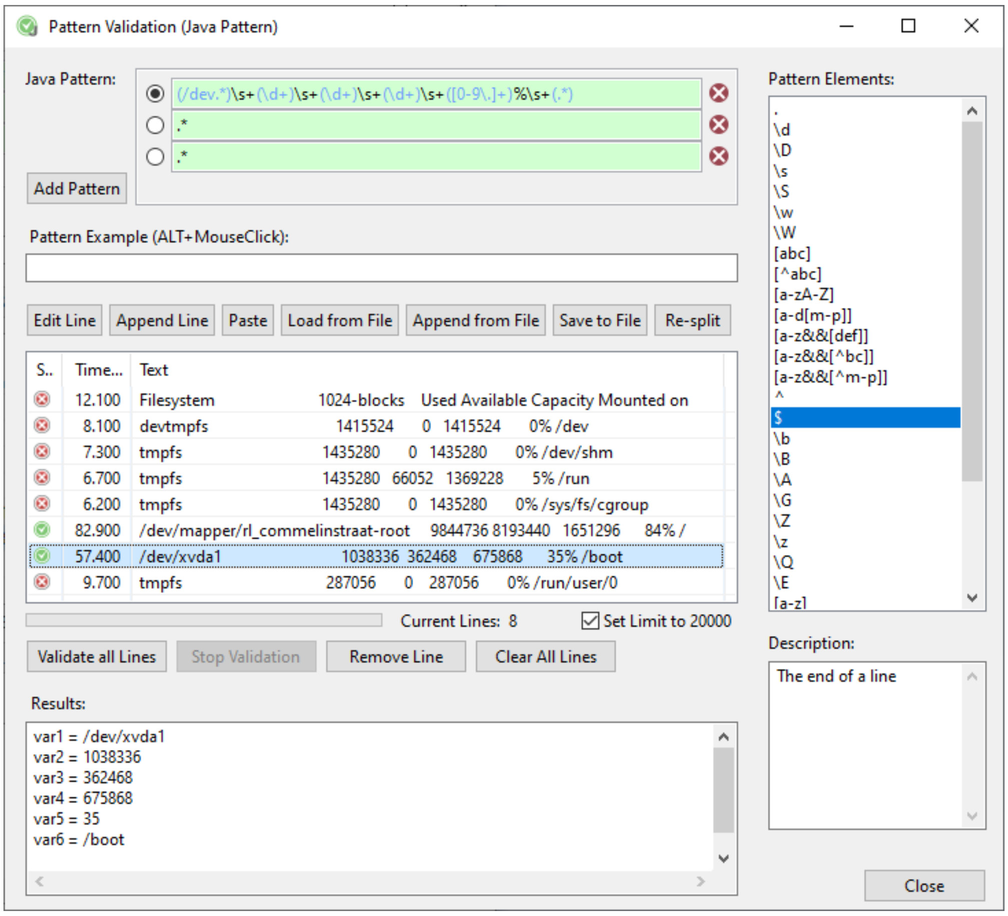
Conditions

>= 90.0	90% on boot
>= 0.0	normal on boot
>= 98.0	98% used

Name:	90% on boot
ID:	dfff32f6-3b21-456d-97e0-5f1747c0c32a
Advice:	

In this example we are using the following pattern, which has been validated by the built-in pattern validator (see screenshot):

```
(/dev.*)\s+(\d+)\s+(\d+)\s+(\d+)\s+([0-9\.]+)%\s+(.*)
```



Pattern Validation (Java Pattern)

Java Pattern:

(/dev.*)\s+(\d+)\s+(\d+)\s+(\d+)\s+([0-9\.]+)%\s+(.*)
.*
.*

Pattern Example (ALT+MouseClick):

S..	Time...	Text
✗	12.100	Filesystem 1024-blocks Used Available Capacity Mounted on
✗	8.100	devtmpfs 1415524 0 1415524 0% /dev
✗	7.300	tmpfs 1435280 0 1435280 0% /dev/shm
✗	6.700	tmpfs 1435280 66052 1369228 5% /run
✗	6.200	tmpfs 1435280 0 1435280 0% /sys/fs/cgroup
✓	82.900	/dev/mapper/rl_commelinstraat-root 9844736 8193440 1651296 84% /
✓	57.400	/dev/xvda1 1038336 362468 675868 35% /boot
✗	9.700	tmpfs 287056 0 287056 0% /run/user/0

Current Lines: 8 [✓ Set Limit to 20000]

Results:

```
var1 = /dev/xvda1
var2 = 1038336
var3 = 362468
var4 = 675868
var5 = 35
var6 = /boot
```

Pattern Elements:

```
.
\d
\D
\s
\S
\w
\W
[abc]
[^abc]
[a-zA-Z]
[a-d[m-p]]
[a-z&&[def]]
[a-z&&[^bc]]
[a-z&&[^m-p]]
^
$
\b
\B
\A
\G
\Z
\z
\Q
\E
[a-z]
```

Description:

The end of a line

This pattern defines 6 capturing groups that provide 6 variables for every matched line (can be seen also in the "Results" area of the patter validator):

var1
partition name

var2
disk size in KB

var3
used space

var4
free space

var5
used space in %

var6
mount point

In the OPM Monitor all defined variables can be accessed as DOUBLE or STRING data type.
If values are provided as DOUBLE and you want to use them as numeric values - use varN.
If you need to get a value as STRING - use svarN.

Every matched line in our example is a potential monitor value. To submit multiple monitor values we need the object attribute. So the line:

```
svar6 = var5
```

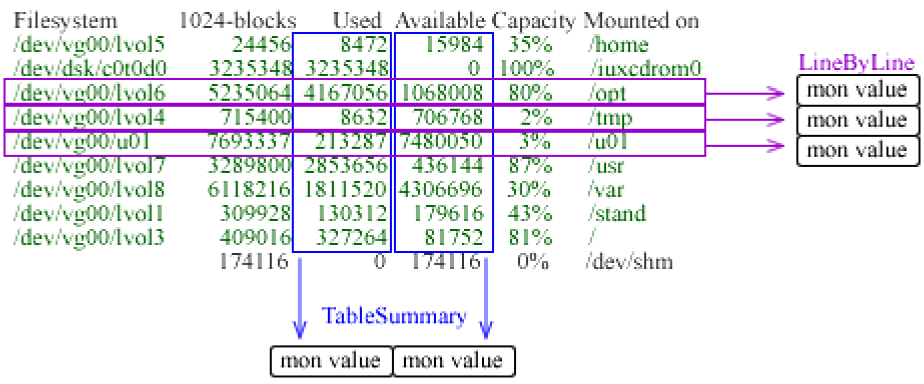
tells to the OPM how to submit the parsed values (the object is defined by "svar6" and the value for this object by "var5").

In this example the monitor defines an object for every partition mounted on a volume whose name begins with string "/dev" and assigns the "Percent Used" value to this object. E.g.: /home = 35.

Three types of OPM are available:

The three OPM types **"LineByLine"**, **"TableSummary"** and **"PunchCard"** define the general behaviour of the monitor:

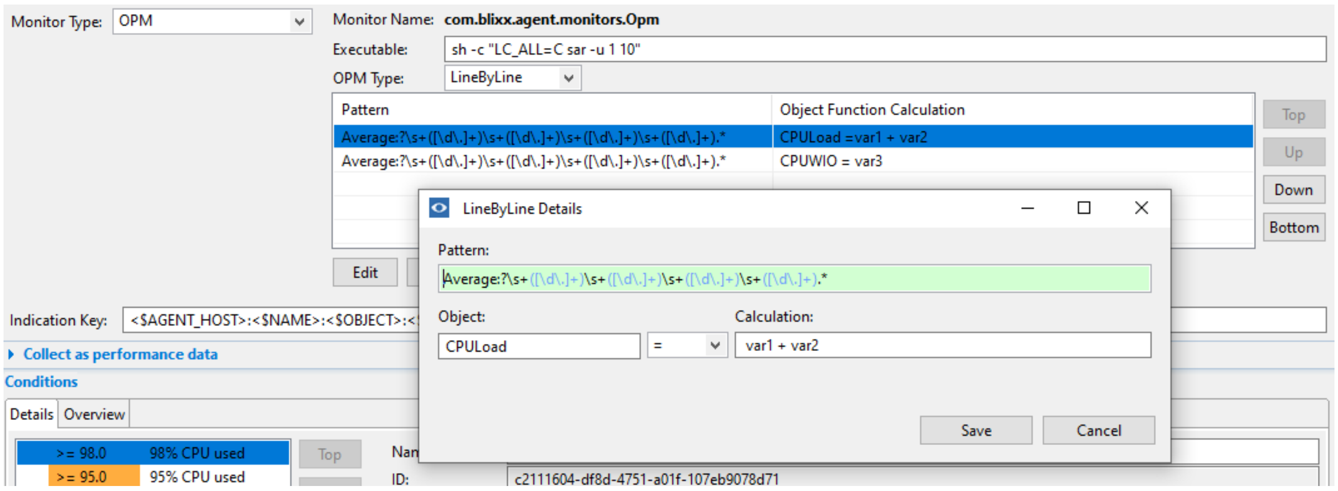
- **"LineByLine"** defines that every line will be processed separately. All filtered lines will be parsed, calculated and submitted as monitor values.
- **"TableSummary"** defines that all lines will be parsed and the result will be converted into a table. Selected columns will be used to calculate the monitor value. Each column is declared by variables respectively a capturing group in the pattern.
- **"PunchCard"** is used for cases where it is necessary to calculate the monitor value using values that are not contained in a single line nor a single column.



```
Filesystem        1024-blocks    Used  Available Capacity Mounted on
/dev/vg00/lvol5        24456     8472      15984    35%    /home
/dev/dsk/c0t0d0      3235348  3235348          0   100%    /iuxcdrom0
/dev/vg00/lvol6      5235064  4167056    1068008    80%    /opt
/dev/vg00/lvol4       715400     8632     706768     2%    /tmp
/dev/vg00/u01        7693337   213287    7480050     3%    /u01
/dev/vg00/lvol7      3289800  2853656     436144    87%    /usr
/dev/vg00/lvol8      6118216  1811520    4306696    30%    /var
/dev/vg00/lvol1       309928   130312     179616    43%    /stand
/dev/vg00/lvol3       409016   327264      81752    81%    /
                      174116        0     174116     0%    /dev/shm
```

The usage of the different OPM types is described in the following configuration overview:

LineByLine:



Monitor Type: OPM

Monitor Name: **com.blixx.agent.monitors.Opm**

Executable: sh -c "LC_ALL=C sar -u 1 10"

OPM Type: LineByLine

Pattern	Object Function Calculation
Average:?\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+).*	CPULoad =var1 + var2
Average:?\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+).*	CPUWIO = var3

Indication Key: <$AGENT_HOST>:<$NAME>:<$OBJECT>:<

LineByLine Details

Pattern: Average:?\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+).*

Object: CPULoad =

Calculation: var1 + var2

>= 98.0	98% CPU used
>= 95.0	95% CPU used

ID: c2111604-df8d-4751-a01f-107eb9078d71

Executable field

definition of the command whose execution generates the output to be parsed.
To add a new object function calculation click the "Add" button to open the **LineByLine Details window**.

Pattern field

defines the pattern for the line(s) to be matched and declares STRING and DOUBLE variables through capturing groups.

Object field

defines the object to be monitored. It has to be a string, a string variable svarN or a combination of these two (e.g. svar1_test). It must not contain a white space character.

Operator drop-down list

for the calculation of the object value one of the following operators can be chosen:

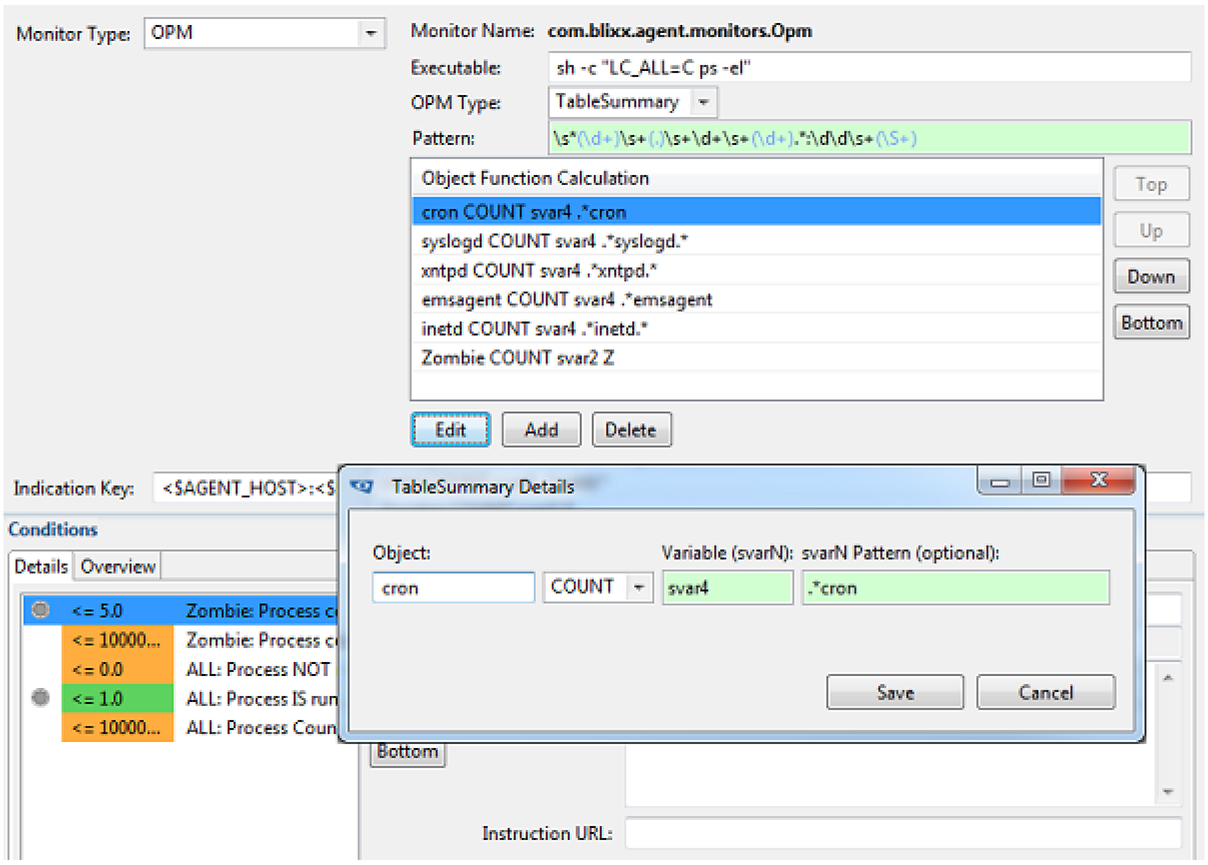
Operator	Description
'='	assigns the result of the calculation to the object
DELTA	assigns only the difference of the previous and the new value to the object

Calculation field

contains the numeric calculation of the value to be assigned to the specified object. The calculation syntax is described further down as it affects all OPM types.

For type **"LineByLine"** many different values can be parsed and calculated through different patterns and assigned to any number of objects.

TableSummary:



Monitor Type: OPM

Monitor Name: **com.blixx.agent.monitors.Opm**

Executable: `sh -c "LC_ALL=C ps -el"`

OPM Type: TableSummary

Pattern: `\s*(\d+)\s+(.)\s+\d+\s+(\d+).*:\d\d\s+(\S+)`

Object Function Calculation
cron COUNT svar4 .*cron
syslogd COUNT svar4 .*syslogd.*
xntpd COUNT svar4 .*xntpd.*
emsagent COUNT svar4 .*emsagent
inetd COUNT svar4 .*inetd.*
Zombie COUNT svar2 Z

Indication Key: <$AGENT_HOST>:<$

TableSummary Details

Object:		Variable (svarN):	svarN Pattern (optional):
cron	COUNT	svar4	.*cron

Executable field

definition of the command whose execution generates the output to be parsed.

Pattern field

defines the pattern for the line(s) to be matched and declares STRING and DOUBLE variables through capturing groups. For type "TableSummary" only one global pattern is defined.

To add a new object function calculation click the "Add" button to open the **TableSummary Details window**

Object field

defines the object to be monitored. It has to be a string and must not contain a white space character.

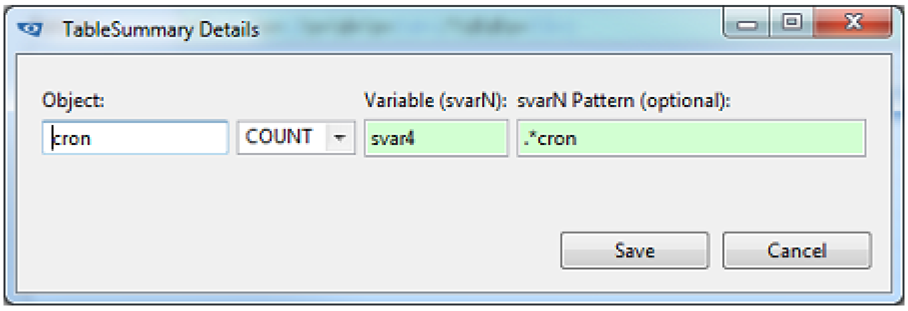
Operator drop-down list

for the calculation of the object value one of the following operators can be chosen:

Operator	Description
SUM	All values of one similar object will be summed.
MAX	The highest value of an object will be taken.
MIN	The lowest value of an object will be taken.
AVG	The average value of an object will be taken.
COUNT	The number of occurrences of an object will be taken.

Depending on the chosen operator, the following fields are available to configure the monitor:

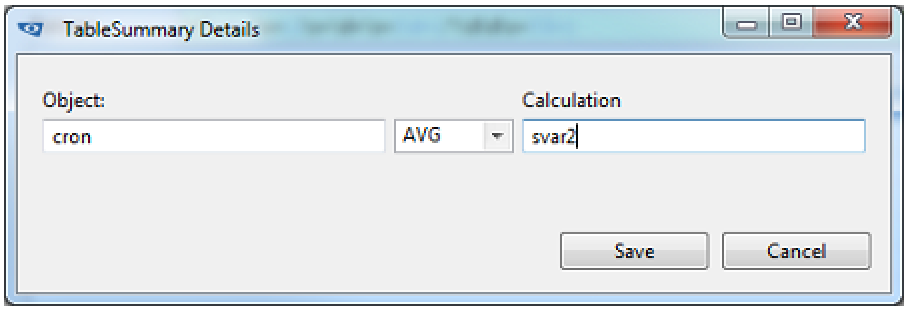
For operator "COUNT"



Object: cron | COUNT
Variable (svarN): svar4
svarN Pattern (optional): .*cron

Variable(svarN) field: contains the string variable which shall be counted.
svarN Pattern field: optionally contains a pattern to filter the string contained in the variable to be counted.

For all other operators

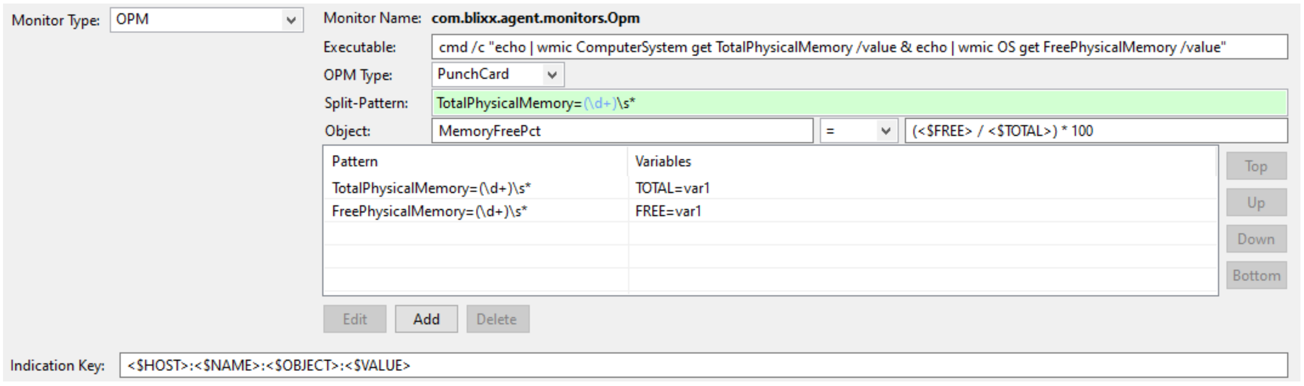


Object: cron | AVG
Calculation: svar2

The object field has to include a string (for both types), a string variable svarN (only for type "LineByLine" or a combination of these two (e.g. svar1_test, only for "LineByLine" type). It must **not contain a white space** character.

Calculation field: contains the numeric calculation whose results will be used by the defined operator to determine the object's value. The calculation syntax is described further down as it affects all OPM types.

PunchCard



Monitor Type: OPM
Monitor Name: **com.blixx.agent.monitors.Opm**
Executable: cmd /c "echo | wmic ComputerSystem get TotalPhysicalMemory /value & echo | wmic OS get FreePhysicalMemory /value"
OPM Type: PunchCard
Split-Pattern: TotalPhysicalMemory=(\d+)\s*
Object: MemoryFreePct | = | (<$FREE> / <$TOTAL>) * 100

Pattern	Variables
TotalPhysicalMemory=(\d+)\s*	TOTAL=var1
FreePhysicalMemory=(\d+)\s*	FREE=var1

Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$VALUE>

Executable field: definition of the command(s) whose execution generates the output to be parsed.

Split-Pattern field: This pattern is used to split the output into blocks, each of it containing the information to calculate all values of one instance, e.g. per network interface, disk, etc. In order to consider the whole output as a one block you can use a regular expression which does not match anything, e.g. "!!!!.*".

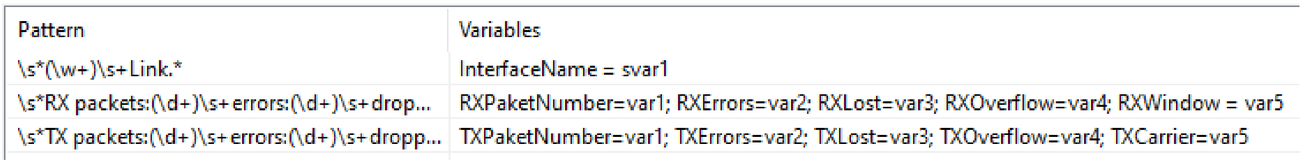
Object fields: the first field defines object instance. It can be a fixed string, a string variable svarN from the variables defined below or a combination of these two (e.g. svar1_test).
In the drop-down list of the second field select the operator that is used to calculate the value:

Operator	Description
'='	assigns the result of the calculation to the object

Operator	Description
DELTA	assigns only the difference of the previous and the new value to the object

In the third field specify the numeric calculation of the value for the object. The calculation syntax is described further down as it affects all OPM types.

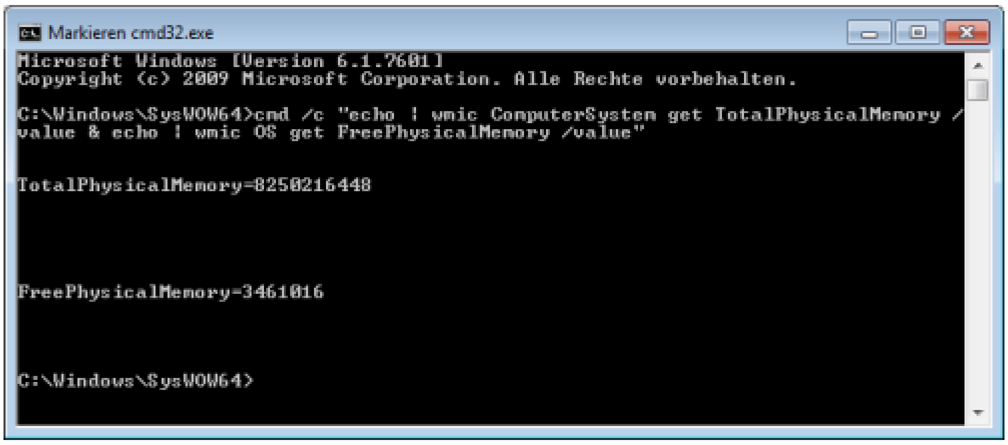
Pattern list: Each line consists of a pattern (regular expression) with capturing groups defined by the brackets "(...)" and the definition of the corresponding named variables based on capturing group values. It is possible to define multiple named variables for a single pattern. In this case named variable assignments need to be separated by semicolon. For a string variable use svar1, for a variable of type Double use var1.



Pattern	Variables
\s*(\w+)\s+Link.*	InterfaceName = svar1
\s*RX packets:(\d+)\s+errors:(\d+)\s+drop...	RXPaketNumber=var1; RXErrors=var2; RXLost=var3; RXOverflow=var4; RXWindow = var5
\s*TX packets:(\d+)\s+errors:(\d+)\s+dropp...	TXPaketNumber=var1; TXErrors=var2; TXLost=var3; TXOverflow=var4; TXCarrier=var5

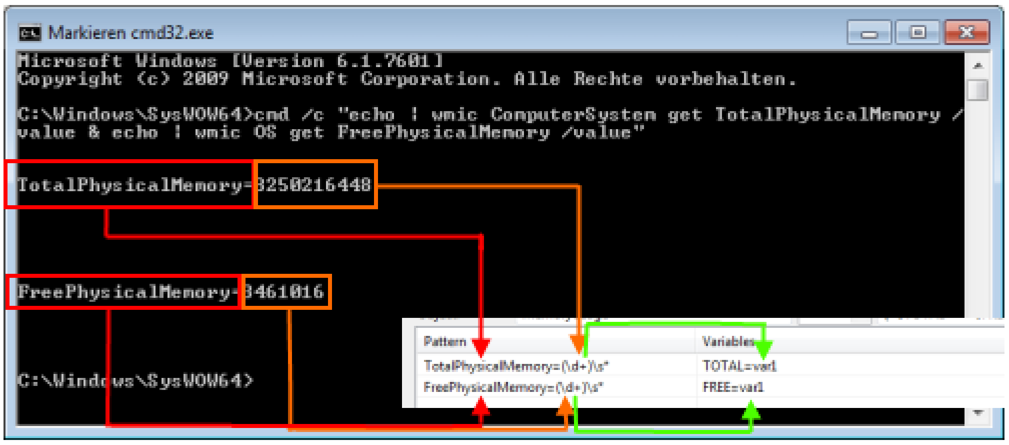
The named variables of type double can be used in the calculation rule for the Object value, the string based variables can be used to construct the object instance name.

Example: the executable call of the policy shown at the beginning of the section will produce the following output:



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\SysWOW64>cmd /c "echo | wmic ComputerSystem get TotalPhysicalMemory /
value & echo | wmic OS get FreePhysicalMemory /value"


TotalPhysicalMemory=8250216448




FreePhysicalMemory=3461016




C:\Windows\SysWOW64>
```

The Split-Pattern matches the line starting with "TotalPhysicalMemory=". Since no other line matches, OPM will handle the complete output as one block and matches the two patterns against this block.

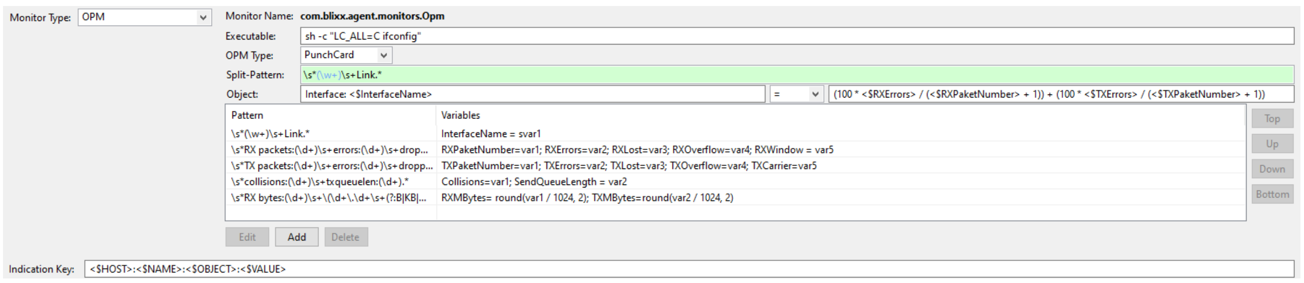


```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Windows\SysWOW64>cmd /c "echo | wmic ComputerSystem get TotalPhysicalMemory /
value & echo | wmic OS get FreePhysicalMemory /value"

TotalPhysicalMemory=8250216448

FreePhysicalMemory=3461016

C:\Windows\SysWOW64>
```

Pattern	Variables
TotalPhysicalMemory=(\d+)\s*	TOTAL=var1
FreePhysicalMemory=(\d+)\s*	FREE=var1

As result the variables TOTAL and FREE are populated in this policy with the values returned by the command call and are used in the Object value fields to calculate the percentage of used memory as (<$TOTAL> - <$FREE>) * 100 / <$TOTAL>.

This sample is using the ifconfig command to show how the processing of multiple blocks and mulitple variables per line can be handled.



Monitor Type: OPM

Monitor Name: **com.blixx.agent.monitors.Opm**

Executable: sh -c "LC_ALL=C ifconfig"

OPM Type: PunchCard

Split-Pattern: \s*(\w+)\s+Link.*

Object: Interface: <$InterfaceName> = (100 * <$RXErrors> / (<$RXPaketNumber> + 1)) + (100 * <$TXErrors> / (<$TXPaketNumber> + 1))

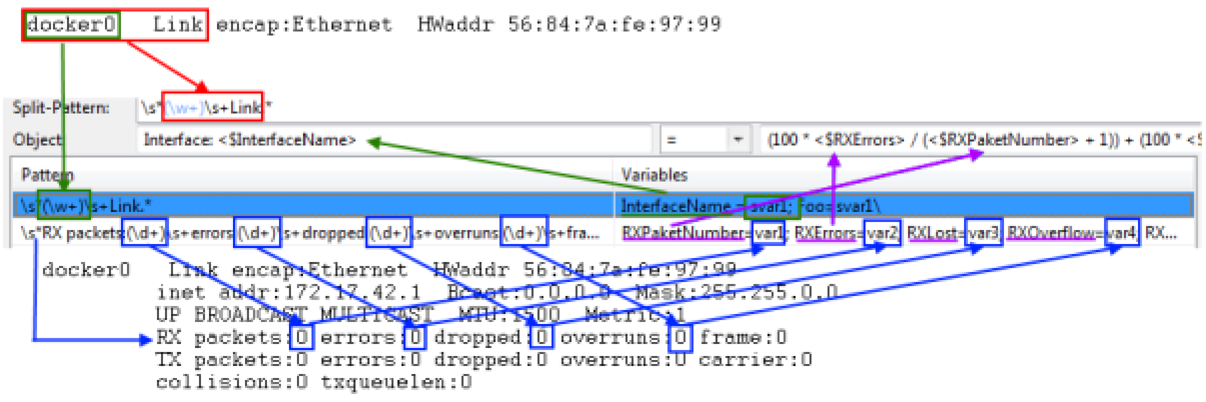
Pattern	Variables
\s*(\w+)\s+Link.*	InterfaceName = svar1
\s*RX packets:(\d+)\s+errors:(\d+)\s+drop...	RXPaketNumber=var1; RXErrors=var2; RXLost=var3; RXOverflow=var4; RXWindow = var5
\s*TX packets:(\d+)\s+errors:(\d+)\s+dropp...	TXPaketNumber=var1; TXErrors=var2; TXLost=var3; TXOverflow=var4; TXCarrier=var5
\s*collisions:(\d+)\s+txqueuelen:(\d+).*	Collisions=var1; SendQueueLength = var2
\s*RX bytes:(\d+)\s+\(\d+\.\d+\s+(?:B|KB|...	RXMBytes= round(var1 / 1024, 2); TXMBytes=round(var2 / 1024, 2)

Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$VALUE>

The ifconfig command returns an output similar to:

```
docker0   Link encap:Ethernet  HWaddr 56:84:7a:fe:97:99
         inet addr:172.17.42.1  Bcast:0.0.0.0  Mask:255.255.0.0
         UP BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
eth0     Link encap:Ethernet  HWaddr 08:00:27:bc:61:76
         inet addr:192.168.123.190  Bcast:192.168.123.255  Mask:255.255.255.0
         inet6 addr: fe80::a00:27ff:febc:6176/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:54360 errors:0 dropped:0 overruns:0 frame:0
         TX packets:10223 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:8575117 (8.5 MB)  TX bytes:837613 (837.6 KB)
lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:367 errors:0 dropped:0 overruns:0 frame:0
         TX packets:367 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:34725 (34.7 KB)  TX bytes:34725 (34.7 KB)
```

The Split-Pattern matches the three lines containing the keyword "Link" and is starting with the interface name as a sequence of characters and numbers. Therefore the Patterns will be matched and the Object evaluated for each of the interfaces:



```
docker0    Link encap:Ethernet  HWaddr 56:84:7a:fe:97:99
```

Split-Pattern: \s*(\w+)\s+Link.*

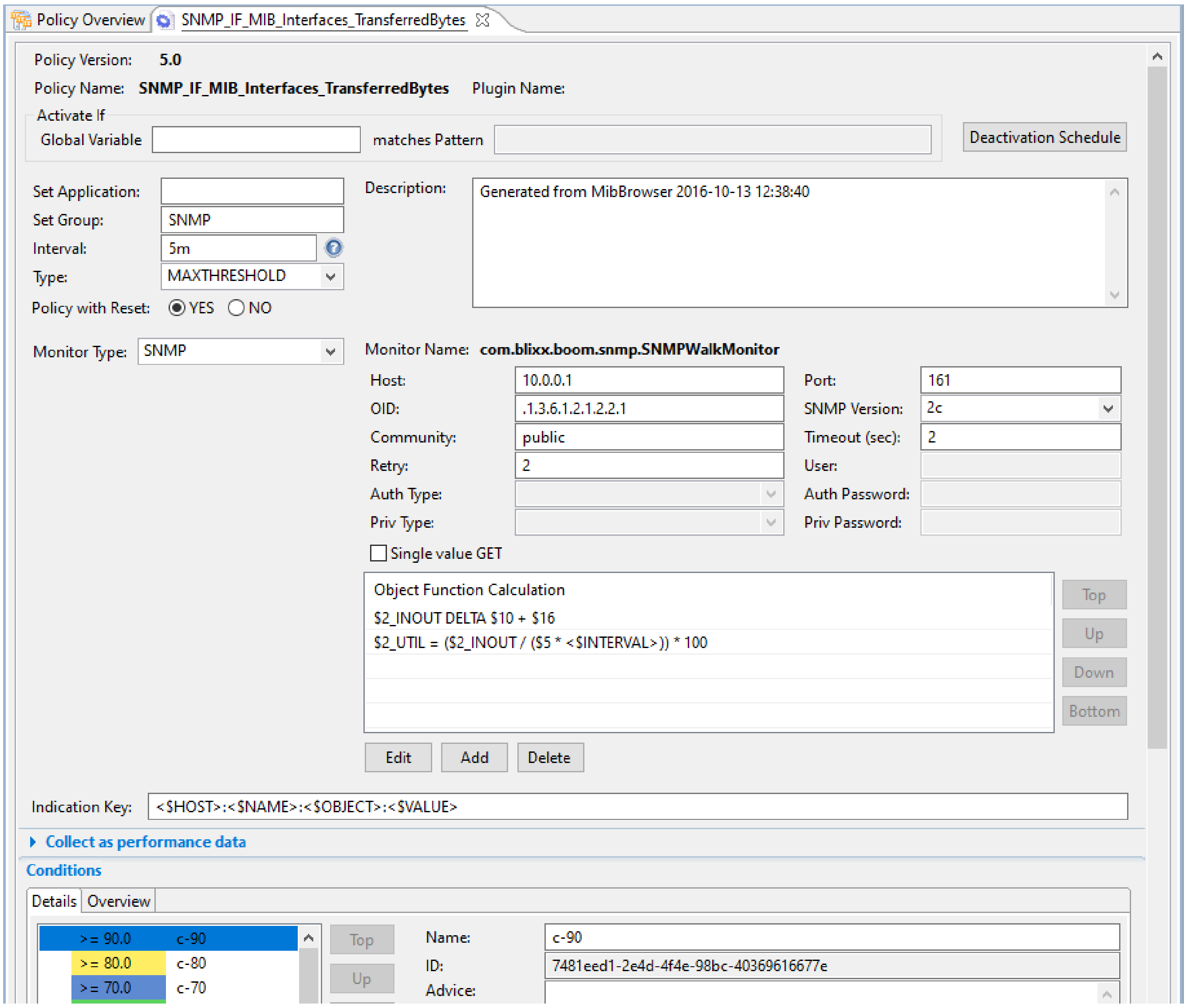
Object: Interface: <$InterfaceName> = (100 * <$RXErrors> / (<$RXPaketNumber> + 1)) + (100 * <$

Pattern	Variables
\s*(\w+)\s+Link.*	InterfaceName = svar1; foo=svar1\
\s*RX packets:(\d+)\s+errors:(\d+)\s+dropped:(\d+)\s+overruns:(\d+)\s+fra...	RXPaketNumber=var1; RXErrors=var2; RXLost=var3; RXOverflow=var4; RX...

```
docker0   Link encap:Ethernet  HWaddr 56:84:7a:fe:97:99
         inet addr:172.17.42.1  Bcast:0.0.0.0  Mask:255.255.0.0
         UP BROADCAST MULTICAST  MTU:1500  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
```

5.1.3. SNMP Monitor

A monitor policy of type SNMP is used for fetching values with SNMP GET or SNMP WALK. These Policies can be easily generated by using the MIB Browser functionality.



Policy Version: **5.0**
Policy Name: **SNMP_IF_MIB_Interfaces_TransferredBytes** Plugin Name:
Activate If
Global Variable matches Pattern

Set Application:
Set Group: SNMP
Interval: 5m
Type: MAXTHRESHOLD
Policy with Reset: [✓ YES] NO
Description: Generated from MibBrowser 2016-10-13 12:38:40

Monitor Type: SNMP
Monitor Name: **com.blixx.boom.snmp.SNMPWalkMonitor**

Host:	10.0.0.1	Port:	161
OID:	.1.3.6.1.2.1.2.2.1	SNMP Version:	2c
Community:	public	Timeout (sec):	2
Retry:	2	User:	
Auth Type:		Auth Password:	
Priv Type:		Priv Password:	

Single value GET

Object Function Calculation
$2_INOUT DELTA $10 + $16
$2_UTIL = ($2_INOUT / ($5 * <$INTERVAL>)) * 100

Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$VALUE>

>= 90.0	c-90
>= 80.0	c-80
>= 70.0	c-70

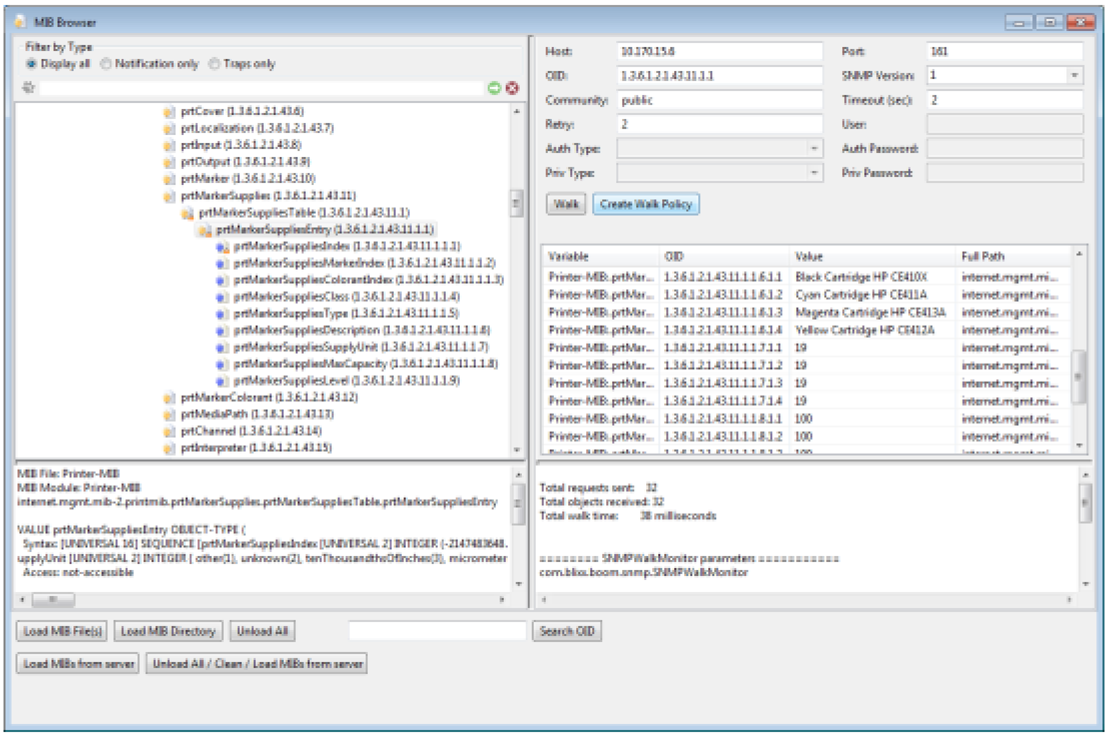
Name: c-90
ID: 7481eed1-2e4d-4f4e-98bc-40369616677e
Advice:

The SNMP Monitor requires the following parameters:

Host	IP Address or agent resolvable hostname of the SNMP Device
Port	Port number on which the SNMP Device listens
OID	OID of the target table or field
Community	Community String that the SNMP Device uses for Authorization (SNMP Version 1, 2c)
Timeout	Timeout in seconds that the agent will wait for the response
Retry	Number of retries before the agent mark the monitor call as failed

SNMP Version	SNMP Version that the agent will use for communication with the SNMP Device (Versions: 1, 2c, 3) Selection of SNMP Version 3 will activate additional fields that allow to specify the credentials and type of Authorization and Privacy that is used for the communication:
User	User Name that is used for Authorization (Community String is not used with SNMP v3)
Auth Type	Authorization Type. The selected type must fit to the configuration on the SNMP Device. A blank field means Type "none", every other value requires to specify an Authorization Passwort.
Auth Password	Password used for Authorization, must be at least 8 characters long.
Priv Type	Privacy Type specifies the encryption method. The selected type must fit to the configuration on the SNMP Device. A blank field means Type "none", every other value requires to specify an Privacy Passwort.
Priv Password	Password used for Privacy, must be at least 8 characters long.
Single value get	Setting this check mark denotes that the queried value is a single value and not a table. If checked the monitor will use a faster method to retrieve the value and it is not required to set an Object Function Calculation Rule.
Object Function Calculation	Specify Calculation Rules to evaluate Object Names and value pairs that will be processed. All rules must have the form of <Object Name>=<Calculation Rule>. These Object names can be used in the Conditions to specify thresholds specific for each object. The object and calculation can use fields of the table that is referenced by the OID simply by selecting the according columns by their number, e.g. $6 for column 6. For details regarding available calculation rules refer to Calculation Syntax for OPM and SNMP Variables.

Example:
Using the MIB Browser we want to check how to query the Cartridge Fill Level of an Printer with SNMP Support:



```
prtCover (1.3.6.1.2.1.43.6)
prtLocalization (1.3.6.1.2.1.43.7)
prtInput (1.3.6.1.2.1.43.8)
prtOutput (1.3.6.1.2.1.43.9)
prtMarker (1.3.6.1.2.1.43.10)
prtMarkerSupplies (1.3.6.1.2.1.43.11)
  prtMarkerSuppliesTable (1.3.6.1.2.1.43.11.1)
    prtMarkerSuppliesEntry (1.3.6.1.2.1.43.11.1.1)
      prtMarkerSuppliesIndex (1.3.6.1.2.1.43.11.1.1.1)
      prtMarkerSuppliesMarkerIndex (1.3.6.1.2.1.43.11.1.1.2)
      prtMarkerSuppliesColorantIndex (1.3.6.1.2.1.43.11.1.1.3)
      prtMarkerSuppliesClass (1.3.6.1.2.1.43.11.1.1.4)
      prtMarkerSuppliesType (1.3.6.1.2.1.43.11.1.1.5)
      prtMarkerSuppliesDescription (1.3.6.1.2.1.43.11.1.1.6)
      prtMarkerSuppliesSupplyUnit (1.3.6.1.2.1.43.11.1.1.7)
      prtMarkerSuppliesMaxCapacity (1.3.6.1.2.1.43.11.1.1.8)
      prtMarkerSuppliesLevel (1.3.6.1.2.1.43.11.1.1.9)
prtMarkerColorant (1.3.6.1.2.1.43.12)
prtMediaPath (1.3.6.1.2.1.43.13)
prtChannel (1.3.6.1.2.1.43.14)
prtInterpreter (1.3.6.1.2.1.43.15)
```

Host:	10.170.15.6	Port:	161
OID:	1.3.6.1.2.1.43.11.1.1	SNMP Version:	1
Community:	public	Timeout (sec):	2
Retry:	2	User:	
Auth Type:		Auth Password:	
Priv Type:		Priv Password:	

Variable	OID	Value	Full Path
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.6.1.1	Black Cartridge HP CE410X	internet.mgmt.mi...
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.6.1.2	Cyan Cartridge HP CE411A	internet.mgmt.mi...
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.6.1.3	Magenta Cartridge HP CE413A	internet.mgmt.mi...
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.6.1.4	Yellow Cartridge HP CE412A	internet.mgmt.mi...
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.7.1.1	19	internet.mgmt.mi...
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.7.1.2	19	internet.mgmt.mi...
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.7.1.3	19	internet.mgmt.mi...
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.7.1.4	19	internet.mgmt.mi...
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.8.1.1	100	internet.mgmt.mi...
Printer-MIB.prtMar...	1.3.6.1.2.1.43.11.1.1.8.1.2	100	internet.mgmt.mi...

```
MIB File: Printer-MIB
MIB Module: Printer-MIB
internet.mgmt.mib-2.printmib.prtMarkerSupplies.prtMarkerSuppliesTable.prtMarkerSuppliesEntry

VALUE prtMarkerSuppliesEntry OBJECT-TYPE (
  Syntax: [UNIVERSAL 16] SEQUENCE [prtMarkerSuppliesIndex [UNIVERSAL 2] INTEGER (-2147483648.
upplyUnit [UNIVERSAL 2] INTEGER { other(1), unknown(2), tenThousandthsOfInches(3), micrometer
  Access: not-accessible
```

```
Total requests sent:    32
Total objects received: 32
Total walk time:        38 milliseconds


========= SNMPWalkMonitor parameters ============
com.blos.boom.snmp.SNMPWalkMonitor
```

After navigating to the standard Printer MIB we can browse the actual values and create the according policy directly from the MIB browser.

The returned prtMarkerSuppliesTable returns a table that contains as 6th column the supply description e.g. Black Cartridge, column 9 gives the actual level and column 8 the total capacity. In the above policy we want to monitor the remaining percentage for each available supply. The Object Function Calculation `$6=round($9 / $8 * 100, 2)` calculates this for every entry in the table and uses the description as object name. It returns e.g.
`"Black Cartridge xxxx"=38,42`
`"Cyan Cartridge xxxx"=45,77`
...

5.1.4. Calculation Syntax for OPM and SNMP Variables

Calculation Syntax

The calculation fields contain variables and values of data type double and arithmetic operations. All elements have to be separated by white space characters (e.g. var1 + var2 / 100).

Syntax of Calculation

'... X1 'operand1' X2 'operand2' X3 'operand3' X4 ...'

```
Xn:        value or variable (varN) of data type double.
operandN:  one of '+', '-', '/' ,'*' (trivial arithmetical operations)
           and special operand '\' Operand '\' - means a "backward" division
           of Xi/Xi-1 e.g. RES = var2 \ var3 is calculated as var3/var2
```

The example above will be calculated in sequence left to right:

```
((X1 'operand1' X2) 'operand2' X3) 'operand3' X4
```

Starting with *boom* version 5 there are extended calculations possible. The simple operations are still supported for backward compatibility.
For compatibility reasons the sequence of operations (no precedence of multiplication and division) is kept as before as well as the backward division by "\". To override the sequence of processing use the brackets to indicate

precedence:

```
((var1 + var2) / (var3 - var4)) * 100
```

Operation	Description
^	Power Example: 2^3 = 8
\	Back division (second argument divided by the first one) Example: 4 \ 8 = 2
/	Division Example: 8 / 4 = 2
-	Subtraction or unary minus. Examples: -5 = -5 5 - 2 = 3
+	Addition, or unary plus. Examples: +5 = 5 2 + 2 = 4
*	Multiplication Example: 2 * 3 = 6
%	Modulo Example: 5 % 2 = 1

Function	Description
floor(x, y)	Floor of x with precision y. Equivalent to floor(x * 10^y) / 10^y Examples: floor(1.234567, 0) = 1 floor(1.234567, 2) = 1.23 floor(1234.567, -2) = 1200

Function	Description
ceil(x ,y)	Ceil of x with precision y. Equivalent to ceil(x * 10^y) / 10^y Examples: ceil(1.234567, 0) = 2 ceil(1.234567, 1) = 1.3 ceil(1234.567, -1) = 1240
round(x, y)	Returns the closest value to the first argument with precision y, with ties rounding up. Examples: round(1.234567, 0) = 1 round(1.5, 0) = 2 round(1.234567, 2) = 1.23 round(1.234567, 3) = 1.235 round(1234.567, -2) = 1200 round(1262.567, -2)= 1300
abs(x)	Absolute value of x
cbrt(x)	Cube root of x
exp(x)	Euler's number e raised to the power of the argument: e^x
expm1(x)	Returns (e^(x)) - 1
pow(x, y)	Value of the first argument raised to the power of the second argument: x^y
sqrt(x)	Positive square root of x
cos(x)	Trigonometric cosine of an angle.
sin(x)	Trigonometric sine of an angle.
tan(x)	Trigonometric tangent of an angle.
log(x)	Natural logarithm (base e) of x
log2(x)	Base 2 logarithm of x
log10(x)	Base 10 logarithm of x
log1p(x)	Natural logarithm of the sum of the argument and 1.
acos(x)	Arc cosine of a value; the returned angle is in the range 0.0 through pi.
asin(x)	Arc sine of a value; the returned angle is in the range -pi/2 through pi/2.
atan(x)	Arc tangent of a value; the returned angle is in the range -pi/2 through pi/2.
cosh(x)	Hyperbolic cosine of x
sinh(x)	Hyperbolic sine of x
tanh(x)	Hyperbolic tangent of x

For the following examples we assume that the filtered string is parsed into these variables:

```
var1 - /dev/vg00/lvol5
var2 - 24456
var3 - 8472
var4 - 15984
var5 - 35
var6 - /home.
```

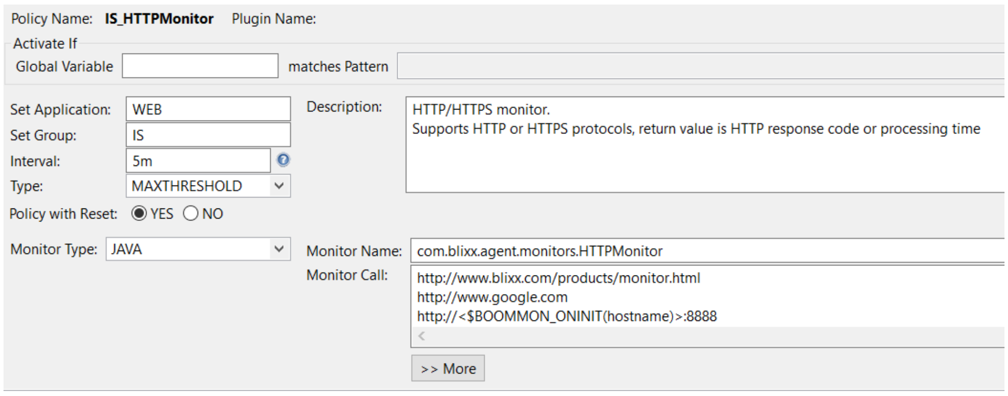
Examples of Object Function Calculations (LineByLine)	Description
FREEDISK = var4	Monitor value = var4 = 15984.0 Object = FREEDISK
svar1 = var4	Monitor value = var4 = 15984.0 Object = /dev/vg00/lvol5
svar6 = round(var4 / 1024, 2)	Monitor value = (var4/1024) = 15.61 (rounded to two digits) Object = /home
svar1 = 100 - var5	Monitor value = (100 - var5) = 75.0 Object = /dev/vg00/lvol5

Examples of Object Function Calculation (TableSummary)	Description
FREEDISK_T SUM var4	Sum of all free spaces Object = FREEDISK_T
FREEDISK_A AVG var2 - var3 /1024	Average of all free spaces in MB calculated as (var2 - var3)/1024 Object = FREEDISK_A
FREEDISK_M MIN var4 /1024	Minimum free space in MB Object = FREEDISK_M

Examples of Object Function Calculation (PunchCard)	Description
(<$TOTAL> - <$FREE>) * 100 / <$TOTAL>	Usage Percentage calculated by variables defined in the pattern section Object = can be a fixed name or constructed with a string variable defined in the pattern section
<$TOTAL> - <$FREE> / 1024	Used Space or Memory in MB

5.1.5. Java Monitor

A monitor policy of monitor type Java lets you trigger special Java class executables which return one or more values to be processed by the defined conditions. It expects a Java monitor deployed to the *boom* agent.



Policy Name: **IS_HTTPMonitor** Plugin Name:
Activate If
Global Variable matches Pattern
Set Application: WEB
Set Group: IS
Interval: 5m
Type: MAXTHRESHOLD
Policy with Reset: [✓ YES] NO
Description: HTTP/HTTPS monitor.
Supports HTTP or HTTPS protocols, return value is HTTP response code or processing time
Monitor Type: JAVA
Monitor Name: com.blixx.agent.monitors.HTTPMonitor
Monitor Call:
```
http://www.blixx.com/products/monitor.html
http://www.google.com
http://<$BOOMMON_ONINIT(hostname)>:8888
```

Following fields have to be configured for the monitor type **"JAVA"**:

Monitor Name field

this field has to contain the Java class name to be executed.

Monitor Call field

this field contains any additional commands and/or parameters used by the Java class to work/respectively to deliver the monitor values.

Following monitoring functions are included in the com.blixx.agent.monitors Java class and can be executed in a monitor policy with type "JAVA". Available parameters and return values are described in the pre-installed sample policies.

Class Name	Description	Sample Policy
com.blixx.agent.monitors.DNSMonitor	DNS monitor (AVAILABILITY and TIME monitor) Can query either specified or locally available DNS servers (auto-detected). Supports TCP/UDP modes, reverse and forward lookup requests.	IS_DNSMonitor
com.blixx.agent.monitors.FTPSMonitor	FTP/FTPS monitor (AVAILABILITY and TIME monitor)	IS_FTPMonitor
com.blixx.agent.monitors.HTTPMonitor	HTTP/HTTPS monitor. Supports HTTP or HTTPS protocols, return value is HTTP response code or processing time	IS_HTTPMonitor
com.blixx.agent.monitors.IMAPMonitor	IMAPmonitor (AVAILABILITY and TIME monitor)	IS_IMAPMonitor
com.blixx.agent.monitors.IMAPSMonitor	IMAPS monitor (AVAILABILITY and TIME monitor) TLSv1+SSLv3	IS_IMAPSMonitor
com.blixx.agent.monitors.LDAPMonitor	LDAP monitor (AVAILABILITY and TIME monitor)	IS_LDAPMonitor
com.blixx.agent.monitors.NNTPMonitor	NNTP monitor (AVAILABILITY and TIME monitor)	IS_NNTPMonitor
com.blixx.agent.monitors.NTPMonitor	NTP monitor (AVAILABILITY and TIME monitor)	IS_NTPMonitor
com.blixx.agent.monitors.POP3Monitor	POP3 monitor (AVAILABILITY and TIME monitor)	IS_POP3Monitor

Class Name	Description	Sample Policy
com.blixx.agent.monitors.RadiusMonitor	Radius monitor (AVAILABILITY and TIME monitor)	IS_RadiusMonitor
com.blixx.agent.monitors.SFTPMonitor	SFTP over SSH monitor (AVAILABILITY and TIME monitor)	IS_SFTPMonitor
com.blixx.agent.monitors.SMTPMonitor	SMTP monitor (AVAILABILITY and TIME monitor)	IS_SMTPMonitor
com.blixx.agent.monitors.SSHMonitor	SSH monitor (AVAILABILITY and TIME monitor)	IS_SSHMonitor
com.blixx.agent.monitors.TelnetMonitor	Telnet monitor (AVAILABILITY and TIME monitor)	IS_TelnetMonitor
com.blixx.boom.snmp.SNMPWalkMonitor	SNMP Walk Monitor	Sample_SNMP_MIB_Monitor_Printer Sample_SNMP_Linux_Mem_Swap Sample_SNMP_Linux_CPU SNMPwalk

5.1.6. Exec Monitor

A monitor policy with monitor type "EXEC" triggers a script or command (available on the *boom* agent), which returns one or more values to the monitor to be processed by the defined condition(s). This script or command has to be set up in the field **"Monitor Call"**. It is the responsibility of the monitoring script to collect the current value of the monitored object and report it to *boom*.
If the script or executable is located outside the agent's binary package path the call can be configured with the full system path or needs to be prefixed with a '#' sign. The '#' instructs the agent to use the system search path to locate the according script or binary i.e.

***testscript.sh** <$NAME>*

if testscript.sh resides in the binary package path `<boom agent install dir>/spi`

***testscript.sh** <$NAME>*

if testscript.sh is available in some directory of the system search path (PATH variable)

Be aware of parameters that are evaluated by the shell and not the command itself. Some parameters like the * in path or file names, pipes, redirections, command concatenations are processed by the shell i.e.

ls -l ../BOOM*

will fail with the error that the file `../BOOM*` was not found

sh -c "ls -l ../BOOM*"

will work as expected and deliver back all files with names starting with BOOM. The shell processes the * and passes the results to the ls command.

For windows the same behavior applies but the agent will automatically execute the specified call with `cmd /c "<command> <parameter>"`

The parameter <$NAME> is replaced by the agent with the monitor policy name which is required later on to return the monitor value.

There are two ways to deliver the monitor values to the according monitor policies:

- On the one hand the monitor value can be submitted by the *boom* command **"boommon"** which is shipped with the agent. This command is called by the defined script and has to be used in the following way:

boommon <**MonitorName**>=<**value**> [(o|object)=<object>] [varName=value]

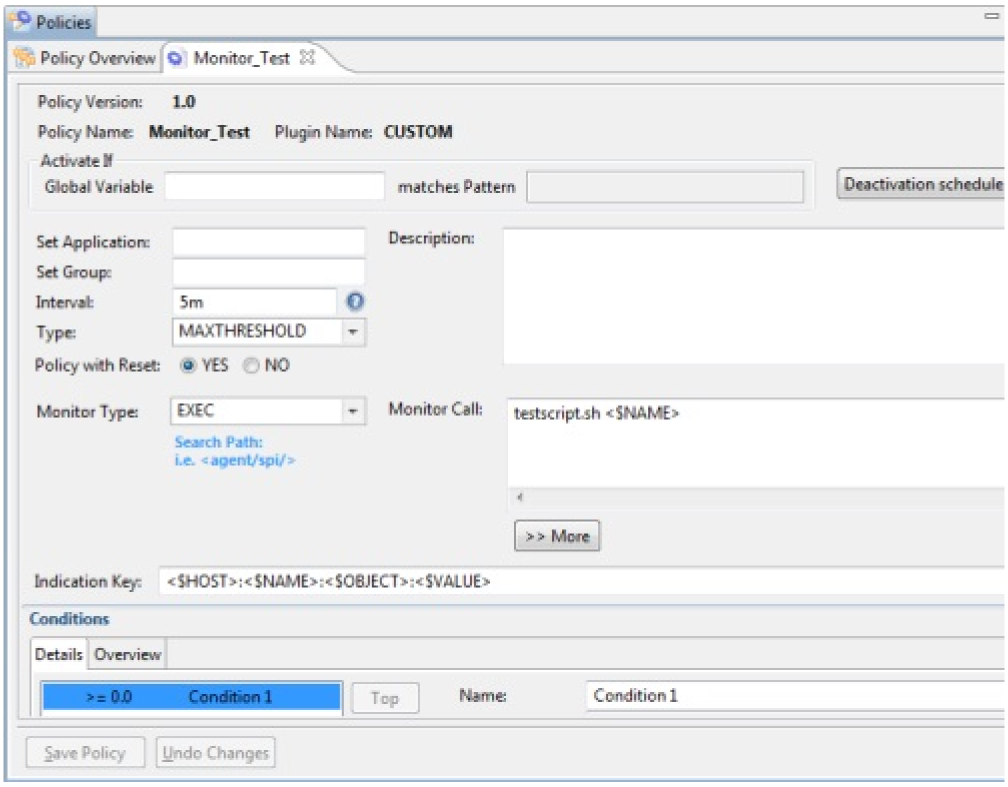
MonitorName:
is the name of the monitor policy. An object and custom variables can be sent optionally.

- On the other hand, the output of the executed script consists of the following:

  STDOUT:<MonitorName>=<value> [(o|object)=objectName] [varName=varValue]

  MonitorName:
  is the name of the monitor policy. It is also possible to create an output which contains multiple output lines with the mentioned content for more than one monitor policy.

If "boommon" is used, it is possible to receive values for different objects and filter the boommon values by using the object filter in the condition properties.



Policy Version: 1.0
Policy Name: **Monitor_Test** Plugin Name: **CUSTOM**
Activate If
Global Variable matches Pattern
Set Application:
Set Group:
Interval: 5m
Type: MAXTHRESHOLD
Policy with Reset: [✓ YES] NO
Description:
Monitor Type: EXEC
Search Path:
i.e. <agent/spi/>
Monitor Call: testscript.sh <$NAME>
Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$VALUE>
>= 0.0 Condition 1
Name: Condition 1

The agent sets the following environment variables that can be used by the scripts or executables:

BOOM_AGENT_HOST
the agents host name.

BOOM_AGENT_ID
the agent’s unique ID (e.g. 07f24f82-2d44-4c4c-98c0-1ac7e3814a77).

BOOM_AGENT_IP
the agent’s IP address.

BOOM_AGENT_PID
the agent’s process id.

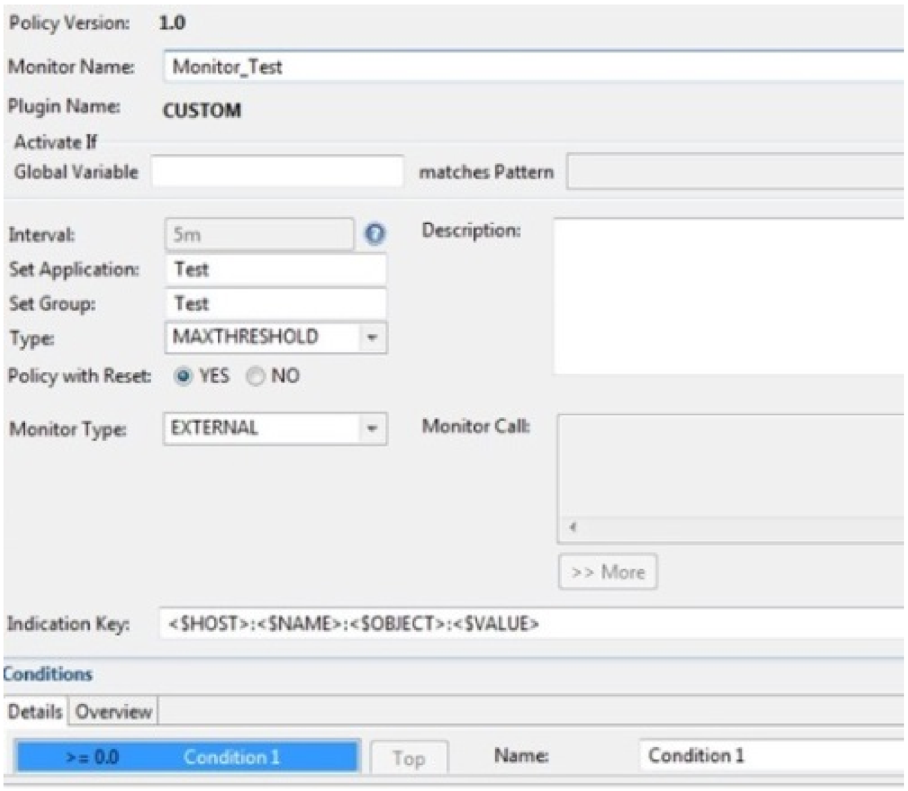
BOOM_AGENT_PORT
the port number the agent is listening on (i.e. 23021).

5.1.7. External Monitor

The monitor policy of type "EXTERNAL" works in the same way as the monitor of type "EXEC" as described above with two exceptions.

This monitor does not trigger a monitor script by itself. The "EXTERNAL" monitor is waiting for a value which is sent by an external script or program using the *boom* command **"boommon"** as described for monitor type "EXEC".

This monitor is not able to receive monitor values only through the output of a script as the "EXEC" monitor does because the script is not triggered by itself.



Policy Version: 1.0
Monitor Name: Monitor_Test
Plugin Name: CUSTOM
Activate If Global Variable: matches Pattern:
Interval: 5m
Description:
Set Application: Test
Set Group: Test
Type: MAXTHRESHOLD
Policy with Reset: [✓ YES] NO
Monitor Type: EXTERNAL
Monitor Call:
Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$VALUE>
Conditions
Details Overview
>= 0.0 Condition 1
Name: Condition 1

5.2. Indication Policy (Event triggered/Text related monitoring)

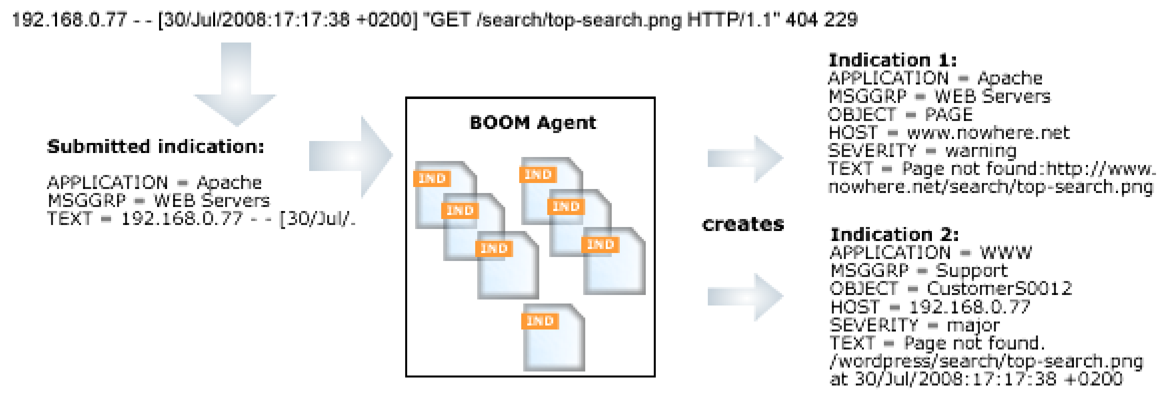
5.2.1. General Information

The second major concept of the *boom* infrastructure is an indication. An indication is an incoming message that describes an event which occurred on the monitored node but has additional attributes like severity, message group, application, object, etc…

In a general picture the monitor value also produces an indication on the *boom* server. This indication states that the threshold level for the monitored metric has been changed.
Any monitoring probe or external application can submit a message to the *boom* agent. The *boom* agent uses indication policies to filter the submitted messages in order to prevent the server from a "message storm". Indication policies contain all necessary information to suppress unimportant or useless messages and allow parsing and constructing new indications in a form that can be used by the administrator or operator.

The results of flexibility inside indication policies are:

- reducing the number of messages that are displayed in the *boom* GUI * the possibility to extract useful parts of the Indication text for remote actions or
- adjusting the Indication itself.



```
192.168.0.77 - - [30/Jul/2008:17:17:38 +0200] "GET /search/top-search.png HTTP/1.1" 404 229
```

The indication correlation engine is the main component which is responsible for processing the incoming messages based on a set of loaded policies. All incoming indications shall be compared against all policies. If multiple policies have matched conditions for one incoming indication, the agent will produce multiple result indications. An exception is a policy that matches all messages (in other words "forwards all messages"). In case one of the other policies has generated an indication the "forwards all messages" policy will be ignored.

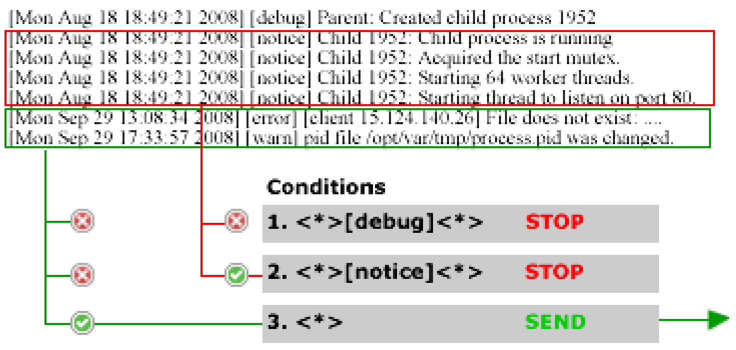
All hybrid policies which will be explained in the next chapter, are excluded from general correlations. The hybrid policies are only working with indications that are submitted by triggers. The triggers must be defined inside the hybrid policy.

The "Application" and "Indication Group" attributes of an indication policy are used as main pre-filtering condition. If these attributes are specified in the policy and the submitted indication contains non empty values, than the agent is able to process a reduced set of policies and increase the speed of the processing.

Conditions

The process of matching incoming indications to an Indication Policy is based on the filter conditions that are defined inside the Policy. The first condition that has been matched stops the processing of all following conditions.

There are two major types of conditions: Conditions that create an Indication and conditions that "STOP" the processing.



```
[Mon Aug 18 18:49:21 2008] [debug] Parent: Created child process 1952
[Mon Aug 18 18:49:21 2008] [notice] Child 1952: Child process is running
[Mon Aug 18 18:49:21 2008] [notice] Child 1952: Acquired the start mutex.
[Mon Aug 18 18:49:21 2008] [notice] Child 1952: Starting 64 worker threads.
[Mon Aug 18 18:49:21 2008] [notice] Child 1952: Starting thread to listen on port 80.
[Mon Sep 29 13:08:34 2008] [error] [client 15.124.140.26] File does not exist: ....
[Mon Sep 29 17:33:57 2008] [warn] pid file /opt/var/tmp/process.pid was changed.
```

Such *STOP Conditions* allow cutting off Indications that are of no interest. For example: A logfile usually contains a big number of informative messages with normal severity. For problem detection and monitoring only warnings and errors should be delivered to an operator. To cut off unnecessary messages and processing, you can use the STOP Condition as the top condition (first condition in the list) in the indication policy.

Patterns and Usage

One of the most powerful and most complicated concepts of indication policies is the pattern concept. Patterns are used for filtering incoming indications to identify the matching policy conditions. By default the 'Simplified Patterns' are used for filtering attributes.

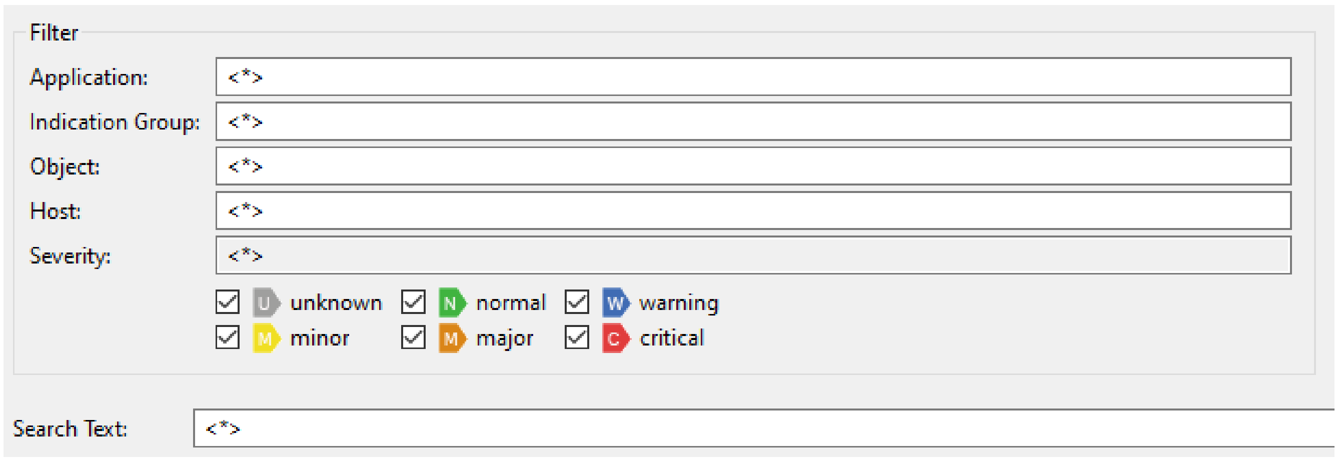
In addition the **"Search Text"** filter can have a fully featured java pattern, because the syntax of 'Simplified Pattern' does sometimes not allow handling complex situations. The Java format of a pattern requires the prefix "java=". I.e **java**=Error:.*

As described later, the UI includes a powerful pattern editor that makes the generation and validation of patterns very easy.
Any indications have the following attributes:

- Application (required)
- Indication Group
- Object
- Host
- Severity
- Text (required)

These values are all used by the filtering process.

An indication must provide at least the "Application" and "Text" attributes. All other attribute values are optional. The default value for all filtering fields in a new Indication Policy is the **"MATCH_ALL" condition**:



Filter

Application:	<*>
Indication Group:	<*>
Object:	<*>
Host:	<*>
Severity:	<*>

✓ unknown ✓ normal ✓ warning
✓ minor ✓ major ✓ critical

Search Text: <*>

This means that a new created policy will process all incoming messages. It is necessary to specify patterns that match only messages that should be processed by this policy.

Application	Group	Object	Host	Severity	Text	Matches
<*>	<*>	<*>	<*>	<*>	<*>	all (MATCH_ALL condition)
APPL1	<*>	<*>	<*>	<*>	<*>	indications with Application="APPL1"
APPL1 | APPL2	<*>	<*>	<*>	<*>	<*>	indications with Application "APPL1" OR "APPL2"
<*>	GRP1 | G2 | G3	OBJECT_<*>	<*>	<*>	<*>	Group ("GRP1" OR "G2" OR "G3") AND Object starts with "OBJECT_"

Application	Group	Object	Host	Severity	Text	Matches
APACHE|Tomcat	WEB	MEMORY	<*>	critical|major	<*>	Application ("Apache" OR "Tomcat") AND Group="WEB" AND Object="MEMORY" AND Severity (major OR critical)

For the "Application", "Indication Group", "Object" and "Host" fields it is allowed to have **simple "OR" notation** like:

APPL1|APPL2|APPL3 instead of [APPL1|APPL2|APPL3]

It is recommended to have only one condition across all policies that match all indications.

It is recommended to have "Application" & "Indication Group Filters" specified with the exact value or list of the expected values by using OR notation. This will reduce time of the processing and increase performance.

5.2.2. boomindi

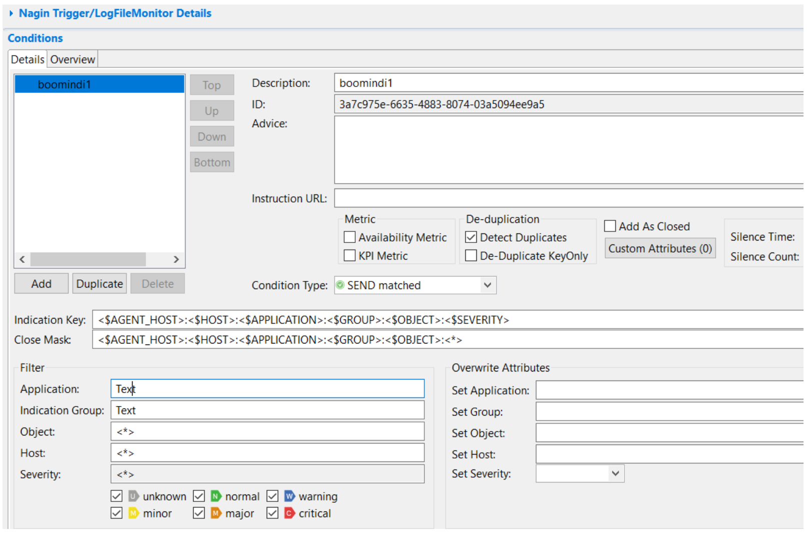
A "boomindi" policy processes messages which were sent to the agent by the boomindi command. The boomindi command is located in the $BOOM_ROOT/spi directory and can be used e.g. in any script. It is shipped with the agent.
For a correct working of the policy it is important **not** to set up a trigger by activating the "Nagin Trigger/LogFileMonitor Details" section.

The boomindi command has to be used as follows:

```
./boomindi (a|application)=<application> [(o|object)=<object>] [(g|group)=<group>]
[(s|severity)=<severity>] [(h|host)=<host>] [varName=value]* (t|text)=<"text">
```

To filter boomindi messages with the policy, any fields in the "Filter" section of a condition can be set up with strings and patterns. In addition it is possible to use the fields **"Search Text"** and **"Match Variables"** to filter boomindi messages/ indications. In the field "Match Variables", variables set by the parameter varName=value can be addressed and filtered and in the field **"Search Text"** substrings and patterns can be defined to filter these messages.

The "Match Variables" and "Search Text" fields can also be used to construct text from variables and message texts and define match conditions.



Description: boomindi1
ID: 3a7c975e-6635-4883-8074-03a5094ee9a5
Advice:
Instruction URL:
Metric: ☐ Availability Metric ☐ KPI Metric
De-duplication: ☑ Detect Duplicates ☐ De-Duplicate KeyOnly
☐ Add As Closed
Custom Attributes (0)
Silence Time:
Silence Count:
Condition Type: SEND matched
Indication Key: <$AGENT_HOST>:<$HOST>:<$APPLICATION>:<$GROUP>:<$OBJECT>:<$SEVERITY>
Close Mask: <$AGENT_HOST>:<$HOST>:<$APPLICATION>:<$GROUP>:<$OBJECT>:<*>

Filter
Application: Text
Indication Group: Text
Object: <*>
Host: <*>
Severity: <*>
☑ unknown ☑ normal ☑ warning
☑ minor ☑ major ☑ critical

Overwrite Attributes
Set Application:
Set Group:
Set Object:
Set Host:
Set Severity:

5.2.3. Hybrid Policy

This type of policy allows to periodically trigger any external script, application or internal Java Monitor. It also allows creating indications based on the produced text output. Nagios® compatible plug-ins are working this way. According to the Nagios® definitions, the output of the monitor must return a result code that indicates the severity and writes a detailed message and optional performance data to the STDOUT.

This kind of plug-ins is supported by the *boom* agent (except the handling of the performance data). Other command line tools, scripts or binaries can also be used as triggers. The hybrid policy supports also Java based triggers. Java based triggers must submit an indication by implementing the exposed interfaces or using the **boomindi** command supplied with the *boom* agent.

The main difference of this policy is the process work flow. New boom indications are generated based on the conditions inside the hybrid policy. These conditions are filtering the output produced by the trigger similar to the concepts described for the indication policies. In contrast to the indication policies, only the conditions of this one hybrid policy are used to process the trigger's output. Also messages coming from other sources than this trigger will not be processed with these policy conditions.

There are several Hybrid Indication Policy **Monitor Types** available:

- JAVA
- NAGIN
- Logfile Monitor
- Logfile Transaction Monitor
- Logfile Transaction Monitor (multiline)
- (MP) Logfile Monitor
- (MP) Logfile Transaction Monitor
- (MP) Logfile Transaction Monitor (multiline)

5.2.3.1. Logfile Monitors

One of the most used features of all the monitoring solutions is the logfile monitoring. This function is implemented as a hybrid policy with Java monitor.

There are several Hybrid Indication Policy **LogFile Monitor Types** available:

- Logfile Monitor
- Logfile Transaction Monitor
- Logfile Transaction Monitor (multiline)
- (MP) Logfile Monitor
- (MP) Logfile Transaction Monitor
- (MP) Logfile Transaction Monitor (multiline)

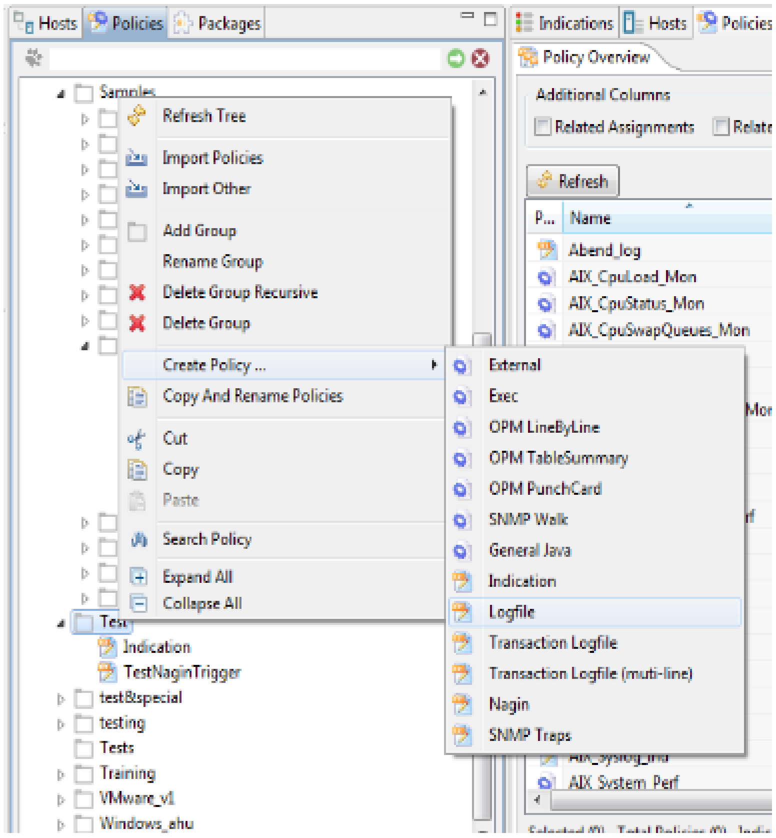
It is **recommended to use the Multipath (MP) Logfile** family instead of the "normal" monitors. The "normal" monitors are kept for backwards compatibility.

One of the most used features of all the monitoring solutions is the logfile monitoring. This function is implemented as a hybrid policy with Java monitor.

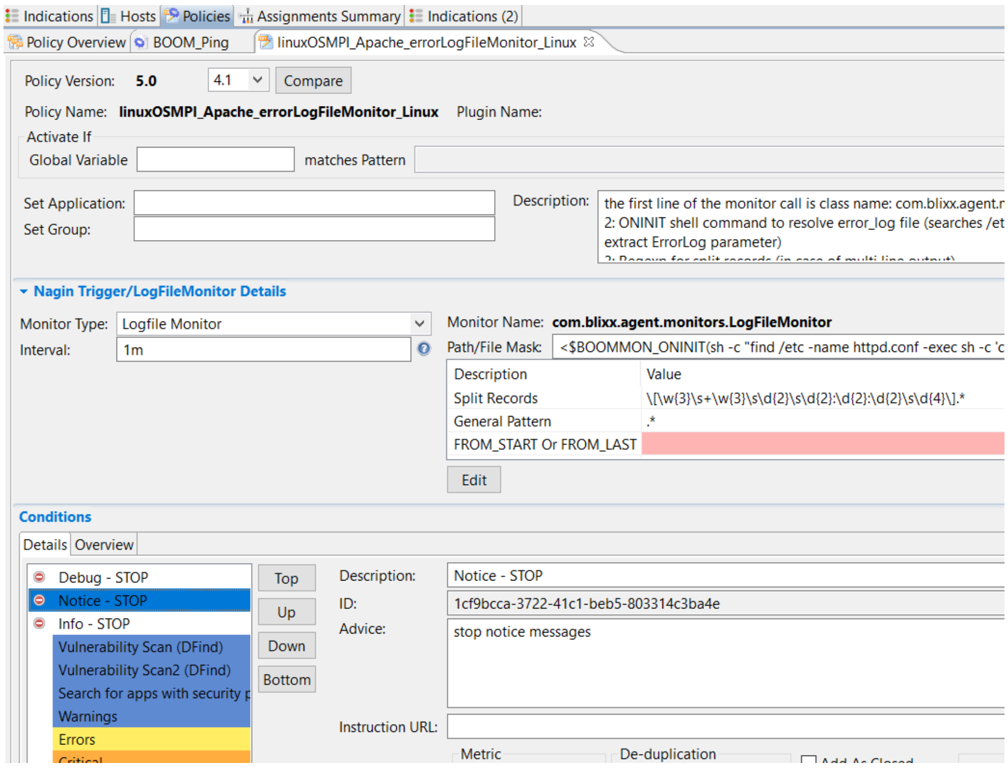
5.2.3.1.1. Logfile Monitor

One of the most used features of all the monitoring solutions is the logfile monitoring. This function is implemented as a hybrid policy using a Java monitor. The implementation of the LogFileMonitor has flexible possibilities to

specify the format of the monitored logfile and pre-process filter. The monitor automatically handles the truncation of the monitored file. It supports file masks for finding rotating log files.
When creating a new log LogFileMonitor Policy, select the type Logfile.



In the Details section select the type of Logfile Monitor that will be used, the polling interval and the other parameters as specified below.



Policy Version: 5.0 4.1

Policy Name: linuxOSMPI_Apache_errorLogFileMonitor_Linux Plugin Name:

Activate If
Global Variable matches Pattern

Set Application:
Set Group:

Description: the first line of the monitor call is class name: com.blixx.agent.
2: ONINIT shell command to resolve error_log file (searches /et
extract ErrorLog parameter)

Nagin Trigger/LogFileMonitor Details

Monitor Type: Logfile Monitor
Interval: 1m

Monitor Name: com.blixx.agent.monitors.LogFileMonitor
Path/File Mask: <$BOOMMON_ONINIT(sh -c "find /etc -name httpd.conf -exec sh -c 'c

Description	Value
Split Records	\[\w{3}\s+\w{3}\s\d{2}\s\d{2}:\d{2}:\d{2}\s\d{4}\].*
General Pattern	.*
FROM_START Or FROM_LAST	🟥

Conditions

Debug - STOP
Notice - STOP
Info - STOP
Vulnerability Scan (DFind)
Vulnerability Scan2 (DFind)
Search for apps with security
Warnings
Errors

Description: Notice - STOP
ID: 1cf9bcca-3722-41c1-beb5-803314c3ba4e
Advice: stop notice messages
Instruction URL:

Monitor Type

Logfile Monitor/Logfile Transaction Monitor - are there for backward compatibility reasons. Please use the Multipath (MP) versions of the Monitor.

(MP) Logfile Monitor - the standard logfile monitor
(MP) Logfile Transaction Monitor - specialized monitor that allows to ensure that certain expected messages show up in the logfile.

Path/File Mask

The path of the logfile that should be monitored. Instead of a fixed path a script can be specified with the BOOMMON_ONINIT parameter that returns the logfile name(s) or a path mask can be specified.

Split Records

Many of the log files contain multi-line entries. It is necessary to have here a Java pattern that is matching the start of the message. Often this is a fixed prefix containing the date, severity, log level, etc.
The default pattern ".*" simply uses the existing line endings (i.e. the logfile is not expected to have multi-line entries)

General Pattern

A Java pattern serving as a pre-process filter. This filter allows to specify a pre-selection to reduce the number of logfile entries that are matched to all the conditions. The pattern ".*" matches all lines - that means all entries will be processed.

FROM_START Or FROM_LAST

FROM_START - indicates that logfile has to be read from the begin on every scheduled interval
FROM_LAST - is the default processing mode. Logfile will be scanned from the last processed point.

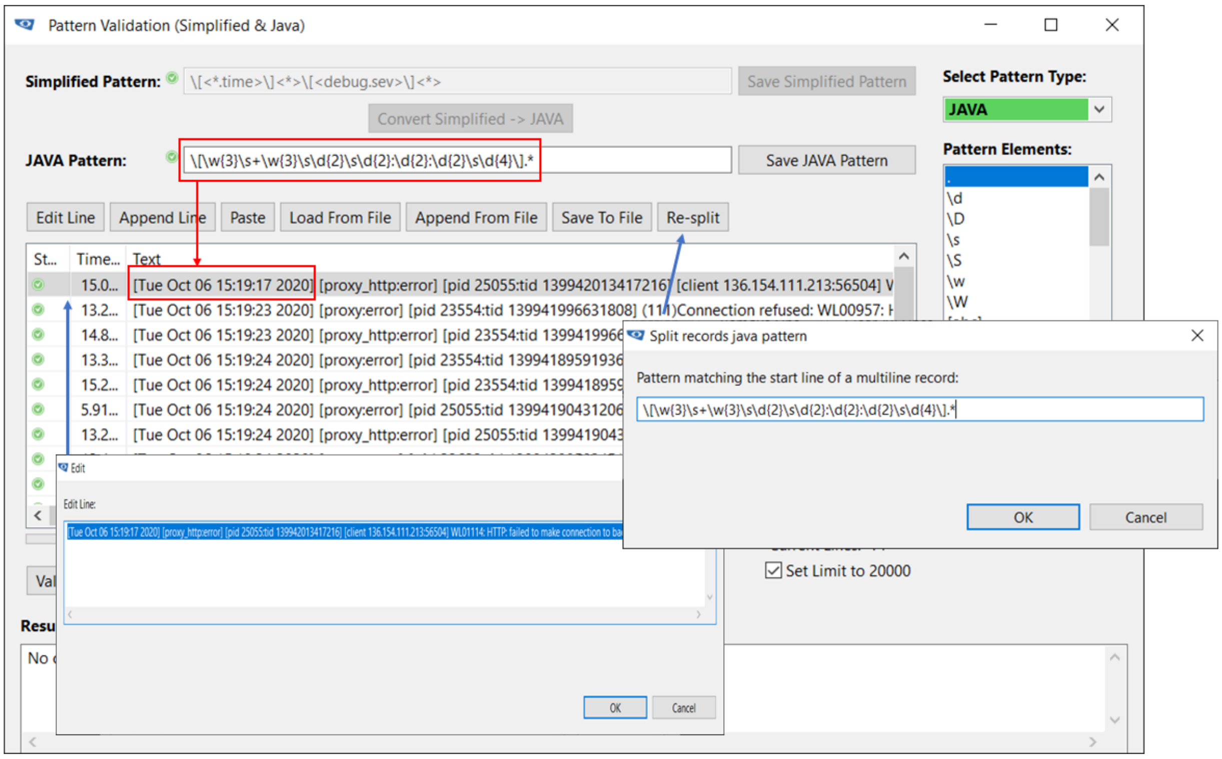
Executable Call (optional)

Allows to specify a trigger that needs to be executed before parsing the logfile.
This can be used together with FROM_START parameter in line 5 to process logfiles re-generated by trigger on every polling interval.

Example for the "Split Records" pattern of the Apache ErrorLogfile Policy:

```
\[\w{3}\s+\w{3}\s\d{2}\s\d{2}:\d{2}:\d{2}\s\d{4}\].*
```

The pattern can be looking a bit terrifying at the first glance. You can easily create and adjust such patterns by using the Pattern Validation Dialog which is integrated in the *boom* GUI (see screenshot below). Fetch an existing logfile from the managed node and load it into the dialog to have a representative set of test data.



Simplified Pattern: `\[<*.time>\]<*>\[<debug.sev>\]<*>`

JAVA Pattern: `\[\w{3}\s+\w{3}\s\d{2}\s\d{2}:\d{2}:\d{2}\s\d{4}\].*`

Select Pattern Type: JAVA

```
15.0... [Tue Oct 06 15:19:17 2020] [proxy_http:error] [pid 25055:tid 139942013417216] [client 136.154.111.213:56504] V
13.2... [Tue Oct 06 15:19:23 2020] [proxy:error] [pid 23554:tid 139941996631808] (111)Connection refused: WL00957: H
14.8... [Tue Oct 06 15:19:23 2020] [proxy_http:error] [pid 23554:tid 1399419966
13.3... [Tue Oct 06 15:19:24 2020] [proxy:error] [pid 23554:tid 13994189591936
15.2... [Tue Oct 06 15:19:24 2020] [proxy_http:error] [pid 23554:tid 1399418959
5.91... [Tue Oct 06 15:19:24 2020] [proxy:error] [pid 25055:tid 13994190431206
13.2... [Tue Oct 06 15:19:24 2020] [proxy_http:error] [pid 25055:tid 1399419043
```

Pattern matching the start line of a multiline record:

`\[\w{3}\s+\w{3}\s\d{2}\s\d{2}:\d{2}:\d{2}\s\d{4}\].*`

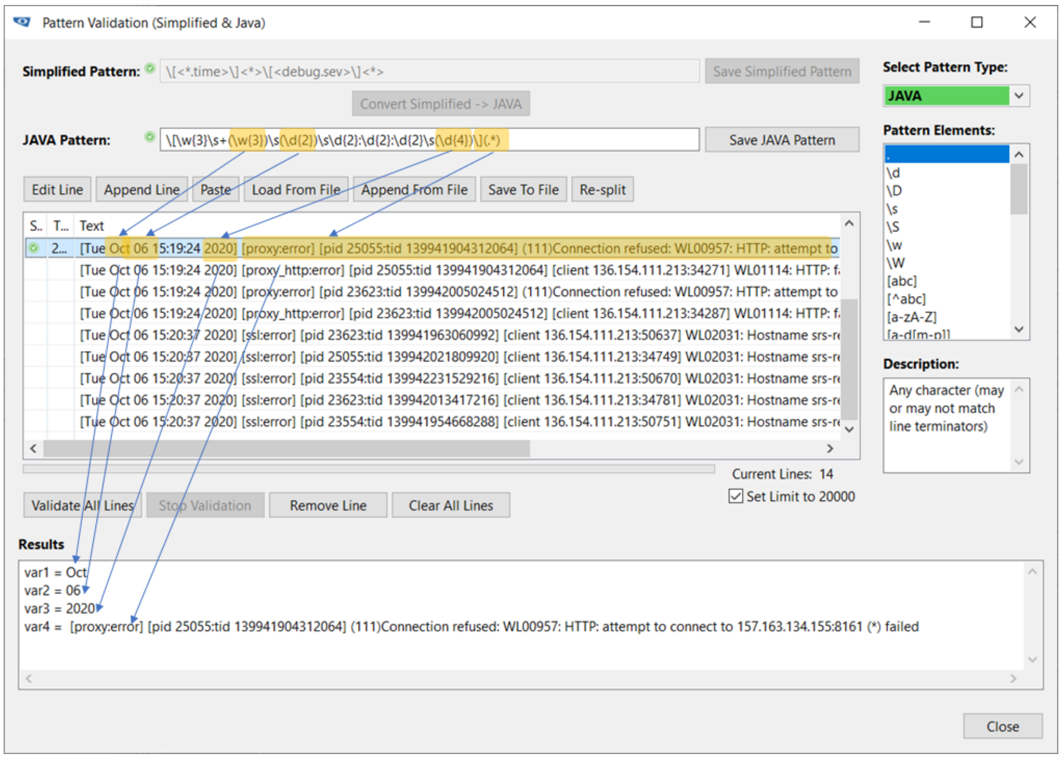
Edit Line:

```
[Tue Oct 06 15:19:17 2020] [proxy_http:error] [pid 25055:tid 139942013417216] [client 136.154.111.213:56504] WL01114: HTTP: failed to make connection to ba
```

Set Limit to 20000

This logfile can contain messages that span multiple lines as can be seen in the "Edit Line" Dialog. After using the Java Pattern in the Re-Split Dialog the multi-line messages will be displayed as a single line. This allows to quickly check that the Split-Pattern is defined correctly.

If you use brackets "()" around parts of the pattern, you can see in the result area which data this part of the pattern is matching.



Simplified Pattern: `\[<*.time>\]<*>\[<debug.sev>\]<*>`

JAVA Pattern: `\[\w{3}\s+(\w{3})\s(\d{2})\s\d{2}:\d{2}:\d{2}\s(\d{4})\](.*)`

Select Pattern Type: JAVA

```
2... [Tue Oct 06 15:19:24 2020] [proxy:error] [pid 25055:tid 139941904312064] (111)Connection refused: WL00957: HTTP: attempt to
     [Tue Oct 06 15:19:24 2020] [proxy_http:error] [pid 25055:tid 139941904312064] [client 136.154.111.213:34271] WL01114: HTTP: f
     [Tue Oct 06 15:19:24 2020] [proxy:error] [pid 23623:tid 139942005024512] (111)Connection refused: WL00957: HTTP: attempt to
     [Tue Oct 06 15:19:24 2020] [proxy_http:error] [pid 23623:tid 139942005024512] [client 136.154.111.213:34287] WL01114: HTTP: f
     [Tue Oct 06 15:20:37 2020] [ssl:error] [pid 23623:tid 139941963060992] [client 136.154.111.213:50637] WL02031: Hostname srs-r
     [Tue Oct 06 15:20:37 2020] [ssl:error] [pid 25055:tid 139942021809920] [client 136.154.111.213:34749] WL02031: Hostname srs-r
     [Tue Oct 06 15:20:37 2020] [ssl:error] [pid 23554:tid 139942231529216] [client 136.154.111.213:50670] WL02031: Hostname srs-r
     [Tue Oct 06 15:20:37 2020] [ssl:error] [pid 23623:tid 139942013417216] [client 136.154.111.213:34781] WL02031: Hostname srs-r
     [Tue Oct 06 15:20:37 2020] [ssl:error] [pid 23554:tid 139941954668288] [client 136.154.111.213:50751] WL02031: Hostname srs-r
```

Description: Any character (may or may not match line terminators)

Current Lines: 14

Set Limit to 20000

```
var1 = Oct
var2 = 06
var3 = 2020
var4 =  [proxy:error] [pid 25055:tid 139941904312064] (111)Connection refused: WL00957: HTTP: attempt to connect to 157.163.134.155:8161 (*) failed
```

5.2.3.1.2. Logfile Transaction Monitor

Another modification of the logfile monitor can be used for monitoring transactions like log entries. The responsibility of this hybrid Java monitor is the detection of not closed or failed transactions by tracking logfile entries and their expected follow-up entries.

Why to use the logfile transaction monitor:

More than one transaction is writing to the same monitored file.
The entries of the different transactions are mixed up.
The START, SUCCESS and FAILED line can be identified individually via patterns.
The log entries are single lined.

There are two type of Logfile Transaction Monitors:

- LogFile Transaction Monitor
- LogFile Transaction Monitor (Multiline)

"Multiline" gives you the possibility to track multiline logfile entries. The delimiter is defined in the "Split Records" field using the pattern notation.

The **Logfile Transaction Monitor** has the following details:

Monitor Type
Logfile transaction Monitor / (MP)Logfile Transaction Monitor

Interval
The interval specifies how often logfile is checked in minutes.

Monitor Name
com.blixx.agent.monitors.LogFileTransactionMonitor

Path/File mask
The logfile path contains the path of the monitored file or it can contain a mask for the filename.

Start Pattern
should match the start line of the transaction log file entry

Finish Pattern
should match the SUCCESS of the successfully transaction end

Fail Pattern
should match the FAIL of the failed transaction end

Timeout
transaction time out

Predefined objects provide the possibility to filter the following transactions:

FAIL_END
for failed transactions (matches the fail pattern)

SUCCESS_END
for successful transactions (matches the finish pattern)

FAIL_TIMEOUT
for timed out transactions (timeout in sec)

The following **predefined variables** can be used:

<$SUCCESS_END_MSG>
includes the finish line

<$FAIL_END_MSG>

includes the fail line

ℹ	The monitored file will always be scanned from the last processed position.

Example1:

Monitoring request

All log entries for failed cron jobs should be identified/filtered from /var/adm/cron/log.

Initial position:

- A good-job is executed every minute and succeeds
- A failed-job is executed every minute and fails

Analysis

- More than one job is writing to the same log file. The entries of the different jobs are mixed up hence the normal logfile monitor doesn't fit.
- Every log entry consists of one single line.
- The START, SUCCESS and FAILED line can be identified via patterns
- Solution: use of the "Logfile Transaction Monitor"

Extract of the cron log file

root : CMD (echo "Failjob"; exit 1) : PID (311444) : Thu Sep 29 14:36:00 2011
root : CMD (echo "Gutjob"; exit 0) : PID (364788) : Thu Sep 29 14:36:00 2011
Cron Job with pid: 364788 Successful
Cron Job with pid: 311444 Failed

The log file contains the following transactions:

Start jobs

root : CMD (echo "Failjob"; exit 1) : PID (311444) : Thu Sep 29 14:36:00 2011
root : CMD (echo "Gutjob"; exit 0) : PID (364788) : Thu Sep 29 14:36:00 2011

Failed job

Cron Job with pid: 311444 Failed

Success finish job

Cron Job with pid: 364788 Successful

Setup of the "Logfile Transaction Monitor":

1. Load the cron log file entries (Start/Fail/Success) into the pattern validation dialog.

   Define the Start Pattern / Finish Pattern / Fail Pattern for the according log entries.

   Add the pattern to the according fields in the hybrid indication policy.

   In the example the job PID defines the dependency of the Start Line and Fail/Finish Line and hence differentiates a single transaction.

   Replace Finish and Fail patterns with the variables from the start pattern.
   In our example a single variable in the start pattern needs to be extracted (see OPM Monitor) and used as in the finish and fail pattern (PID =). The PID has to be the same otherwise the lines don't belong together.

ℹ	It is not possible to extract variables from the fail and finish pattern.

Start Line Pattern

root : CMD (echo "Failjob"; exit 1) : PID (311444) : Thu Sep 29 14:36:00 2011
root : CMD (echo "Gutjob"; exit 0) : PID (364788) : Thu Sep 29 14:36:00 2011
Pattern:

```
^(\w+)\s+:\s+CMD\s+\(\s+(.*)\s+\) : PID\s+\(\s+(\d+)\s+\)\s+:.*
```

Finish Line Pattern

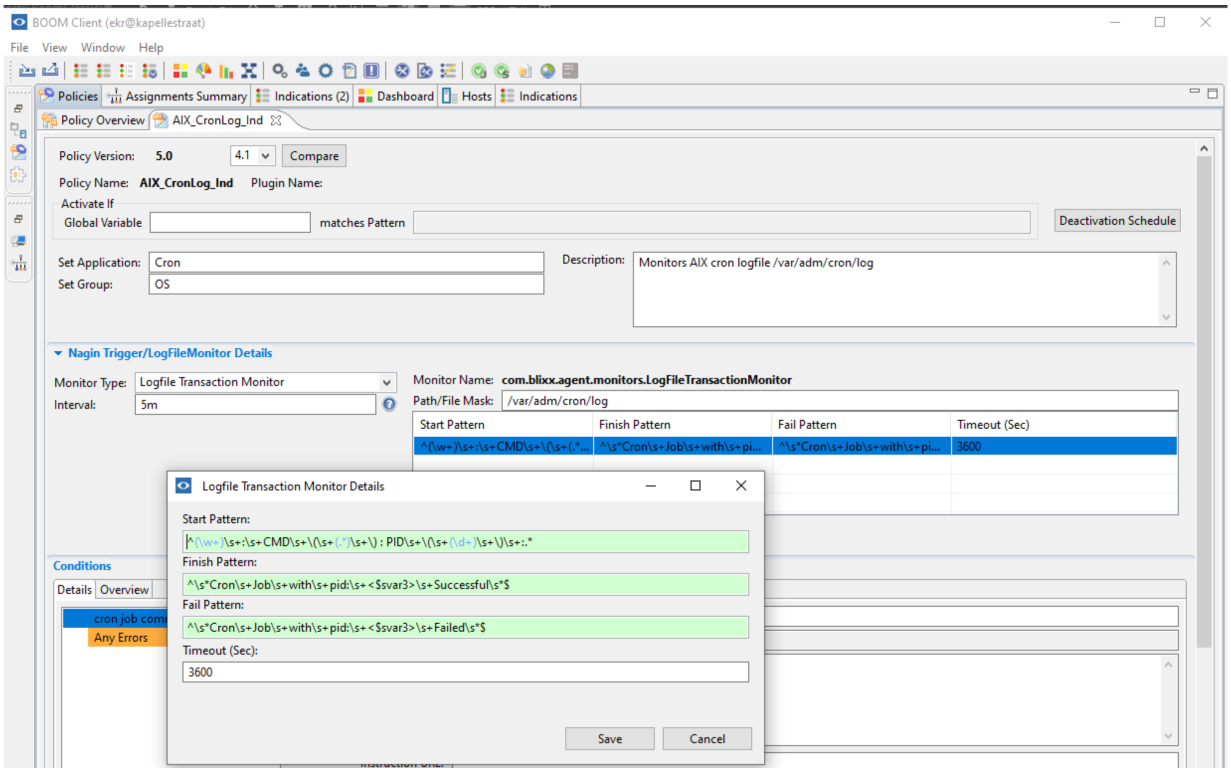
Cron Job with pid: 311444 Failed Pattern:

```
^\s*Cron\s+Job\s+with\s+pid:\s+\s+Successful\s*$
```

Fail Line Pattern

Cron Job with pid: 364788 Successful
Pattern:

```
^\s*Cron\s+Job\s+with\s+pid:\s+\s+Failed\s*$
```



Policy Version: 5.0 4.1
Policy Name: AIX_CronLog_Ind Plugin Name:
Activate If
Global Variable matches Pattern
Set Application: Cron
Set Group: OS
Description: Monitors AIX cron logfile /var/adm/cron/log

Monitor Type: Logfile Transaction Monitor
Interval: 5m
Monitor Name: com.blixx.agent.monitors.LogFileTransactionMonitor
Path/File Mask: /var/adm/cron/log

Start Pattern	Finish Pattern	Fail Pattern	Timeout (Sec)
^(\w+)\s+:\s+CMD\s+\(\s+(.*...	^\s*Cron\s+Job\s+with\s+pi...	^\s*Cron\s+Job\s+with\s+pi...	3600

Logfile Transaction Monitor Details

Start Pattern:
^(\w+)\s+:\s+CMD\s+\(\s+(.*)\s+\) : PID\s+\(\s+(\d+)\s+\)\s+:.*
Finish Pattern:
^\s*Cron\s+Job\s+with\s+pid:\s+<$svar3>\s+Successful\s*$
Fail Pattern:
^\s*Cron\s+Job\s+with\s+pid:\s+<$svar3>\s+Failed\s*$
Timeout (Sec):
3600

2. Define a condition which filters all transaction fail messages (we are not interested in success messages).

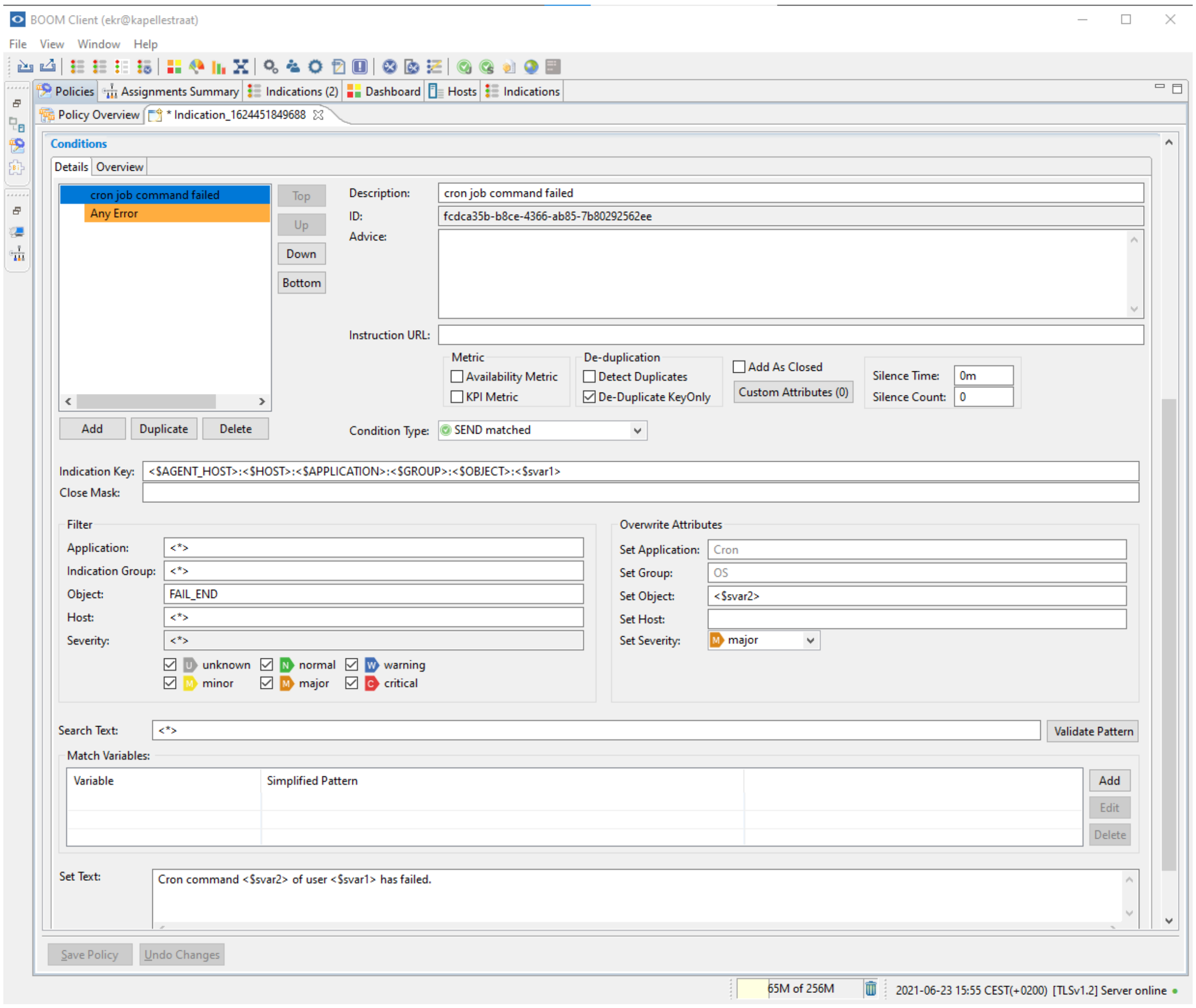
   Predefined objects provides the possibility to filter the following messages:

   FAIL_END: for failed transactions (matches the fail pattern)
   SUCCESS_END: for successful transactions (matches the finish pattern)
   FAIL_TIMEOUT: for timed out transactions (timeout in sec)

   As mentioned above it's not possible to extract any variables from the fail and finish pattern. If you want to extract information from these pattern you have to extract this with the help of predefined variables in the "Match Variables" section using simplified pattern matching.

   The following predefined variables can be used:

<**$SUCCESS_END_MSG**> includes the finish line
<**$FAIL_END_MSG**> includes the fail line


BOOM Client (ekr@kapellestraat)
File View Window Help
Policies | Assignments Summary | Indications (2) | Dashboard | Hosts | Indications
Policy Overview | * Indication_1624451849688
Conditions
Details | Overview
Top
Up
Down
Bottom
Add
Duplicate
Delete
Validate Pattern
Add
Edit
Delete
Save Policy
Undo Changes

Conditions list:
- cron job command failed
- Any Error

Description: cron job command failed
ID: fcdca35b-b8ce-4366-ab85-7b80292562ee
Advice:
Instruction URL:

Metric: ☐ Availability Metric ☐ KPI Metric
De-duplication: ☐ Detect Duplicates ☑ De-Duplicate KeyOnly
☐ Add As Closed
Custom Attributes (0)
Silence Time: 0m
Silence Count: 0
Condition Type: SEND matched

Indication Key: <$AGENT_HOST>:<$HOST>:<$APPLICATION>:<$GROUP>:<$OBJECT>:<$svar1>
Close Mask:

Filter

Application:	<*>
Indication Group:	<*>
Object:	FAIL_END
Host:	<*>
Severity:	<*>

☑ unknown ☑ normal ☑ warning
☑ minor ☑ major ☑ critical

Overwrite Attributes

Set Application:	Cron
Set Group:	OS
Set Object:	<$svar2>
Set Host:	
Set Severity:	major

Search Text: <*>

Match Variables:

Variable	Simplified Pattern

Set Text: Cron command <$svar2> of user <$svar1> has failed.

65M of 256M   2021-06-23 15:55 CEST(+0200) [TLSv1.2] Server online

Example 2:

The LogFileTransactionMonitor Java Monitor has the following call format:

com.blixx.agent.monitors.LogFileTransactionMonitor
<path to the logfile>

<***startPatternT1***>

should match start of transaction logfile entry

<***finishPatternT1***>

should match SUCCESS end of transaction

<***failPatternT1***>

should match FAILED end of transaction

<***timeoutT1***>

in seconds

<***startPatternT2***>

should match start of transaction logfile entry

<finishPatternT2>

should match SUCCESS end of transaction

<failPatternT2>

should match FAILED end of transaction

<timeoutT2>

in seconds

...

Usually such logfiles contain many asynchronously started transactions. Therefore it is recommended that the start pattern includes a capturing group(s) extracting a unique identifier of the transaction and use variables in the finish/fail patterns (<$svarN> extracted during first match) to be able to differentiate the transactions.

Several products use transaction like logging, where a missing 'successful finish message' is just as bad as a logged failure. For example the vmware ESX server logfile:

The Call field in the Hybrid Java Policy can be defined as:

com.blixx.agent.monitors.LogFileTransactionMonitor
logfilePath /var/log/vmware/hostd.log
startPattern: .*ha-eventmgr.*Event \d+ : (\S+) on host.*in ha-datacenter is starting.*
finishPattern: .*ha-eventmgr.*Event \d+ : <$svar1>.* in ha-datacenter is powered on.*
failPattern: .*ha-eventmgr.*Event \d+ : Failed to power on <$svar1>.* in ha-datacenter.*
timeout: 60

/var/log/vmware/hostd.log Example content:

```
[... 22:05:13.459 'ha-eventmgr' 54164400 info] Event 56 : vm33 on host ESX1Server in ha-datacenter is
starting
        Start pattern matched with line above. Extracts <$svar1>=vm33 - enough to uniquely identify
transaction.
        Modify finish/file pattern to:
                .*ha-eventmgr.*Event \d+ : vm33.* in ha-datacenter is powered on.*
                .*ha-eventmgr.*Event \d+ : Failed to power on vm33.* in ha-datacenter.*
        Remember start transaction message (for sending later)
        Start monitoring end of transaction or timeout.

[... 22:05:13.459 'vm:/vmfs/volumes/49c6/vm33/vm33.vmx' 54164400 info] State Transition \ (VM_STATE_OFF ->
VM_STATE_POWERING_ON)
[... 22:05:13.555 'BaseLibs' 35318704 info] VMHSVMCbPower: Setting state of VM \ /vm/#cba6c6929cc15181/ to
powerOn with option soft
[... 22:05:13.556 'BaseLibs' 35318704 info] VMHS: Exec()'ing /usr/lib/vmware/bin/vmkload_app, \
/vmfs/volumes/49c6/vm33/vm33.vmx
[... 22:05:13.920 'BaseLibs' 35318704 info] Established a connection. Killing intermediate child: 10472
[... 22:05:13.924 'BaseLibs' 35318704 info] Mounting virtual machine paths on connection: \
/db/connection/#c1/, /vmfs/volumes/49c6/vm33/vm33.vmx
[... 22:05:13.947 'BaseLibs' 35318704 info] Mount VM completion for vm: /vmfs/volumes/49c6/vm33/vm33.vmx
[... 22:05:13.948 'BaseLibs' 35318704 info] Mount VM Complete: /vmfs/volumes/49c6/vm33/vm33.vmx, \ Return
code: OK
[... 22:05:29.762 'BaseLibs' 35318704 info] VMX status has been set for vm:
/vmfs/volumes/49c6/vm33/vm33.vmx
[... 22:05:29.762 'BaseLibs' 35318704 info] Disconnect check in progress: /vmfs/volumes/49c6/vm33/vm33.vmx
[... 22:05:29.762 'BaseLibs' 35318704 info] Disconnect check in progress: /vmfs/volumes/49c6/vm33/vm33.vmx
[... 22:05:29.784 'BaseLibs' 55413680 info] Connected to /vmfs/volumes/49c6/vm33/vm33.vmx:testAutomation-
fd, \ remote end sent pid: 101091
[... 22:05:30.104 'BaseLibs' 21207984 info] DISKLIB-VMFS : "/vmfs/volumes/49c6/vm33/vm33-flat.vmdk" : \
open successful (17) size = 19327352832, hd = 0. Type 3
[... 22:05:30.123 'BaseLibs' 21207984 info] DISKLIB-VMFS : "/vmfs/volumes/49c6/vm33/vm33-flat.vmdk" :
closed.
[... 22:05:30.196 'ha-eventmgr' 21474224 info] Event 57 : vm33 on ESX1Server in ha-datacenter is powered
on
The line above matched with finishPattern. (SUCCESS) -> sends start transaction message to the boom
server.
                OBJECT=SUCCESS_END
                SEVERITY=normal
                + optional variable:
                <$SUCCESS_END_MSG>="success transaction logfile line"

        OR IF ERROR:

[... 22:05:30.196 'ha-eventmgr' 21474224 info] Event 57 : Failed to power on vm33 on ESX1Server in ha-
datacenter: \
A general system error occurred:
The line above matched with failPattern. (FAIL) -> sends start transaction message to the boom server.
                OBJECT=FAIL_END
                SEVERITY=critical
                + optional variable:
                <$FAIL_END_MSG>="fail transaction logfile line"

        OR IF TIMEOUT:

Sends start transaction message to the boom server.
                OBJECT=FAIL_TIMEOUT
                SEVERITY=critical
```

5.2.3.1.3. Logfile Transaction Monitor (multiline)

Another modification of the logfile monitor can be used for monitoring transactions like log entries. The responsibility of this hybrid Java monitor is the detection of not closed or failed transactions by tracking logfile entries and their expected follow-up entries.

There are two type of Logfile Transaction Monitors:

- LogFile Transaction Monitor
- LogFile Transaction Monitor (Multiline)

"Multiline" gives you the possibility to track multiline logfile entries. The delimiter is defined in the "Split Records" field using the pattern notation.

Why to use the logfile transaction monitor (multiline):
More than one transaction is writing to the same monitored file.
The entries of the different transactions are mixed up.
The START, SUCCESS and FAILED line can be identified individually via patterns.
The log entries are single lined or **multilined**.

The **Logfile Transaction Monitor (multiline)** has the following details:

Monitor Type	Logfile transaction Monitor / (MP)Logfile Transaction Monitor
Interval	The interval specifies how often logfile is checked in minutes.
Monitor Name	com.blixx.agent.monitors.LogFileTransactionMonitor
Path/File mask	The logfile path contains the path of the monitored file or it can contain a mask for the filename.
Split Records	Split Records allows **multi-line processing**, because many of the log files contain multiline entries. It is necessary to have here a Java Pattern that is matching the first line of the Start/Fail/Success message. Usually it can be a fixed prefix containing the date, severity, log level, etc. If a logfile has no multiline entries a simple ".*" (match all) pattern can be used.
Start Pattern	should match the start line of the transaction log file entry
Finish Pattern	should match the SUCCESS of the successfully transaction end
Fail Pattern	should match the FAIL of the failed transaction end
Timeout	transaction time out

Predefined objects provide the possibility to filter the following transactions:

FAIL_END	for failed transactions (matches the fail pattern)
SUCCESS_END	for successful transactions (matches the finish pattern)
FAIL_TIMEOUT	for timed out transactions (timeout in sec)

The following **predefined variables** can be used:

<*$SUCCESS_END_MSG*>

includes the finish line

<*$FAIL_END_MSG*>

includes the fail line

ℹ	The monitored file will always be scanned from the last processed position.

Example:

Monitoring request

All log entries for failed cron jobs should be identified/filtered from /var/adm/cron/log.

Initial position

- A good-job is executed every minute and succeeds
- A failed-job is executed every minute and fails

Analysis

- More than one job is writing to the same log file. The entries of the different jobs are mixed up hence the normal Logfile monitor doesn’t fit.
- The START/Fail/SUCCESS entry consists of **two lines**.
- The START, SUCCESS and FAILED line can be identified via patterns
- Solution: use of the "Logfile Transaction Monitor (multiline)"

Extract of the cron log file

```
> CMD: echo "Failjob";exit 1
> root 2202 c Thu Sep 29 10:05:00 2011
> CMD: echo "Gutjob";exit 0
> root 2203 c Thu Sep 29 10:05:00 2011
< root 2202 c Thu Sep 29 10:05:00 2011 rc=1
! could not obtain latest contract from popen(3C): No such process Thu Sep 29 10:05:00 2011
< root 2203 c Thu Sep 29 10:05:00 2011
! could not obtain latest contract from popen(3C): No such process Thu Sep 29 10:05:00 2011
```

The log file contains the following transactions:

Start jobs

```
> CMD: echo "Failjob";exit 1
> root 2202 c Thu Sep 29 10:05:00 2011
CMD: echo "Gutjob";exit 0
root 2203 c Thu Sep 29 10:05:00 2011
```

Failed job

```
< root 2202 c Thu Sep 29 10:05:00 2011 rc=1
```

Success finish job

```
< root 2203 c Thu Sep 29 10:05:00 2011
```

Setup of the "Logfile Transaction Monitor":

1. Load the cron log file entries into the pattern validation dialog.

Define the Start Pattern / Finish Pattern / Fail Pattern for the according log entries.

Add the pattern to the according fields in the hybrid indication policy.

In the example the job PID and the user define the dependency of the Start Line and Fail/Finish Line and hence differentiate a single transaction.

Replace Finish and Fail patterns with the variables from the start pattern. In our example the single variables in the start pattern needs to be extracted (see OPM Monitor) and used as and in the finish and fail pattern (PID = , User=). The PID and user have to be the same otherwise the lines don't belong together.

ℹ	It is not possible to extract variables from the fail and finish pattern.

Start Line Pattern

```
< CMD: echo "Failjob";exit 1
< root 2202 c Thu Sep 29 10:05:00 2011
Pattern: >\s+CMD:\s+(.*)\n>\s+(\w+)\s+(\d+)\s+.*
```

Fail Line Pattern

```
< root 2202 c Thu Sep 29 10:05:00 2011 rc=1
Pattern: \s+\s+[^\n]+\s+rc=\d+\s*\n*.*
```

Finish Line Pattern

```
< root 2203 c Thu Sep 29 10:05:00 2011
Pattern: \s+\s+[^\n]+\s+\d+\s*(|\n.*)
```

2. Define **"Split Records"** Java Patterns for the Start/Fail and Success information.

   Split Records: (>\s+CMD:|<\s+).* The pattern ">\s+CMD:.*" matches the Start Transaction which matches the following lines:

```
> CMD: echo "Failjob";exit 1
> root 2202 c Thu Sep 29 10:05:00 2011
```

   The pattern "<\s+.*" matches the Fail and Success Transaction which matches the following lines:

```
< root 2202 c Thu Sep 29 10:05:00 2011 rc=1
< root 2202 c Thu Sep 29 10:05:00 2011
```

   Use the Java Pattern Validation Dialog to test the "Split Records" pattern definitions.
   Use the **"Re-split"** button to test the recognition of multiline records.

ℹ	Based on the "Split Records" definitions the monitor matches as many lines as possible. That have to be been taken into account in the Fail and Success patterns definition.

3. Define a condition which filters all transaction fail messages (we are not interested in success messages).

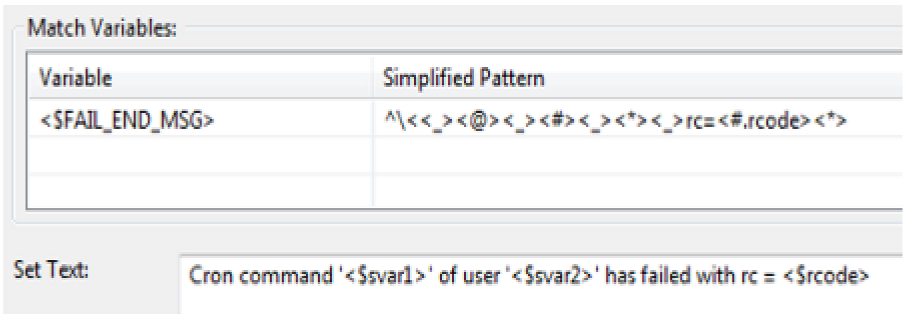
   Predefined objects provides the possibility to filter the following messages:

FAIL_END	for failed transactions (matches the fail pattern)
SUCCESS_END	for successful transactions (matches the finish pattern)
FAIL_TIMEOUT	for timed out transactions (timeout in sec)

As mentioned above it’s not possible to extract any variables from the fail and finish pattern. If you want to extract information from these pattern you have to extract this with the help of predefined variables in the "Match Variables" section using simplified pattern matching.

The following predefined variables can be used:

*<**$SUCCESS_END_MSG**>*	includes the finish line
*<**$FAIL_END_MSG**>*	includes the fail line


Match Variables:

Variable	Simplified Pattern
<$FAIL_END_MSG>	^\<<_><@><_><#><_><*><_>rc=<#,rcode><*>

Set Text: Cron command '<$svar1>' of user '<$svar2>' has failed with rc = <$rcode>

5.2.3.2. MultiPath (MP) Logfile Monitor

The Multipath Logfile Monitors are **extensions of the Logfile Monitors** that we have in the *boom*. The main difference is that the file/path mask is extended to support not only **file masks but also path masks**. It is **recommended to use the MP logfile family** instead of the normal ones. These are kept for backwards compatibility.

The path mask syntax is specialized for finding rotating log files and consists of the following rules:

- '*' (star) symbol matches any char sequence within path element name. Among all matching files, the file with latest modification time should be taken. This is useful to monitor rotating logfiles, so if logging has switched to another file, the monitor will also pick the newest file (after finishing the previous one).
- Several '*' (star) symbols in different path elements (i.e. on different file tree levels) still comprise the same single group for selecting the newest file.
- '**' (double star) symbol matches any char sequence within path element name. All matching files should be taken as separate monitoring target, regardless of modification time.
- The path element consisting of exactly double star (like '/**/') specifies recursive scan and matches arbitrary path depth, including zero.
- double star overrides single star in the same path element, that is all matching files will be taken.
- path starting from wildcard is considered relative to current dir, unless followed by colon on MS Windows. This is reserved as a special notation for selecting all disk drives on MS Windows: **: (is the same as ***:).
- All slashes and backslashes are treated as file (path element) separators, interim double slashes are reduced to single slash. On MS Windows, network paths starting with double (back-)slash are also recognized.
- Several independent masks can be specified, separated by the '|' (pipe) symbol.
- Path mask can be replaced with an exec calls: <$LOGFILES(sh -c "ls -l")> or <$BOOMMON_ONINIT(sh -c "ls -l")>. In this case the MP LFM expects back a list of complete log file paths (one or more) separated by '|' (pipe) or new

line '\n' character. <$LOGFILES> runs every time the policy is triggered to determine the logfiles to monitor, whereas <$BOOMMON_ONINIT()> only runs upon (re)deployment of a policy.

For more information please see chapter Indication Policy for Multipath Logfile Monitors.

Examples:

1. We have the following directory Structure:

   ```
   /web/logs/error1.log
   /web/logs/error2.log
   /web/logs/error3.log
   /web/logs/error4.log
   ```

   Path/File Mask:

   ```
   /web/logs/error*.log
   ```

   Results:
   Assuming that the most recent file is "/web/logs/error4.log" this file will be monitored.

   Explanation:
   picks a newest (most lately modified) file whose name starts with 'error' and ends with '.log' in the directory /web/logs. This is a typical configuration to monitor a rotated log file, e.g. Apache's logs.

2. We have the following directory Structure:

   ```
   /app/services/errors.log
   /app/services/warnings.log
   /app/services/information.log
   /app/services/debug.log
   /app/services/exportedfile.txt
   ```

   Path/File Mask:

   ```
   /app/services/**.log
   ```

   Results:

   ```
   /app/services/errors.log
   /app/services/warnings.log
   /app/services/information.log
   /app/services/debug.log
   ```

   Explanation:
   The Path/File mask selects all files with extension .log in directory /app/services. Unlike the previousexample, this pattern assumes no log rotation, and simply selects a bunch of existing files for batch processing.

3. We have the following directory Structure:

```
/var/log/apache/instance1/access_log1
/var/log/apache/instance1/access_log2
/var/log/apache/instance1/access_log3
/var/log/apache/instance1/access_log4
/var/log/apache/instance2/access_log1
/var/log/apache/instance2/access_log2
/var/log/apache/instance2/access_log3
/var/log/apache/instance2/access_log4
/var/log/apache/instance3/access_log1
/var/log/apache/instance3/access_log2
/var/log/apache/instance3/access_log3
/var/log/apache/instance3/access_log4
```

Path/File Mask:

```
/var/log/apache/instance**/access_log*
```

Results:
Assuming that the last logfile of every directory is the newest one the following files would be selected:

```
/var/log/apache/instance1/access_log4
/var/log/apache/instance2/access_log4
/var/log/apache/instance3/access_log4
```

Explanation:
The Path/File mask separates every directory in a virtual group, so 3 groups will be created:

```
Group1:
    /var/log/apache/instance1/access_log1
    /var/log/apache/instance1/access_log2
    /var/log/apache/instance1/access_log3
    /var/log/apache/instance1/access_log4
```

```
Group2:
    /var/log/apache/instance2/access_log1
    /var/log/apache/instance2/access_log2
    /var/log/apache/instance2/access_log3
    /var/log/apache/instance2/access_log4
```

```
Group3:
    /var/log/apache/instance3/access_log1
    /var/log/apache/instance3/access_log2
    /var/log/apache/instance3/access_log3
    /var/log/apache/instance3/access_log4
```

Now from every group the most recent file will be selected leaving us with the files specified in "results".

So the mask /var/log/apache/instance**/access_log* in our directory structure would be the same as:

```
/var/log/apache/instance1/access_log*
/var/log/apache/instance2/access_log*
/var/log/apache/instance3/access_log*
```

But the advantage of using "**" is that it is flexible in the number of instances (directories) since they don't have to be entered manually.

4. We have the following directory Structure:

```
/archive/2011-09-15/outage1.log
/archive/2011-09-15/outage2.log
/archive/2011-09-15/outage3.log
/archive/2011-09-15/error.log
/archive/2011-09-16/outage1.log
/archive/2011-09-16/outage2.log
/archive/2011-09-16/outage3.log
/archive/2011-09-16/error.log
/archive/2011-09-17/outage1.log
/archive/2011-09-17/outage2.log
/archive/2011-09-17/outage3.log
/archive/2011-09-17/error.log
```

   Path/File Mask:

```
/archive/*/outage*.log
```

   Results:
   Assuming that the last logfile of every directory is the newest one the following files would be selected:

```
/archive/2011-09-17/outage3.log
```

   Explanation:
   The Path/File mask selects a newest file named like outageXXX.log among all files in direct subdirectories of directory /archive. This is a more sophisticated example of log rotation policy, e.g. when logs are grouped per sub-directories by date:

```
/archive/--/outage-.log
```

5. We have the following directory Structure:

```
/var/log/archive/2011-09-15/outage1.log
/var/log/archive/2011-09-15/outage2.log
/var/log/archive/2011-09-15/error.log
/var/log/archive/2011-09-16/outage1.log
/var/log/archive/2011-09-16/outage2.log
/var/log/archive/2011-09-16/error.log
/var/log/archive/2011-09-17/outage1.log
/var/log/archive/2011-09-17/outage2.log
/var/log/archive/2011-09-17/error.log
/var/log/apache/instance1/access_log1
/var/log/apache/instance1/access_log2
/var/log/apache/instance2/access_log1
/var/log/apache/instance2/access_log2
/var/log/apache/instance3/access_log1
/var/log/apache/instance3/access_log2
/var/log/app/services/errors.log
/var/log/app/services/warnings.log
/var/log/app/services/information.log
/var/log/app/services/debug.log
/var/log/app/services/exportedfile.txt
/var/log/web/error1.log
/var/log/web/error2.log
/var/log/web/error3.log
/var/log/web/error4.log
/var/log/test.log
/var/log/error.log
/var/log/messages
```

Path/File Mask:

```
/var/log/**/**.log
```

Results:

```
/var/log/archive/2011-09-15/outage1.log
/var/log/archive/2011-09-15/outage2.log
/var/log/archive/2011-09-15/error.log
/var/log/archive/2011-09-16/outage1.log
/var/log/archive/2011-09-16/outage2.log
/var/log/archive/2011-09-16/error.log
/var/log/archive/2011-09-17/outage1.log
/var/log/archive/2011-09-17/outage2.log
/var/log/archive/2011-09-17/error.log
/var/log/app/services/errors.log
/var/log/app/services/warnings.log
/var/log/app/services/information.log
/var/log/app/services/debug.log
/var/log/web/error1.log
/var/log/web/error2.log
/var/log/web/error3.log
/var/log/web/error4.log
/var/log/test.log
/var/log/error.log
```

Explanation:
The double star "**" will recursively scan every number of directories and subdirectories, including zero (so files directly in "/var/log" will be matched two; in this example /var/log/test.log and /var/log/error.log). The double star

"**" of the files will select every file which ends in ".log".

6. We have the following directory Structure:

```
/var/log/archive/2011-09-15/outage1.log
/var/log/archive/2011-09-15/outage2.log
/var/log/archive/2011-09-15/error.log
/var/log/archive/2011-09-16/outage1.log
/var/log/archive/2011-09-16/outage2.log
/var/log/archive/2011-09-16/error.log
/var/log/archive/2011-09-17/outage1.log
/var/log/archive/2011-09-17/outage2.log
/var/log/archive/2011-09-17/error.log
/var/log/apache/instance1/access_log1
/var/log/apache/instance1/access_log2
/var/log/apache/instance2/access_log1
/var/log/apache/instance2/access_log2
/var/log/apache/instance3/access_log1
/var/log/apache/instance3/access_log2
/var/log/app/services/errors.log
/var/log/app/services/warnings.log
/var/log/app/services/information.log
/var/log/app/services/debug.log
/var/log/app/services/exportedfile.txt
/var/log/web/error1.log
/var/log/web/error2.log
/var/log/web/error3.log
/var/log/web/error4.log
/var/log/test.log
/var/log/error.log
/var/log/messages
```

   Path/File Mask:

```
/var/log/**
```

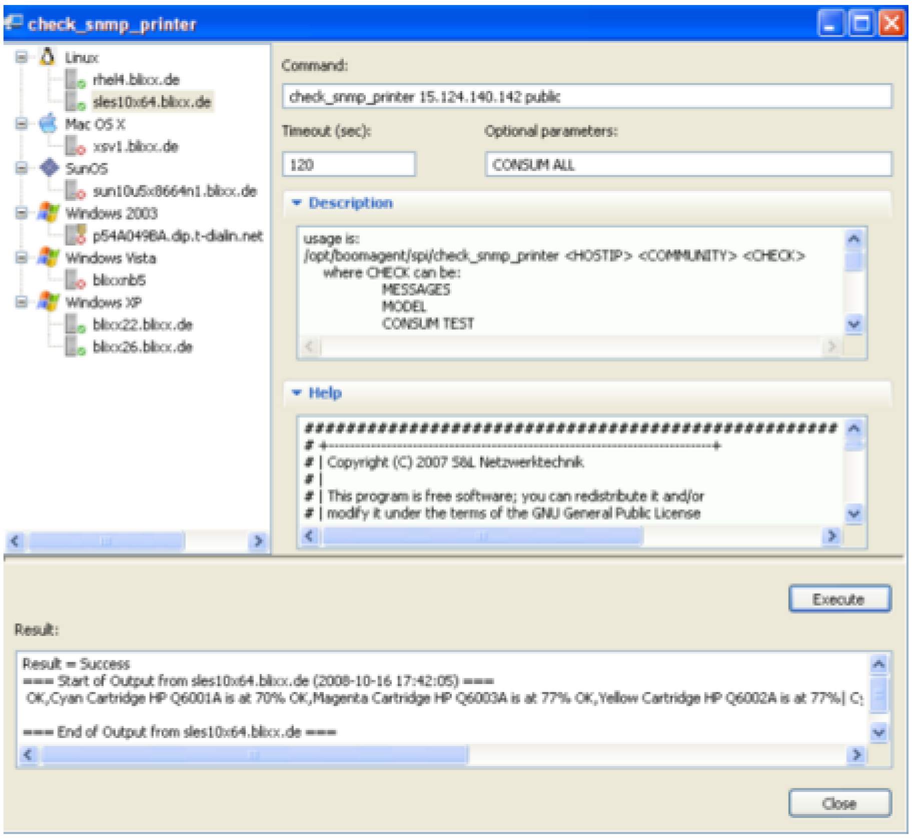
   Results:
   Every file in /var/log and subdirectories will be matched

   Explanation:
   The double star "**" will recursively scan every number of directories and subdirectories and select every file.

5.2.3.3. NAGIN

A Hybrid Policy example that uses a shell script for checking networks printer cartridges

Just to show how this works, we will take the following free plug-in from the Nagios® exchange without any changes. The script expects the parameters Printer IP, SNMP Community and Command. It is easy to create an action inside the *boom* GUI to get some test output from the deployed script.



Command: check_snmp_printer 15.124.140.142 public

Timeout (sec): 120

Optional parameters: CONSUM ALL

```
usage is:
/opt/boomagent/spi/check_snmp_printer <HOSTIP> <COMMUNITY> <CHECK>
    where CHECK can be:
            MESSAGES
            MODEL
            CONSUM TEST
```

```
###########################################################
# +--------------------------------------------------------+
# | Copyright (C) 2007 S&L Netzwerktechnik
# |
# | This program is free software; you can redistribute it and/or
# | modify it under the terms of the GNU General Public License
```

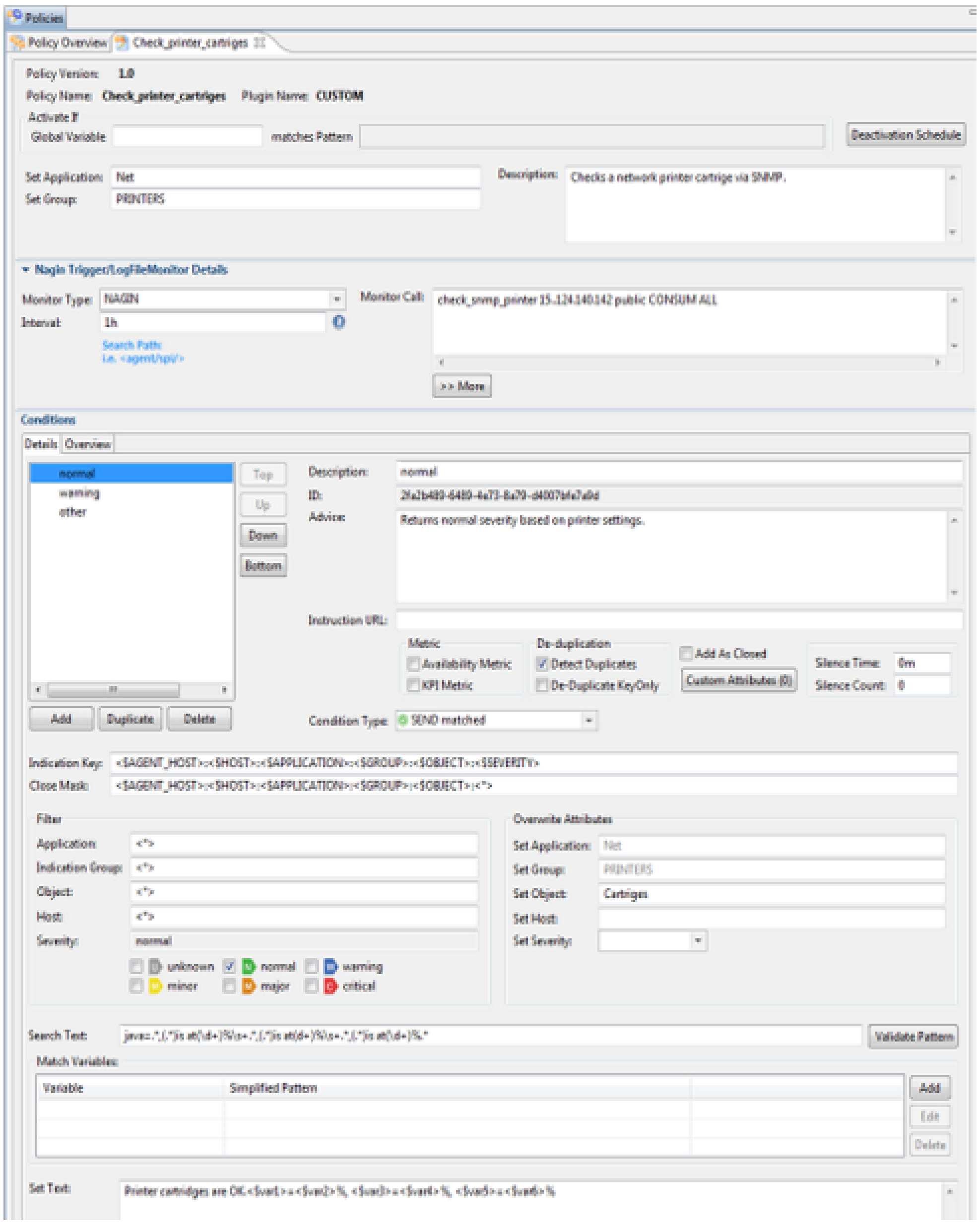
Result:

```
Result = Success
=== Start of Output from sles10x64.blxx.de (2008-10-16 17:42:05) ===
 OK,Cyan Cartridge HP Q6001A is at 70% OK,Magenta Cartridge HP Q6003A is at 77% OK,Yellow Cartridge HP Q6002A is at 77%| C;

=== End of Output from sles10x64.blxx.de ===
```

The result output string is conform with the Nagios® requirements and contains information about the toner level for every cartridge:

```
OK,Cyan Cartridge HP Q6001A is at 70% OK,Magenta Cartridge HP Q6003A is at 77% \ <
OK,Yellow Cartridge HP Q6002A is at 77%\
| Cyan Cartridge HP Q6001A=70;;470; Magenta Cartridge HP Q6003A=77;;470; \
Yellow Cartridge HP Q6002A=77;;470; Result
```

A simple hybrid indication policy can be created with the following settings:



Policy Version: 1.0
Policy Name: Check_printer_cartriges Plugin Name: CUSTOM
Activate If
Global Variable: matches Pattern:
Set Application: Net
Set Group: PRINTERS
Description: Checks a network printer cartrige via SNMP.

Nagin Trigger/LogFileMonitor Details
Monitor Type: NAGIN
Interval: 1h
Search Path: i.e. <agent/spi/>
Monitor Call: check_snmp_printer 15.124.140.142 public CONSUM ALL

Conditions
normal
warning
other
Description: normal
ID: 2fa2b489-6489-4a73-8a79-d4007bfa7a0d
Advice: Returns normal severity based on printer settings.
Instruction URL:
Metric: Availability Metric [ ], KPI Metric [ ]
De-duplication: Detect Duplicates [✓], De-Duplicate KeyOnly [ ]
Add As Closed [ ]
Silence Time: 0m
Silence Count: 0
Condition Type: SEND matched

Indication Key: <$AGENT_HOST>:<$HOST>:<$APPLICATION>:<$GROUP>:<$OBJECT>:<$SEVERITY>
Close Mask: <$AGENT_HOST>:<$HOST>:<$APPLICATION>:<$GROUP>:<$OBJECT>:<*>

Filter
Application: <*>
Indication Group: <*>
Object: <*>
Host: <*>
Severity: normal
[ ] unknown [✓] normal [ ] warning
[ ] minor [ ] major [ ] critical

Overwrite Attributes
Set Application: Net
Set Group: PRINTERS
Set Object: Cartriges
Set Host:
Set Severity:

Search Text: java:.*(.*)is at(\d+)%\s+.*(.*)is at(\d+)%\s+.*(.*)is at(\d+)%.*
Match Variables:

Variable	Simplified Pattern	

Set Text: Printer cartridges are OK.<$var1>=<$var2>%, <$var3>=<$var4>%, <$var5>=<$var6>%

For every specified polling interval the *boom* agent will call the script and process the output.

Filtering and Mapping

The initial severity of the execution is based on the result codes that are defined in the Nagios® Plugin API.

Plugin Return Code	*boom* Severity	Nagios® Service State	Nagios® Host State
0	normal	OK	UP
1	warning	WARNING	UP or DOWN/UNREACHABLE
2	critical	CRITICAL	DOWN/UNREACHABLE
3	unknown	UNKNOWN	DOWN/UNREACHABLE

***If the result code is different from the declared code above - the _boom** agent assumes that an error occurred during the execution and will produce a warning indication with a detailed status._*

Original values of other Indication fields for the NAGIN call type:

Field	Value
Application	NAGIN
Indication Group	NAGIN
Object	empty
Node	Agent’s hostname

All original values can be changed by specifying the **"Overwrite Attributes"** section of the condition.
The call parameters can be defined in a multi-line form. Both notations are identical.

```
Call: check_snmp_printer
    15.124.140.142
    public
    CONSUM ALL
```

If you use the default text pattern <*>, the indication will be delivered as it is. On the other side if you use variables inside the patterns it is easy to construct a new form of text. For example the following pattern:

```
java=.*,(.*) is at (\d+)%\s+.*,(.*) is at (\d+)%\s+.*,(.*) is at (\d+)%.*
```

together with Set Text:

```
Printer cartridges are OK. <$var1>=<$var2>%, <$var3>=<$var4>%, <$var5>=<$var6>%
```

will create the Indication text:

```
Printer cartridges are OK. Cyan Cartridge HP Q6001A=70%, \ Magenta Cartridge HP Q6003A=77%, Yellow
Cartridge HP Q6002A=77%
```

Monitoring multiple hosts and objects with one Policy

There is a special possibility for hybrid policies with a NAGIN call type. The call field can have a special declaration that allows to declare a list of values:

```
check_snmp_printer
<$BOOMMON_NODES(15.124.140.142,15.124.140.143,15.124.140.145)>
public
<$BOOMMON_OBJECTS(PAPER1,PAPER2)>
```

This kind of notation allows to create multiple calls with different parameters inside the same policy.

<$BOOMMON_NODES(comma separated list)> defines a list of parameters that must be delivered as a Host value of indication.

<$BOOMMON_OBJECTS(comma separated list)> defines list of parameters that must be delivered as object value of an indication.

For a more detailed description of these two pre-process functions and their parameters, please refer to the chapter Supported pre-process Functions.

Multiple entries of the same type (BOOMMON_NODES, BOOMMON_OBJECTS) are not allowed in the call field.

As a result of this call declaration the agent will trigger the script 6 times:

```
check_snmp_printer   15.124.140.142   public PAPER1
check_snmp_printer   15.124.140.143   public PAPER1
check_snmp_printer   15.124.140.145   public PAPER1
check_snmp_printer   15.124.140.142   public PAPER2
check_snmp_printer   15.124.140.143   public PAPER2
check_snmp_printer   15.124.140.145   public PAPER2
```

The created indications will be submitted with the Host and Object values that are used in the call. This enables creation of different conditions for different incoming objects and/or hosts.

The default working directory for triggers is "$BOOM_ROOT/spi/". All binaries and script calls must be specified related to this directory. In case it is necessary to use a binary which is placed in a different location but is available in the systems PATH - use the '#' character as a prefix. i.e. #df, #top.

5.2.4. SNMP Traps

An SNMP Traps policy processes traps which were received on the agent through the *boom* SNMP-TrapReceiver. To be able to receive traps it is necessary to assign and deploy the assignment group "SNMP-TrapReceiver" or combination of the "SNMP package" and the "SNMPTrapd_Trigger" policy to the specified agent.
The *boom* SNMP-TrapReceiver converts all incoming traps into *boom* indications for processing by all deployed SNMP Traps policies.

Each SNMP trap policy has to use following **required filters** in every condition:

Filter	Value
Application	SNMPTrapd
Indication Group	SNMP

To filter specific trap OIDs it is recommended to use the **Object** filter by entering the complete Trap OID or a pattern (see screenshot).

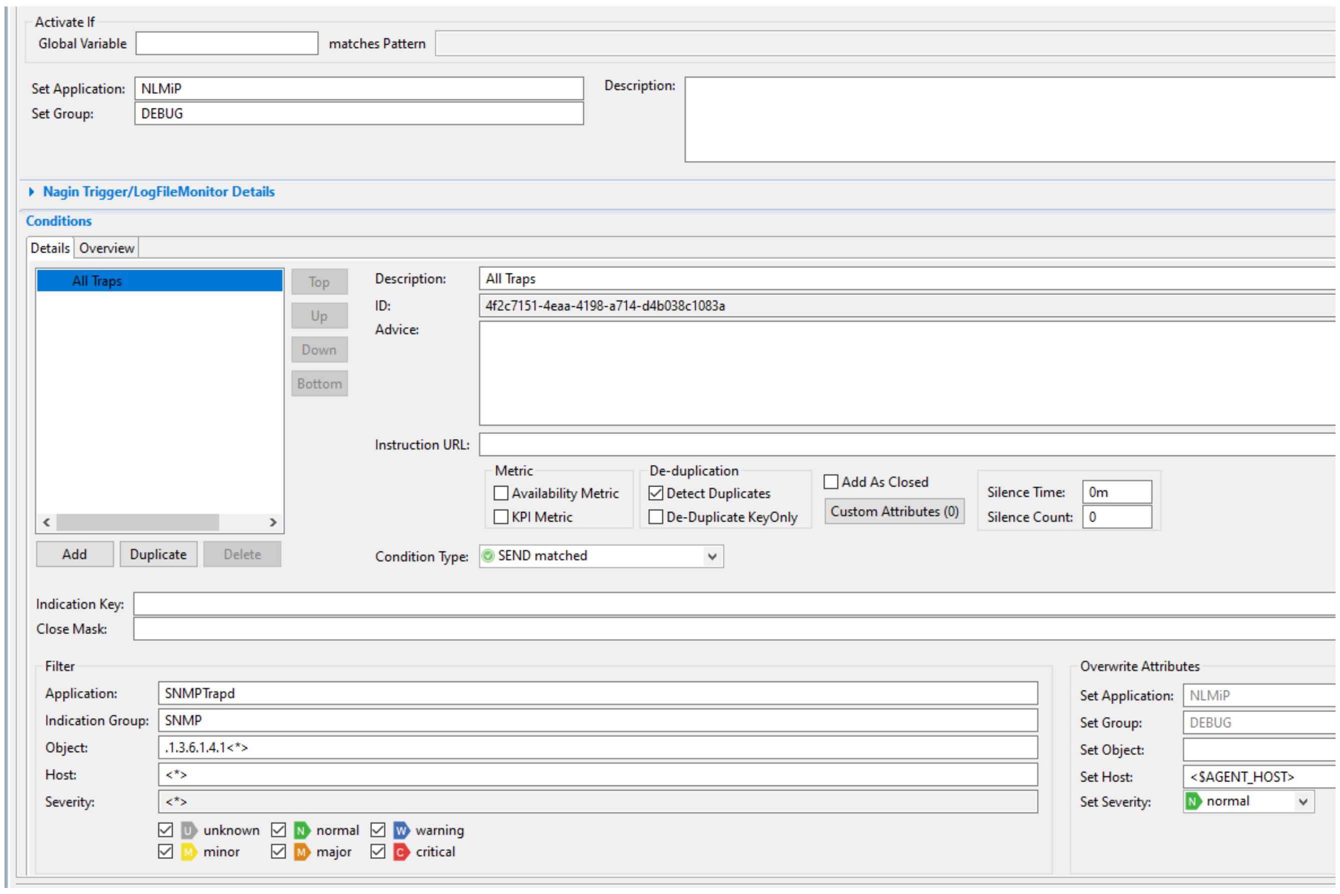
To access the trap variables e.g. for usage in indication text or for filtering in the **"Match Variables"** section, variables <$1>...<$n> are used.

Also the **"Search Text"** field can be used for plain pattern matches on concatenated TrapOIDs and variables.
The "Match Variables" field can also be used to construct text from variables and define match conditions.

Use a "Stop condition" for non matching traps (e.g. different OID) as first condition. This prevent further processing of policy conditions for unwanted traps. "DROP unmatched" Condition Type can be used in this case. i.e. if policy has to match only Link Up (OID=.1.3.6.1.6.3.1.1.5.4) and Link Down (OID=.1.3.6.1.6.3.1.1.5.3) traps - appropriate "Stop condition" will be "DROP unmatched" with filter Object .1.3.6.1.6.3.1.1.5.<*>

Application	SNMPTrapd
Indication Group	SNMP
Object	.1.3.6.1.6.3.1.1.5.<*>
Condition Type	"DROP unmatched"

For further information on how to configure SNMP policies see also chapter Indication Policy for SNMP traps



Activate If
Global Variable: matches Pattern:

Set Application: NLMiP
Set Group: DEBUG
Description:

Nagin Trigger/LogFileMonitor Details

Conditions

Details | Overview

All Traps

Top | Up | Down | Bottom

Add | Duplicate | Delete

Description: All Traps
ID: 4f2c7151-4eaa-4198-a714-d4b038c1083a
Advice:
Instruction URL:

Metric: ☐ Availability Metric, ☐ KPI Metric
De-duplication: ☑ Detect Duplicates, ☐ De-Duplicate KeyOnly
☐ Add As Closed
Custom Attributes (0)
Silence Time: 0m
Silence Count: 0

Condition Type: SEND matched

Indication Key:
Close Mask:

Filter
Application: SNMPTrapd
Indication Group: SNMP
Object: .1.3.6.1.4.1<*>
Host: <*>
Severity: <*>
☑ unknown ☑ normal ☑ warning
☑ minor ☑ major ☑ critical

Overwrite Attributes
Set Application: NLMiP
Set Group: DEBUG
Set Object:
Set Host: <$AGENT_HOST>
Set Severity: normal

5.3. Patterns

5.3.1. Simplified Patterns

Simplified Pattern Syntax

The simplified pattern syntax is available in *boom* because it provides a more readable form. In addition it allows defining names for custom variables. In most cases it is enough to have simple patterns to match incoming indications. On the other side, this format has sensitive limitations and might be not sufficient for real complex text processing.

The rules by which *boom* assigns strings to variables behaves like HP Operations Manager's algorithm. The HP Operations Manager's pattern matching algorithm always scans both the input line and the pattern definition (including alternative expressions) from left to right. <*> expressions are assigned as few characters as possible. <#>, <@> and <_> expressions are assigned as many characters as possible (greedy).

Example: In matching the pattern <*.var1><*.var2> against the string "test", var1 will be assigned an empty string, var2 will be assigned the string "test".

Element	Description
<*>	matches all
<n*>	matches a sequence of 'n' arbitrary characters
<#>	matches a sequence of one or more digits [0-9]
<n#>	matches a sequence of 'n' digits character

Element	Description
<@>	matches a sequence word or digits character without separators [a-zA-Z_0-9]
<n@>	matches a sequence of 'n' word or digits character without separators
<_>	matches a sequence of separators [\t\n\x0B\f\r]
<n_>	matches a sequence of 'n' separators
<![]>	"NOT" operator
<*.varName> <#.varName> <@.varName> <_.varName> <[<n*>].varName> <[<n#>].varName> <[<n@>].varName> <[<n_>].varName>	Stores the matching result in the variable "varName". This variable can be used during construction of indication. To access the variable use <$varName>.
<[aaa|bbbb]> [aaa|bbbb]	"OR" condition - allows to match "aaa" OR "bbbb". "OR" condition does not support declaration of variable directly.
<[<[aaa|bbbb]>].varName>	"OR" condition can be inserted in superset pattern group declaration. it allows to match "aaa" OR "bbbb" and store in the variable declared in the superset pattern group.
<[] -eq n>	Equal
<[] -ne n>	Not Equal
<[] -lt n>	Less Than
<[] -le n>	Less or Equal
<[] -gt n>	Greater Than
<[] -ge n>	Greater or Equal
<n -lt [] -lt n>	Open Interval
<n -le [] -le n>	Closed Interval
<n -gt [] -gt n>	Open Interval
<n -ge [] -ge n>	Closed Interval
<^javapattern^>	Allows to inject a java pattern

It is possible to have inner capturing groups to parse additional variables from extracted parts.

For Example:

Incoming text:

```
--- 01/01/2008 x--x Error404 Page not found
```

It is necessary to extract a single message part: "Error404 Page not found" together with two sub-elements: "404" and "Page not found" as a separate value.

The pattern

```
<*><[Error<[<#.n><_><*.msg>]>].complete>
```

delivers the following result:

```
<$complete>=Error404 Page not found
<$dummy1>=404 Page not found
<$n>=404
<$msg>=Page not found
```

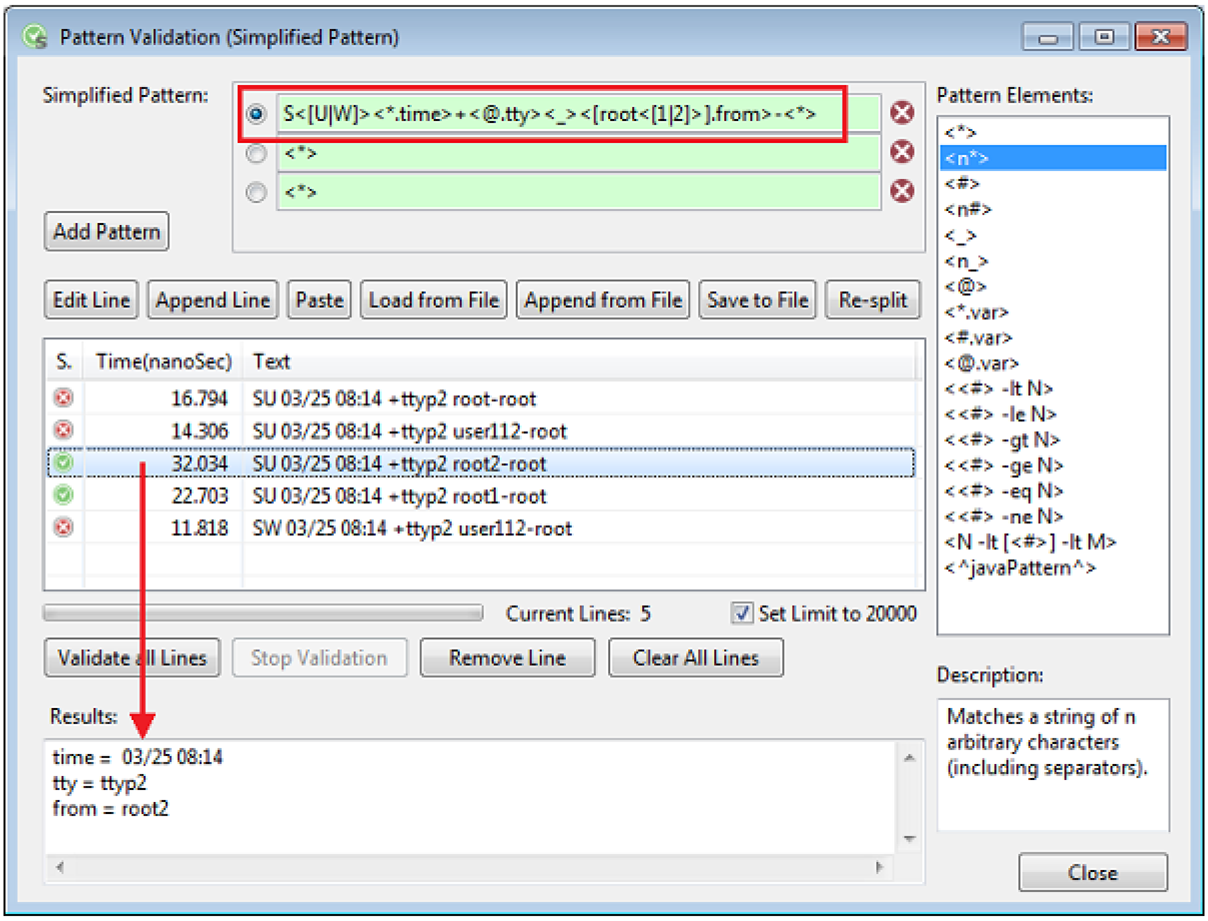
where dummy1 is an intermediate internal variable which is used during the calculation.

The next example shows the use case where the goal of filtering is to match messages for all users except the following: {"root", "user1", and "user2"}.

The complexity of this task is that there are also users available with names like: "user10", "user112".

The proposed pattern do this job by using sub-pattern is:

```
[user[1|2]] combined with <![root].from>
```



Simplified Pattern:

[● S<[U|W]><*.time>+<@.tty><_><[root<[1|2]>].from>-<*>]
○ <*>
○ <*>

S.	Time(nanoSec)	Text
✗	16.794	SU 03/25 08:14 +ttyp2 root-root
✗	14.306	SU 03/25 08:14 +ttyp2 user112-root
✓	32.034	SU 03/25 08:14 +ttyp2 root2-root
✓	22.703	SU 03/25 08:14 +ttyp2 root1-root
✗	11.818	SW 03/25 08:14 +ttyp2 user112-root

Current Lines: 5 ☑ Set Limit to 20000

Results:

```
time = 03/25 08:14
tty = ttyp2
from = root2
```

Pattern Elements:

<*>
<n*>
<#>
<n#>
<_>
<n_>
<@>
<*.var>
<#.var>
<@.var>
<<#> -lt N>
<<#> -le N>
<<#> -gt N>
<<#> -ge N>
<<#> -eq N>
<<#> -ne N>
<N -lt [<#>] -lt M>
<^javaPattern^>

Description:

Matches a string of n arbitrary characters (including separators).

5.3.2. Java Patterns

Java patterns are used for the "Search Text" attribute of indication or hybrid policies and a couple of bundled Java monitors (Opm and LogFileMonitor). The full description of a Java pattern format can be found on Sun's Java Platform API page (see Pattern class).

The *boom* Java patterns also support variables but in a slightly different way than the 'Simplified Pattern Syntax'. By using the Java patterns you can not define custom names but you can use predefined names: <$var1>,<$var2>,<$var3>...

The rule by which the Java pattern matching algorithm assigns strings to variables is always greedy (expressions are assigned as many characters as possible).

Short overview about the Java Pattern Syntax:

Java Pattern Classes	Description
.	any character.
.?	any character once or not at all.
.*	any character zero or more times.
.+	any character one or more times.
a+	'a' character one or more times.
.{n}	matches a sequence of 'n' characters
\d+	matches a sequence of one or more digits [0-9]
\d{n}	matches a sequence of 'n' digits character
\w+	matches a sequence word or digits character without separators [a-zA-Z_0-9]
\w{n}	matches a sequence of 'n' word or digits character without separators
\s+	matches a sequence of separators [\t\n\x0B\f\r]
\s{n}	matches a sequence of 'n' separators
(X) where X - some pattern. i.e (\w+)	Declares a capturing group. Value of capturing group will be stored in the variable "varN", where N is a number of group in the patters started from 0. To access variable - use <$varN>
(?:aaa|bbbb)	"OR" condition - allows to match "aaa" OR "bbbb" as non-capturing group.
(aaa|bbbb)	"OR" condition - allows to match "aaa" OR "bbbb" as capturing group <$varN>.
[a-dA-D]	any character of set: a,b,c,d,A,B,C,D
[^a-dA-D]	any character except a,b,c,d,A,B,C,D
(.*)(?<!abc)	any string EXCEPT "abc". Second non-capturing group looks on result of first
(?!abc)(.*)	any string EXCEPT "abc". First non-capturing group looks on result of second
((?!abc).)*	any string NOT CONTAINING "abc".
(?i)	switch to case insensitive pattern matching.
(?m)	switch to multi-line pattern matching.

5.3.3. Supported Variables and Functions

5.3.3.1. Global Variables

Initialisation/Setting of a Global Variable

There are various possibilities to initialize/set global variables in the *boom* agent:

- <boom_agent_install_dir>/conf/**global.props**
  Add the global variable in the following format: <varname>=<value> e.g. globalvar=1
- Command line tool **"boomsetvar"**
  During runtime any external application can call this command to set or change a global variable e.g. ./boomsetvar globalvar=1 globalvar2=2
- Agent Action [GUI → View Actions → BOOM_AGENTS]: **"Set Agent's Global Variable"** sets the global variable from the *boom* server (as remote action). Global variables can be removed with the action **"Remove Agent's Global Variable"**.

Where to use a Global Variable

All global variables can be used in any policy in the following way:

- As **"usual variable"** like pre-defined variables e.g. <$globalvar>. It can be used to modify the message text or to set indication attributes.
- **Auto-activate** or **Auto-deactivate policies** on the fly. A policy can be configured to be active only if a Global Variable is defined and its value matches with a specified pattern.

5.3.3.2. Supported Variables

Variable	Monitor Policies	Indication Policies	Description
<$NAME>	Yes(1)	Yes	The name of the monitor or indication policy name. Can be used in a "Call" field and will be resolved during the initialization. (1) - Java Monitors receive their name automatically.
<$AGENT_IP>	Yes	Yes	IP address of a *boom* agent
<$AGENT_ID>	Yes	Yes	universal unique ID of a *boom* agent
<$AGENT_HOST>	Yes	Yes	hostname of a *boom* agent
<$HOST>, <$MSG_NODE_NAME>, <$MSG_NODE>	Yes	Yes	Submitted node name or hostname of a *boom* agent.
<$OBJECT>, <$MSG_OBJECT>	Yes	Yes	Object
<$SEVERITY>, <$MSG_SEV>	Yes	Yes	Severity (string)
<$APPLICATION>, <$MSG_APPL>	Yes(2)	Yes	Application (2) Used value that is specified in a monitor policy.
<$GROUP>, <$MSG_GRP>	Yes(3)	Yes	Group (3) Used value that is specified in a monitor policy.

Variable	Monitor Policies	Indication Policies	Description
<$ORIG_TEXT>, <$MSG_ORIG_TEXT>, <$MSG_TEXT>	No	Yes	Original submitted text.
<$OPTIONS>	Yes	Yes	Prints all optional variables that are submitted by a monitor or parsed by an indication policy in a form: varName=value ...
<$THRESHOLD>	Yes	No	Threshold value of the matched condition.
<$VALUE>	Yes	No	Monitor value
<$other_optional_variables>	Yes	Yes	Any other optional variables submitted with monitor value or messages. Also variables parsed in the message text by pattern specified in indication policy.
<$ORIG_VALUE>	STDDEV only	No	Original monitor value
<$DA_AVERAGE>	STDDEV only	No	The Mean of previuosly observed values
<$DA_STDDEV>	STDDEV only	No	Standard deviation value
<$DA_COUNT>	STDDEV only	No	Count of previous observations

Where to use

Please see the above support matrix to find out whether a variable is supported by monitor and/or indication policies. The supported variables can be used in the "Text"/"Set Text", "Indication Key", "Close Mask", "Auto Action" and "Operator Action" fields of a policy as well as in custom attributes.

<$NAME> can be also used in a "Call" field as a notification to an executable or a script about the expected monitor name.

All submitted optional variables can be used in a form: <$varname>

The *boom* server and agent truncates most indication attributes to a maximum size of characters. For detailed information see chapter Indication Attribute Sizes.

5.3.3.3. Supported pre-process Functions

Overview of the supported pre-process Functions

Function	Monitor Policies		Indication Policies		Description
	EXEC	JAVA	NAGIN	JAVA	
<$BOOMMON_ONINIT(...)>	Yes	Yes	Yes	Yes	Function that calls an external executable or script during the initialization of a Policy. Multiple calls in the same policy are allowed
<$BOOMMON_ONSTART(...)>	Yes	No	Yes	No	Function that calls an external executable or script before the scheduled run. Multiple calls in the same policy are allowed

Function	Monitor Policies		Indication Policies		Description
<$LOGFILES(...)>	No	No	Yes	No	Function that calls an external executable or script during the initialization of a Policy. Multiple calls in the same policy are allowed
<$BOOMMON_OBJECTS("list of objects")>	Yes	No	Yes	No	A comma separated list of values that are used for repeatable calls and override "Object" field of an Indication. Multiple calls in the same policy are [red]#No#T allowed
<$BOOMMON_OBJECTS(file:spi/filename)>	Yes	No	Yes	No	The specified file will be read to identify the objects as described above.
<$BOOMMON_[red]#No #DES("list of [red]#No#des")>	Yes	No	Yes	No	A comma separated list of values that are used for the repeatable call and override "Host" field of an indication. Multiple calls in the same policy are [red]#No#T allowed
<$BOOMMON_[red]#No #DES(file:spi/filename)>	Yes	No	Yes	No	The specified file will be read to identify the hosts as described above

Detailed description of the supported pre-process Functions

- **Function <BOOMMON_ONINIT(<command>):**
  This function returns the output of the executable it has been called with. The function can be used as an executable in a policy or as a part of it. It will only be resolved after the first policy initialization (during deployment of the policy) or after restart of the *boom* agent.

  For example:

  ```
  "<BOOMMON_ONINIT(hostname)>" will return the hostname of the agent:
  "HTTPMonitor.exe http://<$BOOMMON_ONINIT(hostname)>:<$BOOMMON_ONINIT(get_port.sh)>/"
  After initialization and execution the functions will be resolved:
  HTTPMonitor.exe http://hostname.domain.net:1234/
  ```

- **Function <$BOOMMON_ONSTART(<command>)>:**
  This function returns the output of the executable it has been called with. The function can be used as an executable in a policy or as a part of it. It will be resolved on every start of the policy processing.

  For example:

  ```
  "<BOOMMON_ONSTART(hostname)>" will return the hostname of the agent:

  "HTTPMonitor.exe http://<$BOOMMON_ONINIT(hostname)>:<$BOOMMON_ONINIT(get_port.sh)>/"
  After initialization and execution the functions will be resolved:
  HTTPMonitor.exe http://hostname.domain.net:1234/
  ```

- **Function <$LOGFILES(<command>)>:**
  This function returns the output of the executable it has been called with. The function can be used as an executable in file mask of a logfile policy or as a part of it. It will be resolved on every start of the policy processing and its return values are used as file paths for the policy. The result code of the query must have an

exit code of 0. Note: In general it is recommended to use the standard syntax with * or ** to define file paths.

For example:

```
"<$LOGFILES(sh -c "ls -ld /path/to/example/dir")>" will return all paths of files inside the specified
directory:
```

- **Function <$BOOMMON_OBJECTS("list of objects"|file:<filename>)>:**
  This function can be used in an execution call to repeat the execution of a command or a script for every object contained in the list without using a loop. The specified objects will override the "Object" attribute of the indication. For the use of the "file:" option, see below the description in $BOOMMON_[red]#No#DES.

  For example:

  ```
  ifconfig <$BOOMMON_OBJECTS("eth0,eth1,eth2")> will execute the command "ifconfig" for each interface
  specified in the list.
  ```

- Function <$BOOMMON_[red]#No#DES("list of [red]#No#des"|file:<filename>)>:
  This function can be used in an execution call to repeat the execution of a command or a script for every [red]#No#de contained in the list without using a loop. The specified [red]#No#des will override the "Host" attribute of the indication.

  For example:

  ```
  ping -c 5 <$BOOMMON_[red]#No#DES("server1, server2,server3")> will execute the command "ping" for each
  [red]#No#de specified in the list.
  ping -c 5 <$BOOMMON_[red]#No#DES(file:spi/hosts.cfg)> will execute the command "ping" for each
  [red]#No#de specified in the file <boom agent dir>/spi/hosts.cfg
  ```

- **Using the "file:" option:**
  The file is read to retrieve a list of [red]#No#des or objects that will be used in the monitoring call. Each line can contain one value or multiple values separated by comma ",". If necessary values can be enclosed in double quotes.

  For example:

  ```
  object1
  object2, "object 3", object4
  object5
  ```

  Results in the list: `object1,object2,object 3,object4,object5`

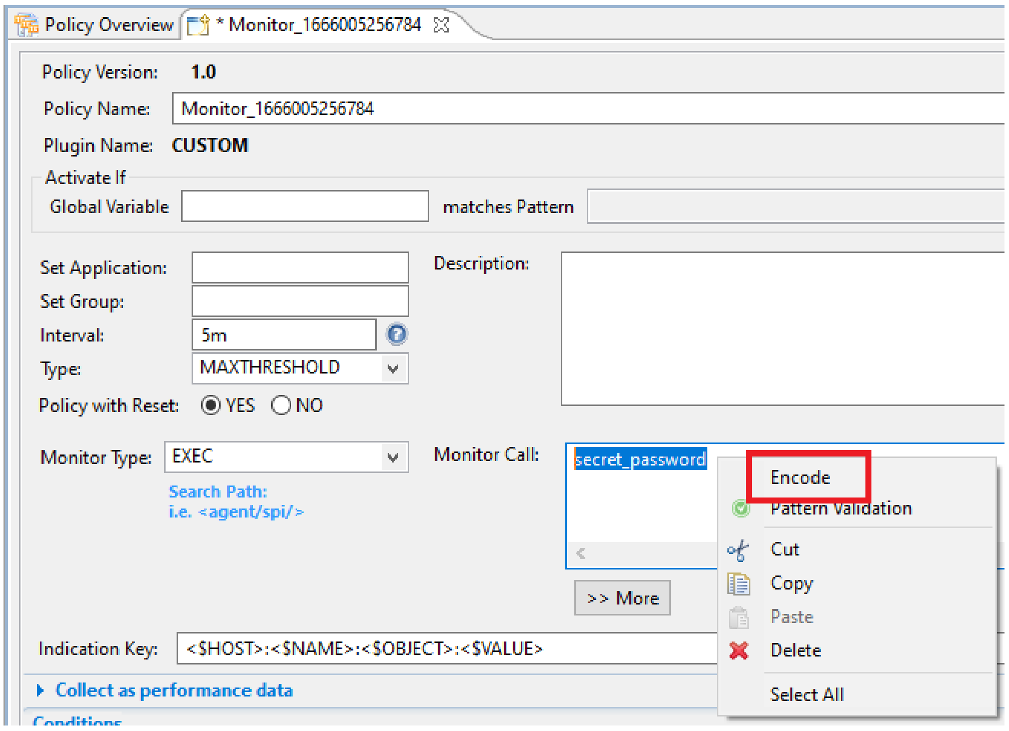
  The file location is relative to the *boom* agent installation path. If the file is deployed inside a package via the *boom* server, the file will reside relative to the `<boom agent>/spi` directory, e.g. `spi/my[red]#No#des.cfg`. The file path can also be specified as absolute path e.g. `/etc/boom/my[red]#No#des.cfg`
  The Agent will report any problem detected with the file as error message, e.g. file [red]#No#t found or access denied.

5.3.4. Hiding Cleartext Passwords

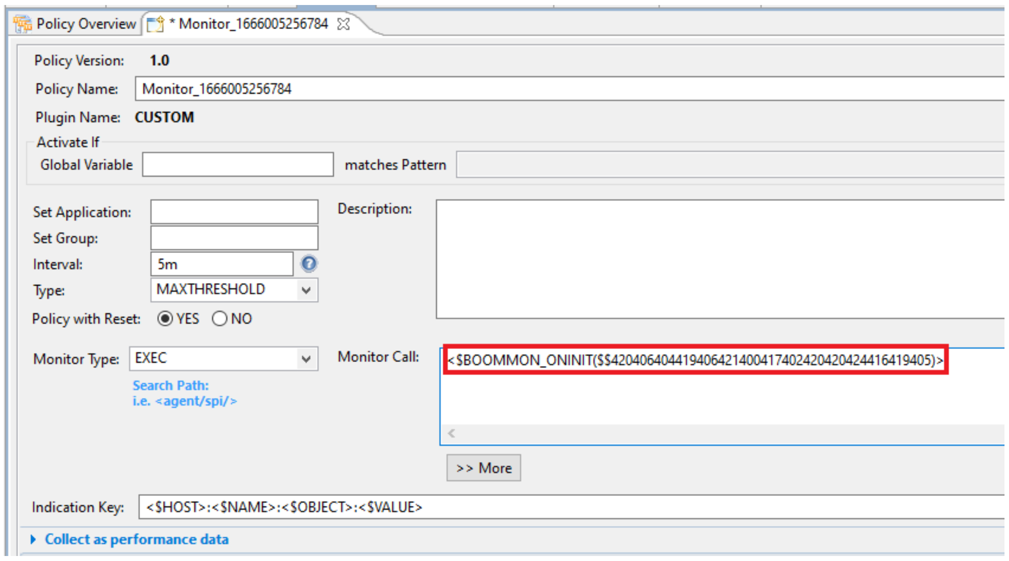
The BOOM GUI provides a mechanism to encrypt hardcoded passwords during policy creation. This mechanism is binhex based and only to prevent passwords from being read by others.

Currently, not every policy type does support the "encode" function, therefore a mock-policy has to be created to encode a password. The mock policy can be deleted after copying the encoded policy.

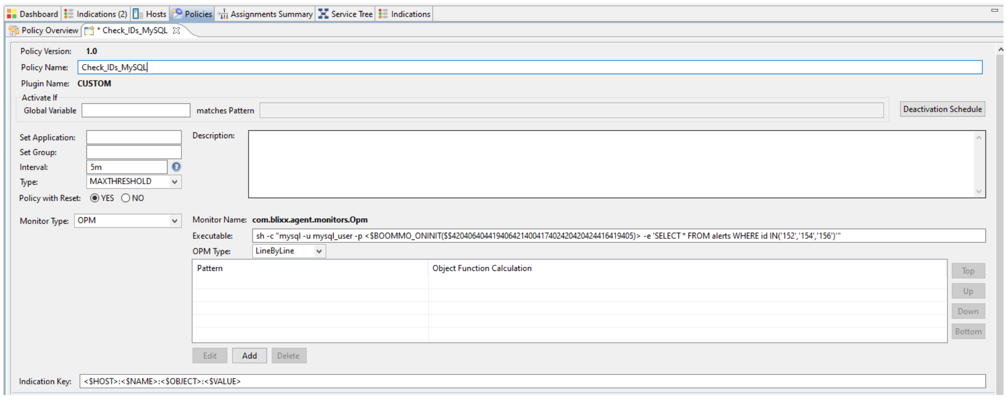
Step 1: Create a Monitor Policy of type Exec
Step 2: Write the password or any text to encrypt into the field "Monitoring Call"
Step 3: Mark the text to encrypt and right-click onto it
Step 4: Select "encode"
Step 5: Copy the resulting code including the <$BOOMMON_ONINIT($)> into any monitoring call



Policy Version: 1.0
Policy Name: Monitor_1666005256784
Plugin Name: CUSTOM
Activate If
Global Variable matches Pattern
Set Application:
Set Group:
Interval: 5m
Type: MAXTHRESHOLD
Policy with Reset: [✓ YES] NO
Monitor Type: EXEC
Search Path: i.e. <agent/spi/>
Description:
Monitor Call: secret_password
Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$VALUE>



Policy Version: 1.0
Policy Name: Monitor_1666005256784
Plugin Name: CUSTOM
Activate If
Global Variable matches Pattern
Set Application:
Set Group:
Interval: 5m
Type: MAXTHRESHOLD
Policy with Reset: [✓ YES] NO
Monitor Type: EXEC
Search Path: i.e. <agent/spi/>
Description:
Monitor Call: <$BOOMMON_ONINIT($$4204064044194064214004174024204204244416419405)>
Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$VALUE>



Policy Version: 1.0
Policy Name: Check_IDs_MySQL
Plugin Name: CUSTOM
Activate If
Global Variable: matches Pattern:
Set Application:
Set Group:
Interval: 5m
Type: MAXTHRESHOLD
Policy with Reset: [✓ YES] NO
Description:
Monitor Type: OPM
Monitor Name: com.blixx.agent.monitors.Opm
Executable: sh -c "mysql -u mysql_user -p <$BOOMMO_ONINIT($$42040640441940642140041740242042042416419405)> -e 'SELECT * FROM alerts WHERE id IN('152','154','156')'"
OPM Type: LineByLine

Pattern	Object Function Calculation

Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$VALUE>

5.4. Assignments

Assignment Concept

An assignment is a concept focused on grouping policies and binaries packages and their deployments based on the system role. Any managed system can be categorized in roles by many attributes. For example: OS type, kinds of applications running, is it a productive or a test systems, server classes and so on.

Different kinds of roles have to be monitored in a different way. Thus the IT department requires multiple set of policies and related binaries for every kind of role or aspect of the system. The assignment is a container that enables to create such sets.

Assignments provide several benefits:

- They allow to create a group of policies, monitor executables and actions that belongs to a specific aspect of the monitored system.
- One-step deployment of all child elements.
- One-step un-deployment.
- Automatic enqueue deployments or un-deployments of removed, added or renamed elements.
- Allow to check consistency of the deployed elements on remote agents.

Deployment of an assignment on a low level means deployment of all polices and binary packages contained in selected assignment. Logical tree folder structure has no relation to the deployment. Please note that *boom* agent has no information about deployed assignment groups, but knows about all deployed policies. To provide a status of assigned elements the *boom* server uses internal data stored in the database, xml configuration files and queries to remote agents. This allows to show the real picture of differences between agent's reality and server's expectations.

The handling of binary packages is simplified at the moment. The _boom agent does not track binary packages deployed on the node, so the server operates only with expectations. Extended tracking of binary packages is targeted to next release._
All assignments that are deployed on a remote *boom* agent are stored in the *boom* database. An agent can have multiple assignments deployed.

Assignment Group "BOOM" or "BOOM_Basics"

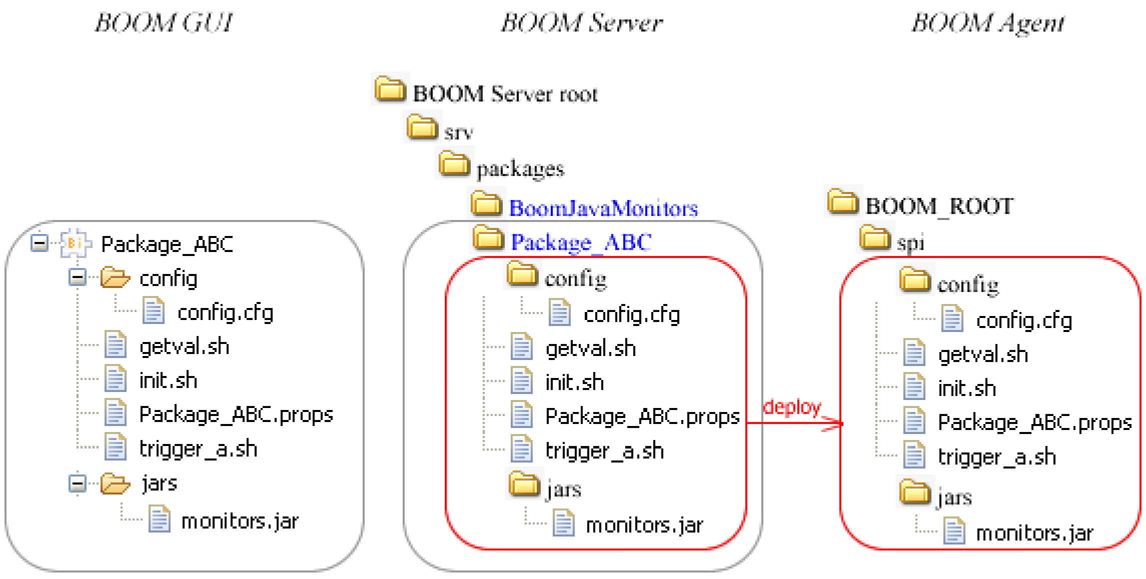
For full agent functionality, deploy the assignment group "BOOM" or "BOOM_Basics" to the agent, which contains the policy "BOOM_Messages" for internal messages and the package "BoomJavaMonitors" for agent specific Java classes.

Please read related sections:

Assignments
Assignments Summary

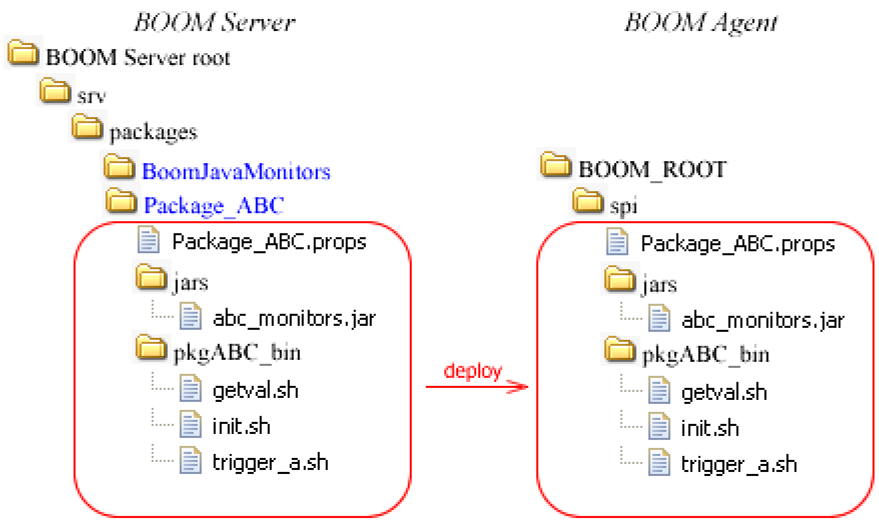
5.5. Binary Packages

Scripts, executables and configuration files that belong to one monitoring area or to one management plug-in are grouped together in so called "Binary Packages". These packages are directly reflected on the file system structure of the *boom* server.



As you can see from the picture, the content of the package (Package_ABC folder) will be deployed to the default package location on the agent (%BOOM_ROOT%/spi/) without the package name. Any subdirectory structure below the package folder will be reflected 'as is' on the agent side. The %BOOM_ROOT%/spi directory is the default working directory for all monitor calls and triggers that are declared in the policies.

Please note: The files in the different binary packages must have unique names to avoid conflicts. Therefore it is recommended to place these files into a uniquely named subfolder below the package name directory. Just the package property file should be placed in the top level folder (Package_ABC/) and the Java monitors have to be placed in the predefined location (Package_ABC/jars/).



The *boom* GUI reflects the file structure that is stored on the *boom* server system (srv/packages/). For the creation of a new binary package it is enough to create a new directory under srv/packages/. Changes on the server side are automatically reflecting in the *boom* client. The *boom* GUI also provides helper file oriented actions in the packages view. Together with the integrated editor that allows editing the server side files it helps to maintain packages even without direct access to the *boom* server.

The deployment on single file level is not supported. You can deploy the binary packages directly or as part of an assignment.

The renaming of the top package folder directly on the file system will cause problems in the assignment tree and on the stored assignments. Please use the *boom* GUI action "Rename package" for this purpose.

During the deployment of a package, the content of the package folder will be copied to the remote agent system.

All Java monitors (if any) in a package must be delivered as .jar files and must be placed in:

```
srv/packages/Package_ABC/jars/*.jar
```

The *boom* agent automatically recognizes the jar files and loads them without a restart.

If the package contains the property file named <packageName>.props - the *boom* agent will try to perform special processing steps during the deployment. This file can be used for defining STOP and START commands when the package has a stand-alone collector running separately from the agent.

```
# Package_ABC property file
# command to start collector (relative to %BOOM_ROOT%/spi/)
START=pkgABC_bin/init.sh -start
# command to stop collector (relative to %BOOM_ROOT%/spi/)
STOP=pkgABC_bin/init.sh -stop
# timeout in seconds for both commands
TIMEOUT=30
```

When an agent recognizes the package property file, it will execute the START/STOP commands when the *boom* agent gets started or stopped. During the package distribution the agent will execute the STOP command, (re)places files and triggers START as the last step. This allows replacing the files without running into trouble with files that are locked by a running collector.

If a binary package is no longer needed on the agent side, the deployed files can be simply removed.

5.6. Performance Data

5.6.1. Overview

A *boom* Agent can collect and forward performance data to the server where data is persisted in BOOM_PERF database according to performance class name included with each performance measurement.
For each incoming performance class server creates a separate table in the database. The performance class table will be automatically created/altered to store all incoming data.

i.e. if Agent1 sends performance record for class "C" with fields ("A","B") and Agent2 sends performance record for class "C" with fields ("A","Z") - server will ajust table "C" to have columns: "A", "B", and "Z" to be able to store data from both Agents.

Each incoming record automatically enriched by "_TIME" and "BOOMAGENTID" columns.

- "*TIME" - milliseconds since epoch when performance record was received by the _boom* agent.
- "BOOMAGENTID" - ID of the Agent that forwarded this record to the boom server.

The boom server provides several possibilities to view performance data:

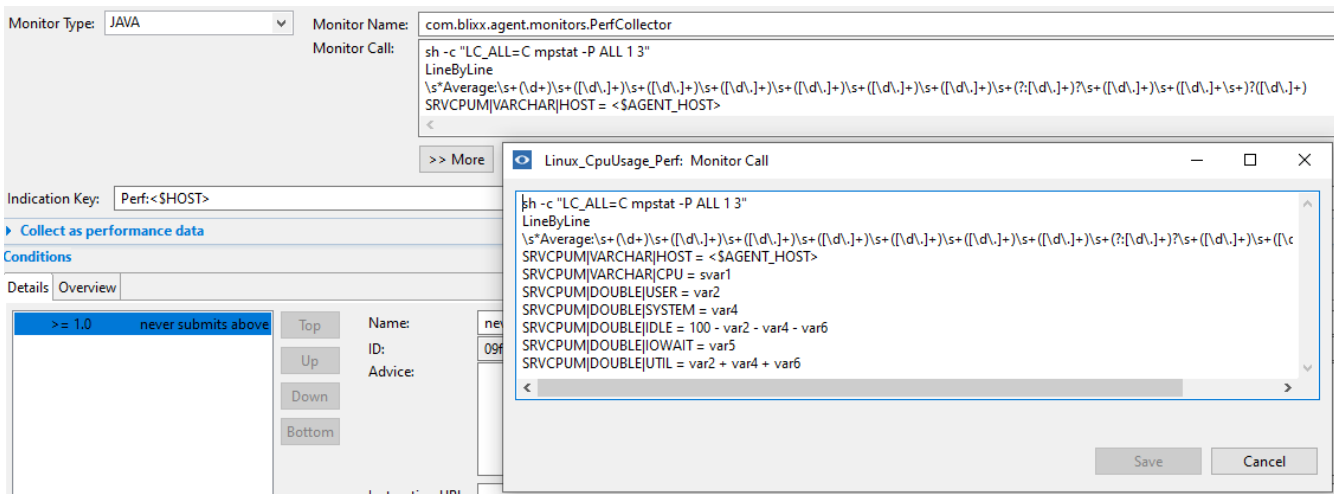
- Simple web page performance data browser (see Performance Data)
- Dashboards (see chapter: Dashboards)
- Grafana integration (contact us: Contact)

Performance Data Handling

The *boom* supports the submitting of custom performance data using performance data collecting policies, management plug-ins (MPI) and external sources.

Performance Data Collecting Policies

Boom allows definition of policies which collect performance data and forward it to the server to be stored in the performance database.



Monitor Type: JAVA

Monitor Name: com.blixx.agent.monitors.PerfCollector

Monitor Call:

```
sh -c "LC_ALL=C mpstat -P ALL 1 3"
LineByLine
\s*Average:\s+(\d+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+(?:[\d\.]+)?\s+([\d\.]+)\s+([\d\.]+\s+)?([\d\.]+)
SRVCPUM|VARCHAR|HOST = <$AGENT_HOST>
```

Indication Key: Perf:<$HOST>

Linux_CpuUsage_Perf: Monitor Call

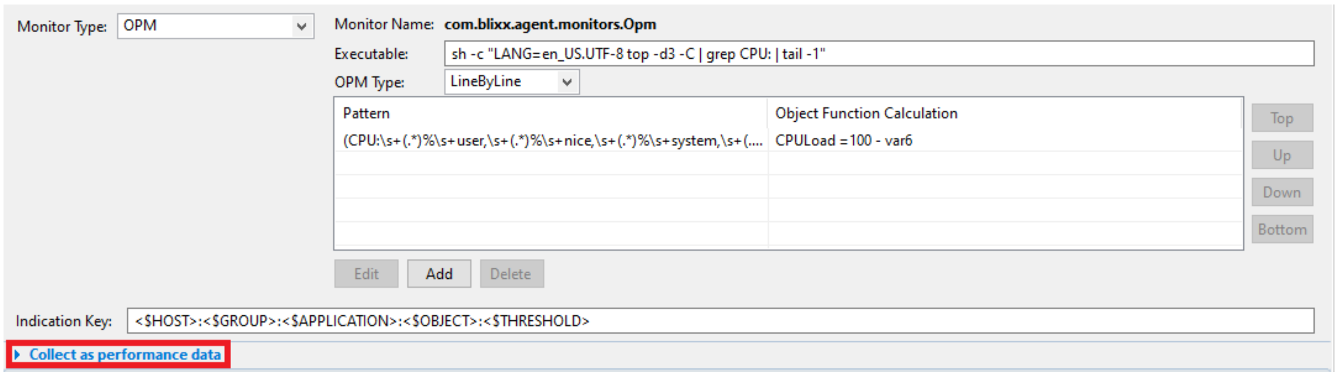
```
sh -c "LC_ALL=C mpstat -P ALL 1 3"
LineByLine
\s*Average:\s+(\d+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+([\d\.]+)\s+(?:[\d\.]+)?\s+([\d\.]+)\s+([\c
SRVCPUM|VARCHAR|HOST = <$AGENT_HOST>
SRVCPUM|VARCHAR|CPU = svar1
SRVCPUM|DOUBLE|USER = var2
SRVCPUM|DOUBLE|SYSTEM = var4
SRVCPUM|DOUBLE|IDLE = 100 - var2 - var4 - var6
SRVCPUM|DOUBLE|IOWAIT = var5
SRVCPUM|DOUBLE|UTIL = var2 + var4 + var6
```

To configure such policy please create a *General Java* policy, choose *JAVA* as monitor type and *com.blixx.agent.monitors.PerfCollector* as monitor name. Monitor Call field is structured as follows:

- First line defines the command which produces the output containing values to be used as performance measurements.
- Second line defines the OPM policy processing style (LineByLine).
- Third line is a regular expression to be used to parse the command output.
- Next lines define the measurement which is to be stored in a performance database as a single row.
  These lines follow the format:
  <Performance Class>|<Column type>|<Column Name> = <variable | expression>

Monitor Policies with Performance Collection

There is also a possibility to enhance existing threshold policies to submit performance data also.



Monitor Type: OPM

Monitor Name: **com.blixx.agent.monitors.Opm**

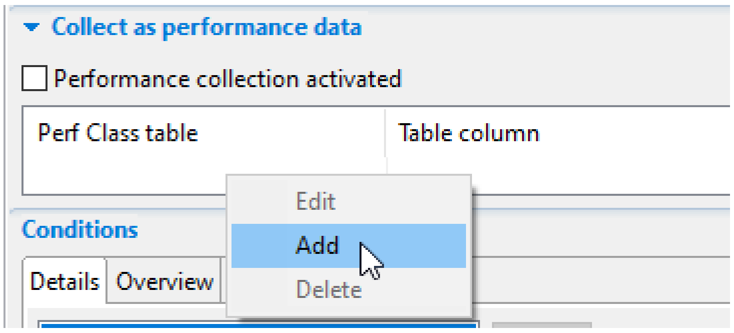
Executable: sh -c "LANG=en_US.UTF-8 top -d3 -C | grep CPU: | tail -1"

OPM Type: LineByLine

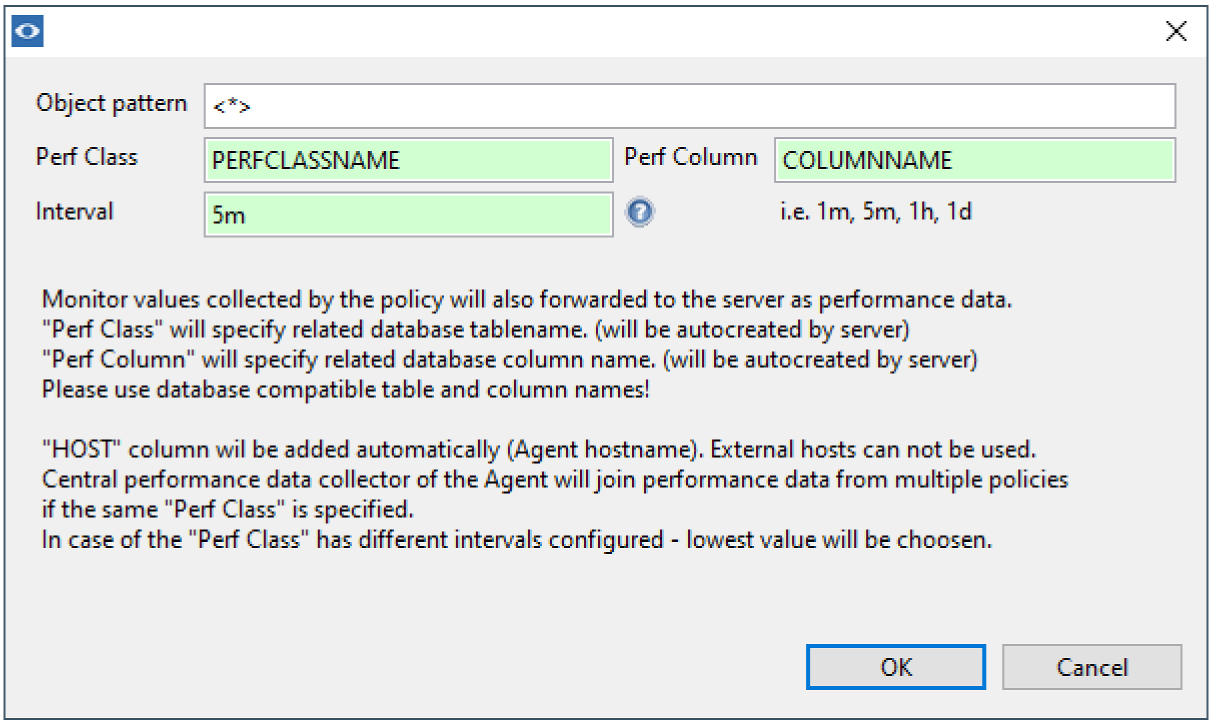
Pattern	Object Function Calculation
(CPU:\s+(.*)%\s+user,\s+(.*)%\s+nice,\s+(.*)%\s+system,\s+(....	CPULoad =100 - var6

Indication Key: <$HOST>:<$GROUP>:<$APPLICATION>:<$OBJECT>:<$THRESHOLD>

Expanding the section *Collect as performance data* you will get to the following screen where you can add a performance data collection settings by pressing right mouse button and selecting *Add*.



In this dialog you can specify which objects should be considered for performance data collection, the storage table name, table column and the interval for performance data collection.Also don’t forget to check the *Performance collection activated* checkbox to enable this feature.



Object pattern: <*>
Perf Class: PERFCLASSNAME
Perf Column: COLUMNNAME
Interval: 5m
i.e. 1m, 5m, 1h, 1d

Monitor values collected by the policy will also forwarded to the server as performance data.
"Perf Class" will specify related database tablename. (will be autocreated by server)
"Perf Column" will specify related database column name. (will be autocreated by server)
Please use database compatible table and column names!

"HOST" column wil be added automatically (Agent hostname). External hosts can not be used.
Central performance data collector of the Agent will join performance data from multiple policies
if the same "Perf Class" is specified.
In case of the "Perf Class" has different intervals configured - lowest value will be choosen.

5.6.2. Agent command based performance data collection

In order to submit performance data via Agent command, one must first register a performance class declaration with the *boomperfreg* script.

Usage:

boomperfreg full/path/to/file.spec

Specification file format:

```
CLASS <className>

<fieldname1> = <positionNumber>
BOOM_DATATYPE DOUBLE

<fieldname2> = <positionNumber>
BOOM_DATATYPE BIGINT

<fieldname3> = <positionNumber>
BOOM_DATATYPE VARCHAR 128 ...
```

Supported types:

- DOUBLE - the main type for performance values
- BIGINT - represents long values (can be used for time)
- VARCHAR <size> - any string values

After the specification file was processed by the agent, the MPI can submit the performance data by using the boomperfstore script.

Usage:

boomperfstore <classname> <values separated by space>

If the specification file is not registered, the submitting of performance data will be rejected. If the number of submitted values is different from the number of fields that are registered with spec file - the submitting of performance data will also be rejected.

The *boom* server automatically alters the appropriate database table if a new version of the class specification was submitted by any system. But the table keeps the old field untouched and adds only not existing fields from the new registered specification. This allows to store data from multiple systems even if they are operating with different versions of the performance class.

Local Interprocess Communication between the agent and the scripts submitting data

The scripts **boomperfreg** and **boomperfstore** are using **telnet or netcat** on some platforms (HP-UX, Solaris) to communicate with the local agent. Netcat is faster, more reliable and has less overhead compared to telnet. Netcat isn't available on all platforms per default. If you want to use netcat instead of telnet you have to install the netcat utility.

The agent installation script checks if netcat is available on the agent and creates the file <install directory >/spi/**netcat.sh**. If netcat is available the installation script sets the flag USE_NC to 1 or 2 (**USE_NC=1** to use the nc command, **USE_NC=2** to use **netcat**) in that file. Otherwise the flag is set to **USE_NC=0** to use **telnet**. This flag can be changed on the fly, an agent restart is not necessary.

5.6.3. Ignore Performance Classes on server

Sometimes it is necessary to ignore some of the performance classes coming to the server. Please use srv/etc/ignorePerfClasses.dat file to configure list of performance classes that needs to be ignored. The format is one classname per line:

```
PerfClass1
PerfClass2
```

To avoid restart of server, you can use server action "Reload ignorePerfClasses.dat" (RELOAD_IGNORE_PERF_CLASSES_FILE) to refresh list on the running server.

5.6.4. Cleaning data

Boom server has a server job ("CleanPerf DB") that responsible for cleaning old performance records. See Server Jobs.

5.7. Actions

Actions are used to execute predefined commands or scripts on the *boom* agents or the *boom* server for maintenance reasons, diagnostics, problem resolution, or task automation.
Beside the benefit of improving reaction times and avoiding typos the actions can be used to support advanced security concepts. Action groups can be enabled/disabled for certain user roles. In combination with agent filters it

is possible to define which role is allowed to execute which commands on which servers. Users with administrative rights can extend and customize the available actions and groups, so that a very dedicated user concept can be implemented and a manual login of the users on the monitored systems is no longer necessary.

Following types of actions are available:

Action Type	Description
Remote Action	Any script, command or Java class can be executed on one or more remote system(s) where the *boom* Agent is installed. Several actions are pre-installed and available in the *boom* GUI in the Actions Panel. Existing *boom* actions can be customized or new actions can be created to meet the own requirements.
Server Action	Every Remote Action is also executable on the *boom* server. In addition there are some server specific actions pre-installed in the Actions Panel in the *boom*-GUI. These actions can be customized or extended by own actions. *boom* server actions will only be executed on the *boom* server.

Remote Actions

Usual Remote Actions are based on a script or executable located on a remote system. These actions can be used for multiple purposes. The main purpose is corrective actions and system or application tools that are able to deliver more detailed status information from the managed node. Any scripts or executables that are not available by default must be first deployed to the system by using the Binary Packages.
In the ***boom*** GUI a separate **Actions View** exists for the stored remote actions that can be used by operators and administrators as instrumentation repository.
All action groups are stored as XML files on the filesystem on the *boom* server.
All pre-installed actions and all available remote action operations including import and export are described in chapter Actions (GUI part).

If pipes, semicolons, redirection etc. are used the following call is mandatory:

on Unix:

```
sh -c "<command>"
```

on Windows:

```
cmd /C "<command>"
```

This will correctly quote your command

If Java Classes are used the action has to be executed as follows:
`IAction <full java class name> <parameters>`

A "Java Action" is an action that is implemented as a Java class and deployed to the *boom* agent system as a jar file. Usually it’s part of a binary packages. Such classes must implement a simple *com.blixx.ext.IAction* java interface. All pre-defined "Java Actions" are already implemented as samples e.g. the JMX actions.

Server Actions

Server Actions are working in the same way as the Remote Agent Actions. A description of all pre-installed server actions is available in chapter Actions (GUI part)

5.8. Submitting Monitor Values / Indications

5.8.1. Submitting Monitor Values

Monitors triggered by the *boom* agent can submit their monitor values by printing to STDOUT or use the **boommon** executable or script that is supplied with the *boom* agent. External or asynchronous monitors always need to use **boommon**.

STDOUT format:

monitorName=value [(o|object)=objectName] [varName=varValue]*

where

monitorName

The name of the Monitor as specified in the deployed Policy. If a Policy was not deployed on the *boom* agent the submitted value will be rejected.

value

The numeric value of the monitor.

objectName

Object string

varName

Optional variable name.

varValue

An optional variable string value.

*It’s possible to submit multiple optional variables separated by space.
*If objectName or varValue contain spaces they must be quoted with "" character.

The **boommon** command syntax is similar to STDOUT format:

```
boommon monitorName=value [(o|object)=objectName] [varName=varValue]*
```

Example:

```
boommon ExternalMonitor_test1=44,5 o=testObject1
```

5.8.2. Submitting Indications

To submit messages use the **boomindi** command supplied with the *boom* agent.

```
boomindi    ( a|application)=applicationName
            [( o|object)=objectName]
            [( g|group)=groupName]
            [( s|severity)=severityStr]
            [( h|host)=hostName]
            [varName=varValue]*
            [ POLICY_NAME=policyName]
            ( t|text)=message
```

where

application

(required) application string

object

(optional) object string

group

(optional) group string

severity

(optional) one of "unknown","normal","warning","minor","major","critical"

host

(optional) hostname if different from the agent’s hostname

varName

(optional) variables (zero or more)

POLICY_NAME

(optional) The agent uses only the specified policy to match the incoming message.

text

(required) message text

*It’s possible to submit multiple optional variables separated by space.
*If string values contain spaces they must be quoted with "" character. Example:

```
boomindi a="Application 1" o=Object1 s=normal optVar1="BS01_12" t="This is a messaget text"
```

5.8.3. Submitting Performance Data

To submit performance data via boom action use the **boomperfstore** command supplied with the *boom* agent. One must first register a performance class declaration with the *boomperfreg* script, before actually submitting performance data with the **boomperfstore** command.

Usage:

boomperfreg full/path/to/file.spec

Specification file format:

```
CLASS <className>

<fieldname1> = <positionNumber>
BOOM_DATATYPE DOUBLE

<fieldname2> = <positionNumber>
BOOM_DATATYPE BIGINT

<fieldname3> = <positionNumber>
BOOM_DATATYPE VARCHAR 128 ...
```

Supported types:

- DOUBLE - the main type for performance values
- BIGINT - represents long values (can be used for time)
- VARCHAR <size> - any string values

Usage:

boomperfstore <classname> <values separated by space>

If the performance class is not registered or the number of values do not fit the ones defined in the performance class, the data submit is rejected.

5.8.4. Local Interprocess Communication

Local Interprocess Communication between the agent and the executables/scripts submitting data

The scripts **boomindi** and **boommon** are using **telnet or netcat** on some platforms (HP-UX, Solaris) to communicate with the local agent. Netcat is faster, more reliable and has less overhead compared to telnet. Netcat isn't available on all platforms per default. If you want to use netcat instead of telnet you have to install the netcat utility.

The agent installation script checks if netcat is available on the agent and creates the file <install directory >/spi/**netcat.sh**. If netcat is available the installation script sets the flag USE_NC to 1 or 2 (**USE_NC=1** to use the **nc** command, **USE_NC=2** to use **netcat**) in that file. Otherwise the flag is set to **USE_NC=0** to use **telnet**. This flag can be changed on the fly, an agent restart is not necessary.

5.9. Indication Reduction in *boom*

In *boom* there are several ways to accomplish indication reduction:

1. Duplicate Indication Suppression
2. Silence Count for Monitor Policies
3. Indication Correlation via Indication Key and Close Mask

5.9.1. Duplicate Indication Suppression

The *boom* server compares the indication attributes of incoming indications with the indication attributes of existing indications. The *boom* server increases the duplicate counter of an existing indication if both indications are identical.

Policy conditions can be configured to suppress duplicate indications in the following way:

De-Duplication parameter

- Detect Duplicate
  If this flag is set the following indication attributes must be the same to suppress an incoming indication: AGENT_HOST,HOST,APPLICATION,GROUP,OBJECT,SEVERITY,INDICATION KEY, MESSAGE TEXT
- De-Duplicate KeyOnly If this flag is set the INDICATION KEY attribute is used for recognizing duplicates.

It's not possible to disable duplicate detection if the CLOSE MASK is configured in the condition.

Timer / Counter parameter

- Interaction of Silence Count and Silence Time for Indication Policies

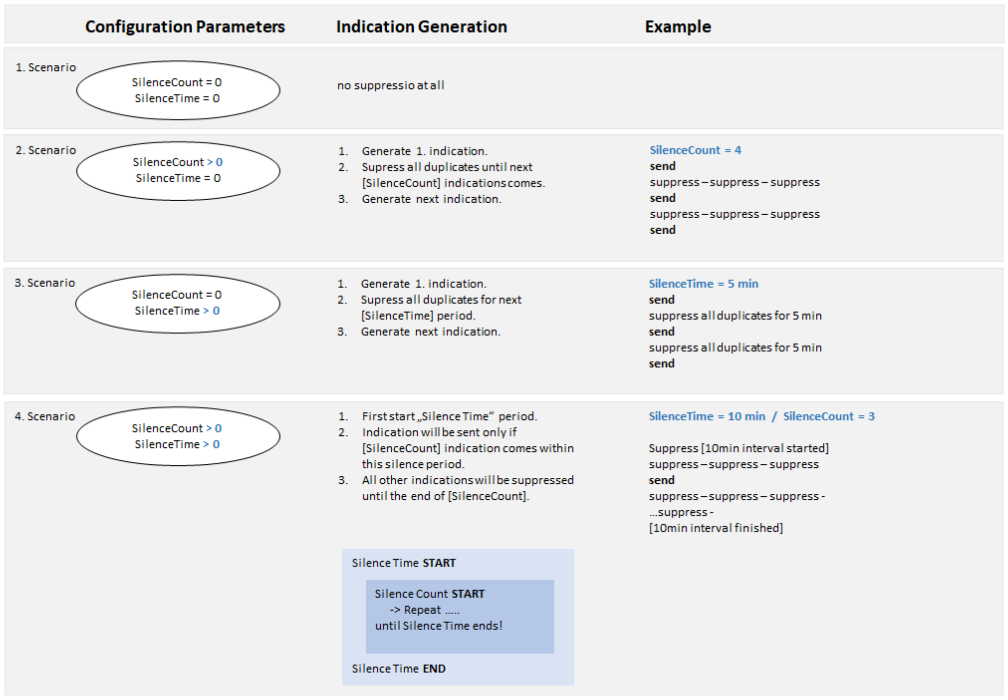
  The number of duplicates suppressed can be configured with the following parameters:

  Silence Time:

  Suppression time interval since the first match.
  The default setting for this parameter is 0.

  Silence Count:

  Suppress number of messages since the first Indication has been sent.
  The default setting for this parameter is 0.



Configuration Parameters		Indication Generation	Example

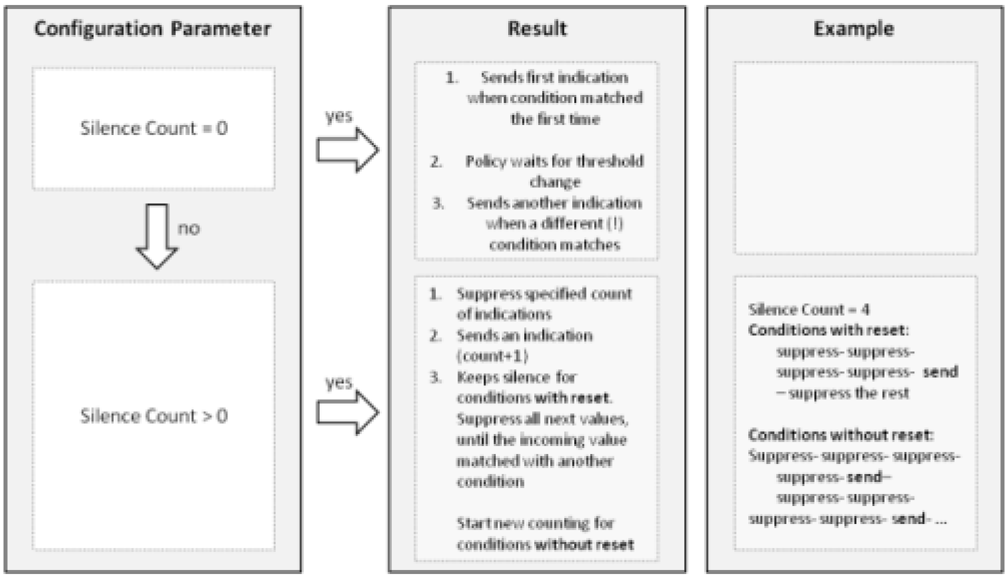
1. Scenario	SilenceCount = 0
	SilenceTime = 0
		no suppressio at all	

2. Scenario	SilenceCount > 0 SilenceTime = 0	1. Generate 1. indication. 2. Supress all duplicates until next [SilenceCount] indications comes. 3. Generate next indication.	SilenceCount = 4 **send** suppress – suppress – suppress **send** suppress – suppress – suppress **send**
3. Scenario	SilenceCount = 0 SilenceTime > 0	1. Generate 1. indication. 2. Supress all duplicates for next [SilenceTime] period. 3. Generate next indication.	SilenceTime = 5 min **send** suppress all duplicates for 5 min **send** suppress all duplicates for 5 min **send**
4. Scenario	SilenceCount > 0 SilenceTime > 0	1. First start „Silence Time“ period. 2. Indication will be sent only if [SilenceCount] indication comes within this silence period. 3. All other indications will be suppressed until the end of [SilenceCount]. Silence Time **START** Silence Count **START** -> Repeat ..... until Silence Time ends! Silence Time **END**	SilenceTime = 10 min / SilenceCount = 3 Suppress [10min interval started] suppress – suppress – suppress **send** suppress – suppress – suppress - ...suppress - [10min interval finished]

5.9.2. Silence Count for Monitor Policies

The **Silence Count** parameter of a condition can be used when it is necessary to suppress a couple of first generated values. The *boom* agent will ignore the specified amount of submitted values that match with the condition before an indication will be sent to the *boom* server.

Silence Count: A count of values since the first match of the threshold that must be suppressed and not delivered to the server. Used when it is necessary to ignore short and not important peaks.



Configuration Parameter	Result	Example
Silence Count = 0	1. Sends first indication when condition matched the first time 2. Policy waits for threshold change 3. Sends another indication when a different (!) condition matches	
Silence Count > 0	1. Suppress specified count of indications 2. Sends an indication (count+1) 3. Keeps silence for conditions **with reset**. Suppress all next values, until the incoming value matched with another condition Start new counting for conditions **without reset**	Silence Count = 4 **Conditions with reset:** suppress- suppress- suppress- suppress- **send** – suppress the rest **Conditions without reset:** Suppress- suppress- suppress- suppress- **send**– suppress- suppress- suppress- suppress- **send**- ...

Reset Condition State: Optional time interval that allows to reset Silence Count. The interval started when first matching value arrives.
During specified interval the agent can send only one indication if count reached. At the end of interval the Silence Count will be dropped and any new matching will re-initiate interval processing.

5.9.3. Indication Correlation via Indication Key and Close Mask

An indication key is used to define a logical correlation attribute that allows to define indications that belong together.This allows to define for example indications belonging to a certain monitored object, even the indications are generated by different conditions or even from different sources. It can be constructed using supported variables and a fixed text to describe the message event.

```
Example:  boom_test:<$AGENT_HOST>:<$APPLICATION>:<$OBJECT>:<$GROUP>
```

The Indication Key will be resolved during the processing of an incoming message. The Result string (any variables are resolved) is used for the correlation with the following Indication for auto acknowledgment of previous indications and duplicate suppression.

```
Example: boom_test:boom_host01:boom_appl01:boom_object01:boom_group01
```

The close mask attribute of a condition is a pattern that defines that all active indications with a matching indication key will be automatically closed, when this indication is received. It can contain supported variables as well as patterns and fixed text. Patterns are used to match a set of indication keys.

```
Example:  boom_test:<$AGENT_HOST>:boom_<*>:<$OBJECT>:<$GROUP>
```

When a message matches the condition, the values of the variables are substituted in the close mask, leaving a pattern to be used by the management server for comparing with the indication keys of previous indications.

```
Example: boom_test:boom_Host01:boom_<*>:boom_object:boom_group01
```

Indications with the following indication key are matched:

```
Example:  boom_test:boom_Host01:boom_appl01:boom_object:boom_group01
```

5.10. Heartbeat Polling

Both the *boom* server and the *boom* agent use heartbeats to ensure if a counterpart is reachable and alive.

On the socket level it can be represented by the following steps:

1. TCP connect: can fail with Connect Timeout (default 3 sec.)
2. Write CHAR
3. Read CHAR: can fail with Read Timeout - default 10 sec. or configurable by calling server action "SET_HB_RTIMEOUT" (ms)
4. Close connection

The heartbeat fails if a 'Connect' or 'Read Timeout' happens or the write fails. These 4 steps build **one heartbeat cycle**. The heartbeat cycle interval is set by default to 10 seconds.

5.10.1. Server Heartbeats

Not Firewalled Agent:

If the heartbeat cycle was marked as failed 3 times and the agent itself does not send a successful heartbeat the agent’s state will be set to OFFLINE:

If the agent’s state was set before to ONLINE (state change happened) the server generates

- An indication "Agent is offline"
- A notification to the GUI changing the host symbol to status "Agent is not running" 🟥

The first successful heartbeat to the Agent will change the state to ONLINE:

If the agent's state was set before to OFFLINE (state change happened) the server generates

- An indication "Agent is online"
- A notification to the GUI changing the host symbol to status "Agent

Firewalled Agent:

agent is running and agent is firewalled) the server will not try to send heartbeats and just tracks the last conversation time of incoming connections. This includes heartbeats from agent or any data transfers. If the last activity from the agent is older than 1 minute, the agent's state will be set to OFFLINE :

If the agent's state was set before to ONLINE (state change happened) the server generates

- An indication "Agent is offline"
- A notification to the GUI changing the host symbol to status "Agent

The first successful connection from the the agent will change the state to ONLINE.

If the agent's state was set before to OFFLINE (state change happened) the server generates

- An indication "Agent is online"
- A notification to the GUI changing the host symbol to status "Agent is

Also the agent will be switched from "Agent is not running" to status "Agent is OK"
if the server's heartbeat continues to fail but the agent communicates with the server without problems.

Activation of the server side heartbeat polling:

The activation of the server's side heartbeat polling can be controlled by the server:

- setting agent to Firewalled Mode = Deactivation (HB OFF)
- setting agent NOT to Firewalled Mode = Activation (HB ON)

Customization of server side heartbeat polling:

To customize the server side heartbeat polling (global setting operative to all agents) set the following parameters:

- Connect Timeout (default 3 sec) configurable by calling server action "SET_HB_CTIMEOUT" (ms)
- Read Timeout (default 10 sec) configurable by calling server action "SET_HB_RTIMEOUT" (ms)
- Heartbeat interval (default 10 sec) configurable by calling server action "SET_HB_INTERVAL" (s)

5.10.2. Agent Heartbeats

Failed heartbeat cycle:

If the heartbeat cycle was marked as failed the agent operates now in mode:

- Buffering all indications
  Sending of indications to the server will be deactivated until the next successful heartbeat
- If the heartbeats from the server are successfully retrieved and the active heartbeat failed for 3 heartbeat intervals the agent will inform the server with the heartbeat response to pull the data.

Successful heartbeat cycle:

If the heartbeat cycle was marked as successful the agent operates now in mode:

- Normal operation
  Agents actively send data to the server.

Depending on the agent’s Mode7 (see description below) the heartbeat of the agent can be activated or not. In "MODE7" the agent does not heartbeat the server and waits for server requests.

Activation of agent side heartbeat polling:

The activation of the agent side heartbeat polling can be controlled by server calling:

- "Set Agent →Server communication ON" = BOOM_AGENT SET_MODE0
  - Informs an Agent to communicate with the server. Default (HB ON)
- "Set Agent →Server communication OFF" = BOOM_AGENT SET_MODE7
  - Informs an agent to keep silence and waits for server requests (HB OFF)

Customization of agent side heartbeat polling:

To customize the agent side heartbeat (configuration settings inside "conf/agent.conf") set the following parameters:

- Connect Timeout (default 3 sec.) configurable by setting "HB_CTIMEOUT=" (ms)
- Read Timeout (default 10 sec.) configurable by setting "HB_RTIMEOUT=" (ms)
- Heartbeat interval (default 10 sec) configurable by setting "HB_INTERVAL=" (s)

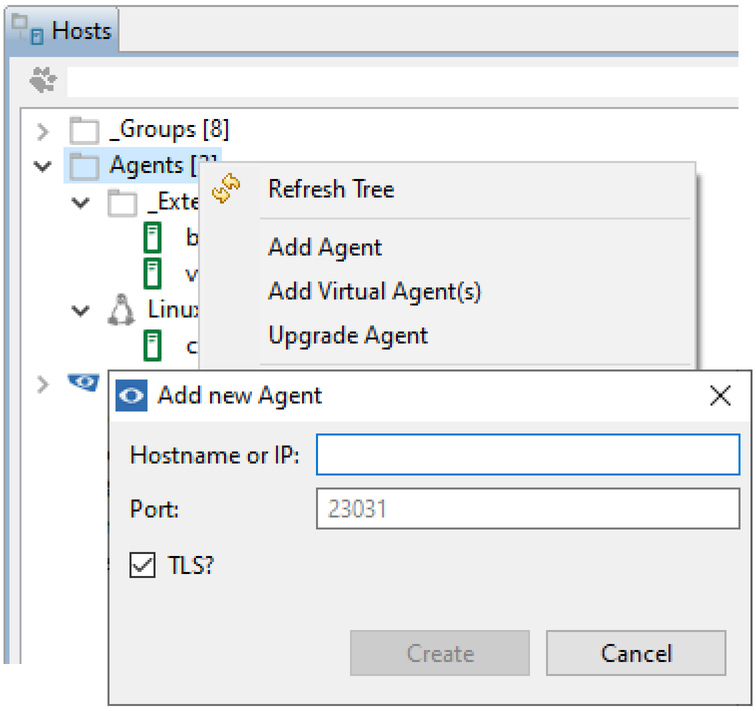
Setting an agent to **Firewalled Mode AND to Mode7** can be done in principle BUT causes that **NO more communication** takes place between the server and the agent. The server waits on agent requests e.g. providing incidents and the agent waits on server requests e.g. collecting incidents.

5.11. Agent Management

5.11.1. Add and Approve Agent

Add Agent:

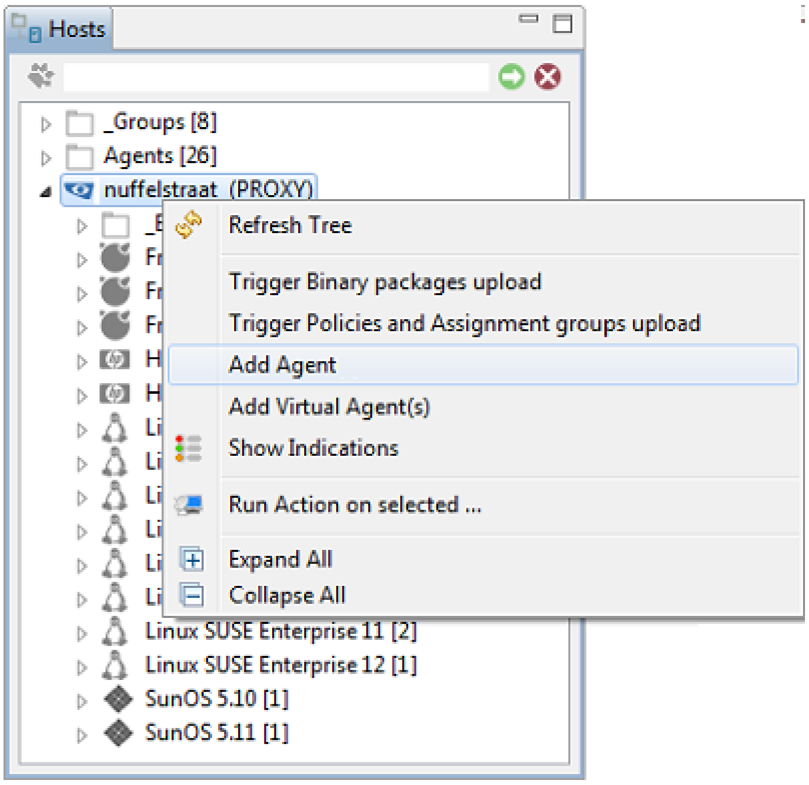
Typically there is no need to manually add agents, since they will register and appear automatically at the *boom* server. In environments with firewalls or special NAT configurations the communication direction from agent to the server might be blocked so that the agent is not able to communicate with the server. In this case the Agent need to be manually added to the *boom* server to initiate the communication from the server to the agent.



In the Add new Agent Dialog just enter an IP addresses or a resolvable host name. If necessary change port and TLS settings and press the OK Button. The server will add the agent and then tries to communicate to it like the agent has been self registered and approved.

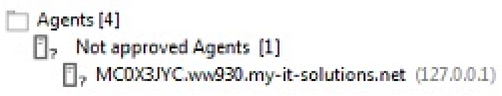
The default port for TLS connection is 23031 and for NON-TLS connection 23021.

In environments with Master and Slave servers the *boom* UI allows to add agent systems to a slave server, even if the UI is connected to the Master. Just right click on the Slave Server to add a new Agent.


Hosts
_Groups [8]
Agents [26]
nuffelstraat (PROXY)
Refresh Tree
Trigger Binary packages upload
Trigger Policies and Assignment groups upload
Add Agent
Add Virtual Agent(s)
Show Indications
Run Action on selected ...
Expand All
Collapse All
Linux SUSE Enterprise 11 [2]
Linux SUSE Enterprise 12 [1]
SunOS 5.10 [1]
SunOS 5.11 [1]

Approve Agent:

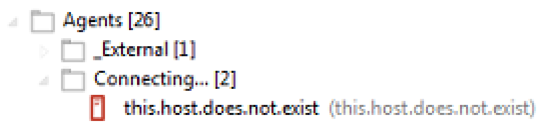
After the agent installation the agent registers itself at the *boom* server (Agent is waiting for approval).
The administrator has to approve the agent in the *boom* UI to become operational. If the system was manually added, the system does not need to be approved again.


Agents [4]
Not approved Agents [1]
MC0X3JYC.ww930.my-it-solutions.net (127.0.0.1)

- To approve an agent right click on the "not approved agent" and select "Approve Agent"
- A unique identification (agent ID) is assigned to the agent.
- The agent collects inventory data and sends these data to the server.
- The server identifies the operating system of the agent.
- An operating system specific host group is added to the agent tree [Agents [no. of OS specific host groups]] in the UI "Hosts" view (if not existing).
- The agent is added to the created or existing host group.
- Now the agent is operational.

Agents in Connecting... state:

After manually adding an agent system, the agent might not show up as expected in operating specific group. Instead there is suddenly a new host group named Connecting...


Agents [26]
_External [1]
Connecting... [2]
this.host.does.not.exist (this.host.does.not.exist)

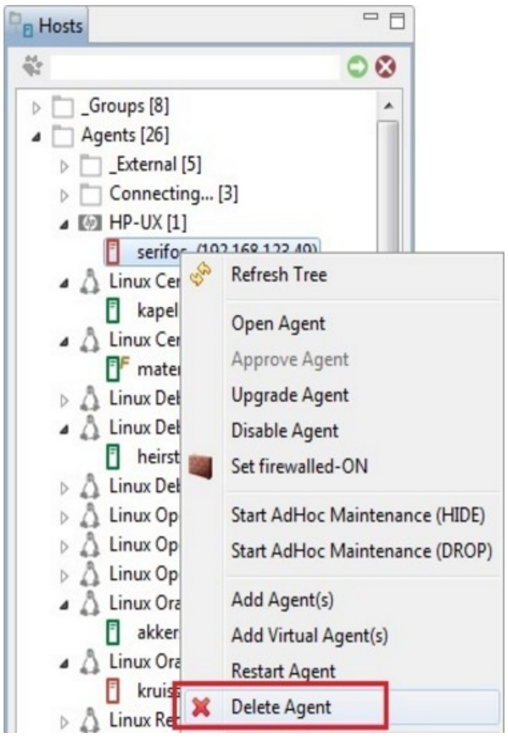
If the *boom* server can’t reach the newly added system it will be placed into this dynamic group that will vanish if there is no more system in this state. To solve this problem verify that the entered IP or name is correct, is resolvable, can be reached and the system and agent are running. As soon as the server is able to communicate with the agent and retrieves the basic information like the operating system, it will move the agent to correct host group.

5.11.2. Delete Agent

Delete Agent:

The "Delete Agent" operation is a functionality of the user interface! The agent is deleted from the UI "Hosts" view but is still installed on the node. Additionally the corresponding "agent card" for this agent is deleted from the server. The "agent card" contains all important information about the agent e.g. agent ID, which policies and assignments are deployed to the specified agent, etc.

To delete an agent from the UI right click on the agent and select "Delete Agent".



- The agent is no longer visible in the UI
- The "agent card" is deleted
- All links in the host groups are deleted
- All policy and assignment information about the specified agent is deleted (agent card)
- The agent specific indications are unchanged
- The deployed policies are **not** deleted from the agent (/opt/boom/agent/policies)
- The deployed packages are **not** deleted from the agent (/opt/boom/agent/spi)
- After the next heartbeat cycle the agent reregisters itself at the (Agent is waiting for approval).

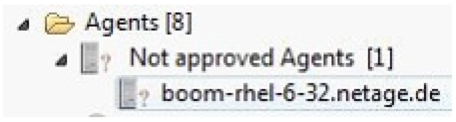
The agent has to be stopped manually (before the "Delete Agent" function is executed) to avoid the re-registration.

Please see the chapter Agent Uninstallation to uninstall an agent from a node.

Re-Approve Agents:

As mentioned above the agent will register itself at the *boom* server (Agent is waiting for approval) after the next heartbeat cycle if the agent has been deleted via the "Delete Agent" operation. Since the "Delete Agent" operation is a functionality of the user interface the agent itself doesn't know anything about it.

To re-approve an agent right click on the "not approved agent" and select "Approve Agent"



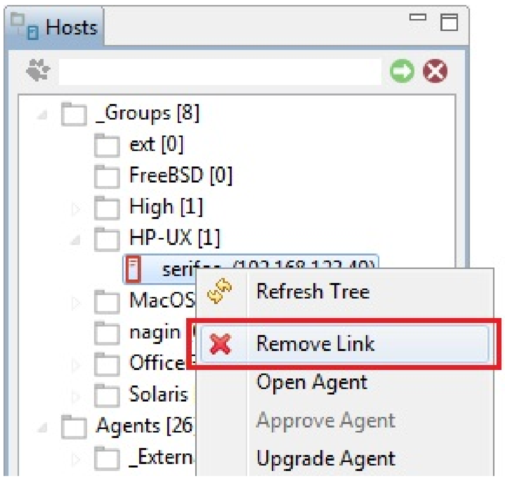
- Agent is again visible in the UI and sorted to the OS specific host group
- Previous links to other host groups are not restored
- Deployed policies on the agent are recognized as "Single deployed" and "unexpected by server" . The server has no more information about the deployed policies for this agent. This information is part of the agent card which has been deleted with the "Delete Agent" operation.
- Deployed packages on the agent (/opt/boom/agent/spi) are not recognized. This information is part of the

agent card which has been deleted with the "Delete Agent" operation.

- The server assigns a new ID to the agent.
- Please follow the steps below to give the server all policy and assignment related information about the corresponding agent:
    - Select the according assignment groups or single policies and packages, right click and select "Mark as Deployed on..."

      or
    - Undeploy all policies and packages from the agent
    - Deploy all assignment groups or policies and packages on the agent

Remove Link:

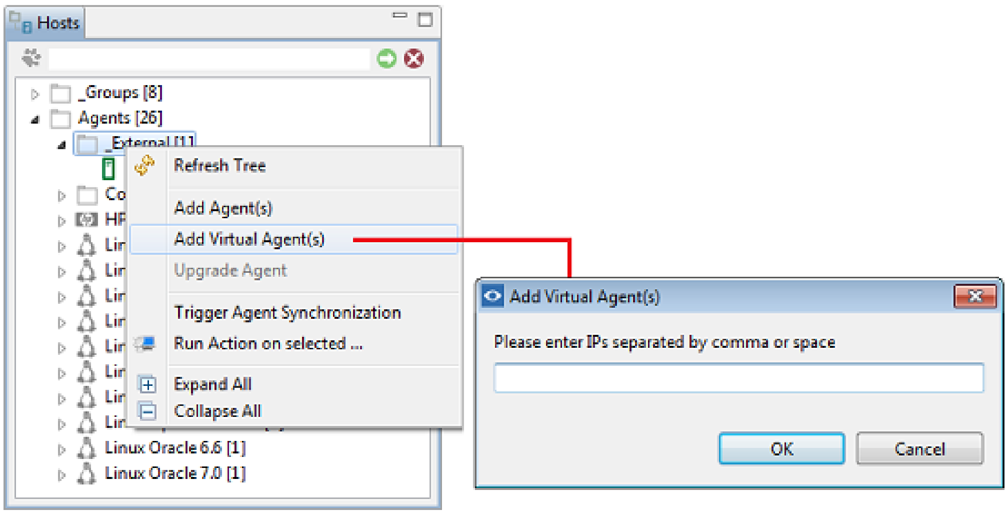


- Link is deleted
- Host is removed from the host group ##No other impact

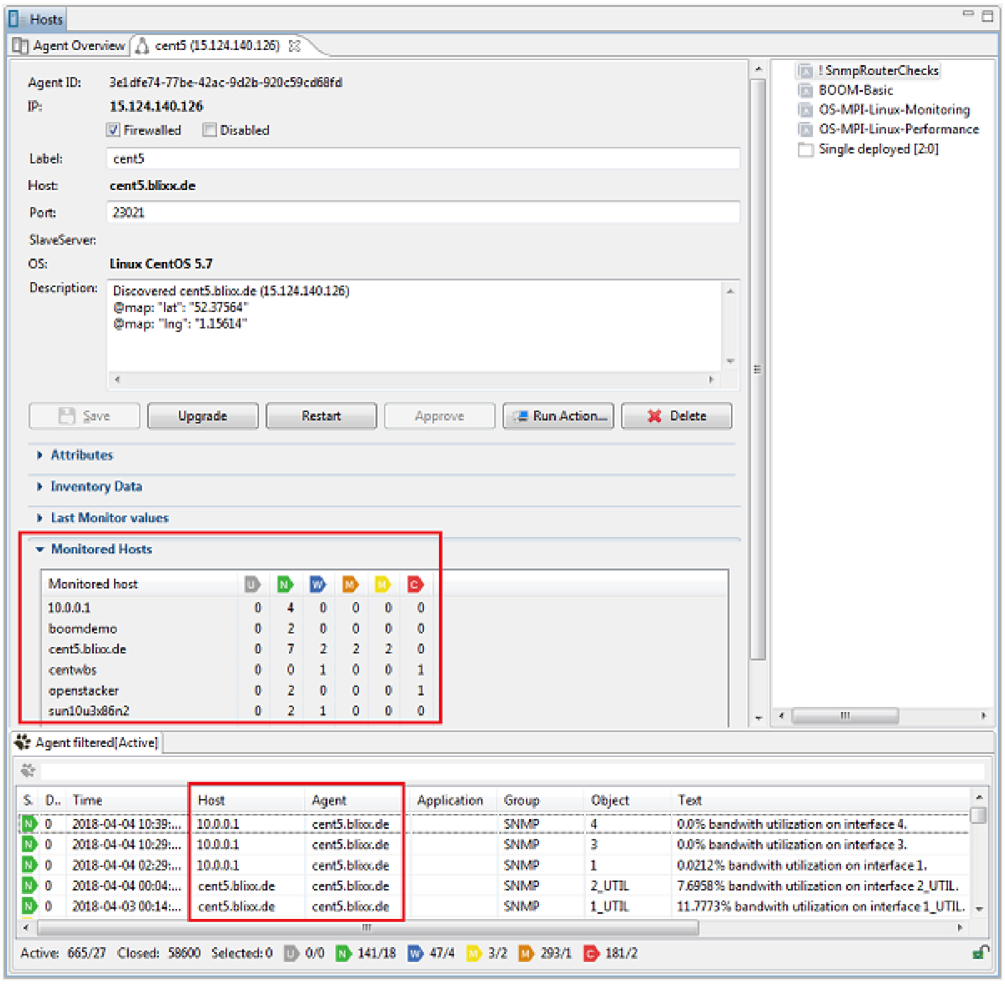
5.11.3. Virtual Agents

An virtual agent is basically a placeholder for a system or device that does not have a *boom* agent installed. Virtual agents are typically used to be able to visualize critical components or systems that are monitored remotely. Any indication with a host attribute matching the virtual agent name will be linked to this agent. These agents are available in the host tree and can be linked to host groups just like a real agent allowing to use them seamless in the definition of e.g. server filters for rights and notification management.

Add Virtual Agent(s):



To add virtual agents, just enter the names or IP addresses exactly as they are received in the host field of the indications. This allows that the indications are assigned to the virtual agents. If indications for a single virtual host are received with multiple different values in the indications host field, e.g. with mixed short or long host name and IP address, according name mappings can be defined in the *boom* servers Configuration View.



Agent ID:	3e1dfe74-77be-42ac-9d2b-920c59cd68fd
IP:	15.124.140.126
	[✓ Firewalled] Disabled
Label:	cent5
Host:	cent5.blixx.de
Port:	23021
SlaveServer:	
OS:	Linux CentOS 5.7
Description:	Discovered cent5.blixx.de (15.124.140.126) @map: "lat": "52.37564" @map: "lng": "1.15614"

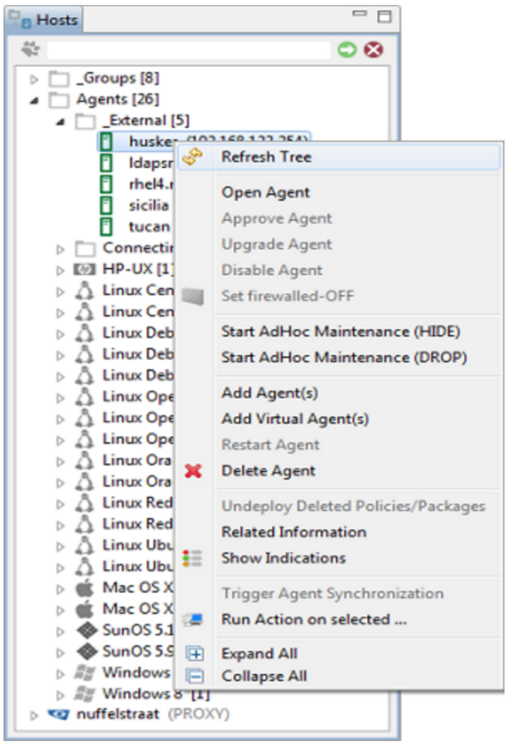
Monitored host	U	N	W	M	M	C
10.0.0.1	0	4	0	0	0	0
boomdemo	0	2	0	0	0	0
cent5.blixx.de	0	7	2	2	2	0
centwbs	0	0	1	0	0	1
openstacker	0	2	0	0	0	1
sun10u3x86n2	0	2	1	0	0	0

S.	D..	Time	Host	Agent	Application	Group	Object	Text
N	0	2018-04-04 10:39:...	10.0.0.1	cent5.blixx.de		SNMP	4	0.0% bandwith utilization on interface 4.
N	0	2018-04-04 10:29:...	10.0.0.1	cent5.blixx.de		SNMP	3	0.0% bandwith utilization on interface 3.
N	0	2018-04-04 02:29:...	10.0.0.1	cent5.blixx.de		SNMP	1	0.0212% bandwith utilization on interface 1.
N	0	2018-04-04 00:04:...	cent5.blixx.de	cent5.blixx.de		SNMP	2_UTIL	7.6958% bandwith utilization on interface 2_UTIL.
N	0	2018-04-03 00:14:...	cent5.blixx.de	cent5.blixx.de		SNMP	1_UTIL	11.7773% bandwith utilization on interface 1_UTIL.

Active: 665/27 Closed: 58600 Selected: 0 U 0/0 N 141/18 W 47/4 M 3/2 M 293/1 C 181/2

The list of incoming host names are visible in the indication browser and in the host details view of the agent systems that perform the remote monitoring.

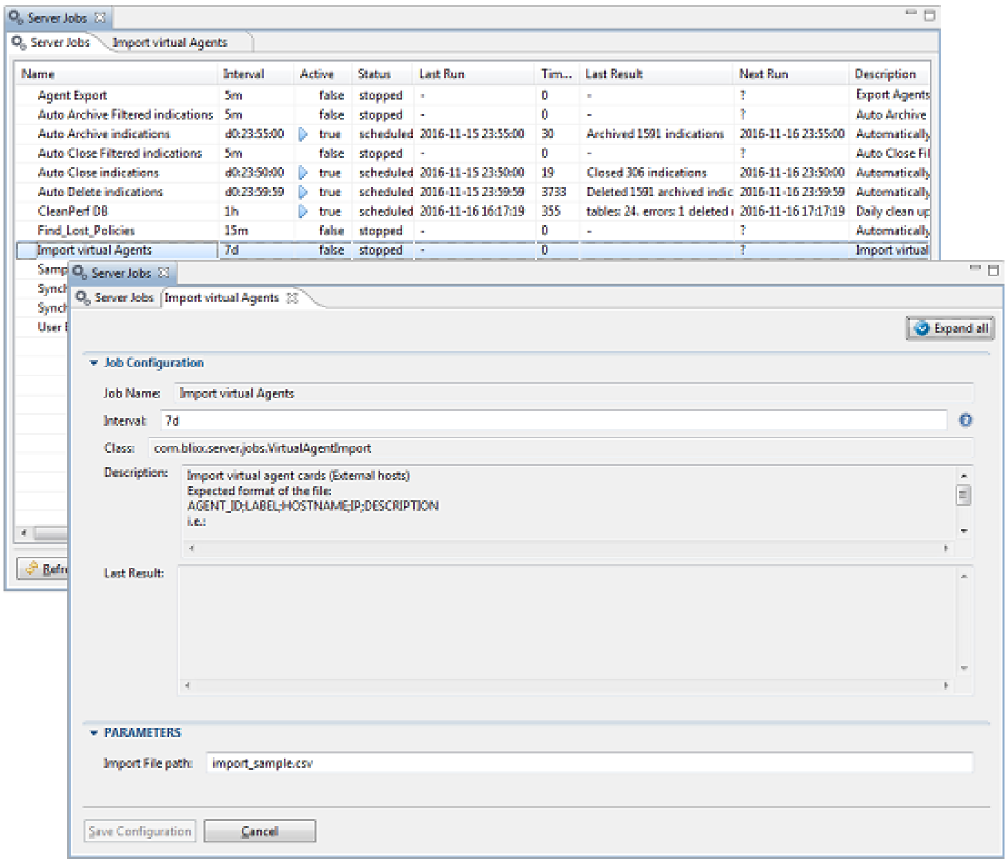
Virtual agents are added automatically in the host group _External since there is no operating system detection.



The virtual agents support the use of common functionality like the shortcuts in the pop up for opening a filtered indication view, handling of maintenance and similar.

Import Virtual Agent(s):

Virtual Agents can be imported and updated by the server job "Import virtual Agents". The job takes the interval and filename as parameters. It expects the file in the directory <boom_server>/srv/jobs/vagentsimport.



Name	Interval	Active	Status	Last Run	Tim...	Last Result	Next Run	Description
Agent Export	5m	false	stopped	-	0	-	?	Export Agents
Auto Archive Filtered indications	5m	false	stopped	-	0	-	?	Auto Archive
Auto Archive indications	d0:23:55:00	true	scheduled	2016-11-15 23:55:00	30	Archived 1591 indications	2016-11-16 23:55:00	Automatically
Auto Close Filtered indications	5m	false	stopped	-	0	-	?	Auto Close Fil
Auto Close indications	d0:23:50:00	true	scheduled	2016-11-15 23:50:00	19	Closed 306 indications	2016-11-16 23:50:00	Automatically
Auto Delete indications	d0:23:59:59	true	scheduled	2016-11-15 23:59:59	3733	Deleted 1591 archived indic	2016-11-16 23:59:59	Automatically
CleanPerf DB	1h	true	scheduled	2016-11-16 16:17:19	355	tables: 24. errors: 1 deleted	2016-11-16 17:17:19	Daily clean up
Find_Lost_Policies	15m	false	stopped	-	0	-	?	Automatically
Import virtual Agents	7d	false	stopped	-	0		?	Import virtual
Samp								
Synch								
Synch								
User								

Job Name: Import virtual Agents

Interval: 7d

Class: com.blixx.server.jobs.VirtualAgentImport

Description:
```
Import virtual agent cards (External hosts)
Expected format of the file:
AGENT_ID;LABEL;HOSTNAME;IP;DESCRIPTION
i.e.:
```

Last Result:

Import File path: import_sample.csv

The file has to be in the format of one line per virtual agent with the fields separated by the ";" character

```
<AGENT_ID>;<LABEL>;<HOSTNAME>;<IP>;<DESCRIPTION>
```

AGENT_ID:

the ID that the *boom* server assigned to this virtual agent (for updates of existing virtual agents) or blank

LABEL:

the label that should be displayed for this virtual agent or blank

HOSTNAME:

the host name that should be used

IP:

the IP address of the virtual agent

DESCRIPTION:

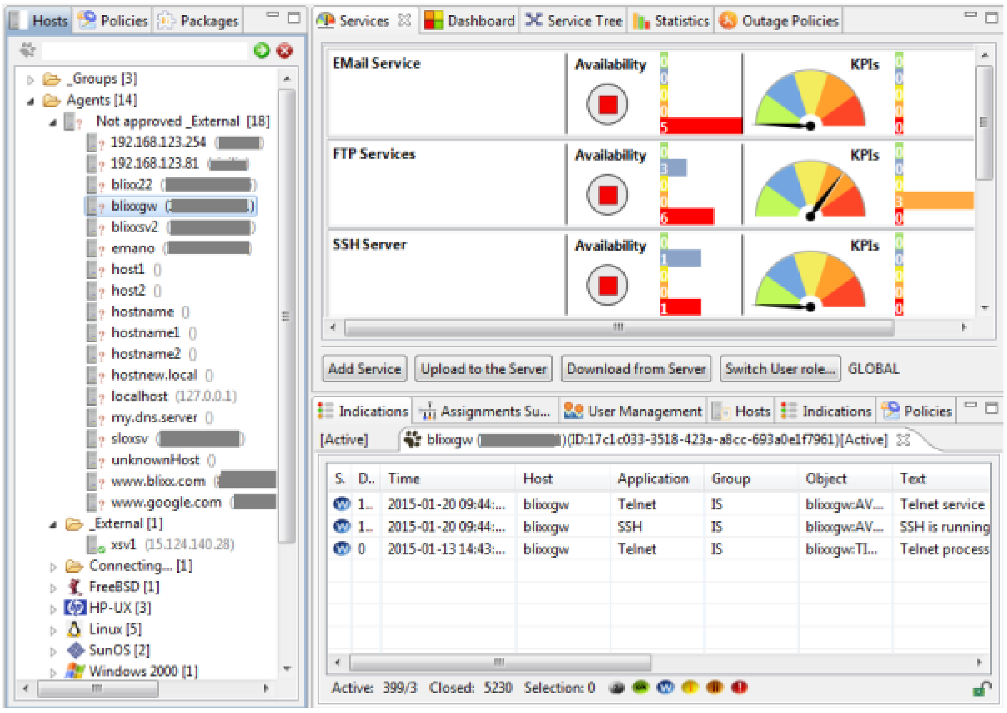
optional description text

Samples:

```
;;NewHostName;1.1.1.1;
a9f14a8a-770c-483c-9a80-a719fa096feb;MyLabel;UpdatedHostName;1.1.1.1;"Updated Description"
```

Automatically Detect Virtual Agent(s):

The server can be configured to automatically create new Virtual Agent entries for hosts that are not known so far. Refer to the chapter *boom* Server Configuration Files for details regarding setting the configuration flag for the discovery. The server will scan the host field of incoming indications and check if these are already known. If it detects an host name that can not be resolved to an existing agent or virtual agent, a new virtual agent will be created in the **Not approved _External** host group.



_Groups [3]
Agents [14]
  Not approved _External [18]
    192.168.123.254 ([redacted])
    192.168.123.81 ([redacted])
    blixx22 ([redacted])
    blixxgw ([redacted])
    blixxsv2 ([redacted])
    emano ([redacted])
    host1 ()
    host2 ()
    hostname ()
    hostname1 ()
    hostname2 ()
    hostnew.local ()
    localhost (127.0.0.1)
    my.dns.server ()
    sloxsv ([redacted])
    unknownHost ()
    www.blixx.com ([redacted]
    www.google.com ([redacted]
  _External [1]
    xsv1 (15.124.140.28)
  Connecting... [1]
  FreeBSD [1]
  HP-UX [3]
  Linux [5]
  SunOS [2]
  Windows 2000 [1]

[Active] blixxgw ([redacted])(ID:17c1c033-3518-423a-a8cc-693a0e1f7961)[Active]

S.	D..	Time	Host	Application	Group	Object	Text
W	1..	2015-01-20 09:44:...	blixxgw	Telnet	IS	blixxgw:AV...	Telnet service
W	1..	2015-01-20 09:44:...	blixxgw	SSH	IS	blixxgw:AV...	SSH is running
W	0	2015-01-13 14:43:...	blixxgw	Telnet	IS	blixxgw:TI...	Telnet process

Active: 399/3 Closed: 5230 Selection: 0

Approving such a virtual host will create this entry permanently and move it to the **_External** host group.

If you are working with multiple *boom* servers in a master/slave concept, the discovery of virtual agents should be turned on either on the Master or on the Slave servers, but not on both sides. If the discovery would be turned on on both sides, conflicting virtual agent entries will be created which need to be manually sorted out.

5.12. Auditing

The *boom* server uses a file based auditing. Auditing tracks the user interaction with boom.

Auditing records the following user actions to the BOOMAudit log file:

Agent Card	add, delete, approve, restart, upgrade, enable/disable, set Firewalled ON/OFF, link/unlink from HostGroups, modify Agent attributes
Host Groups	add, rename, delete, change
Policies	add, modify, delete
Policy tree update	create, rename, delete elements
Binary Packages	create, upload, rename, delete files create, delete packages
Assignment tree changes	create, rename, delete elements
Deployments	deploy, redeploy, undeploy, queue deployments, trigger, cancel
Maintenance, Modify Server Policies, Attribute Filters	add, modify, delete, enable, disable
Server/Agent: Synchronizing the configuration	trigger synchronization
Users	add, modify, delete users or user roles
Actions	create, rename, modify, delete actions and action groups

5.13. Indication Enrichment/Modification

Indication Enrichment/Modification is designed to change attributes of incoming indications on the *boom* server to derive more accurate and enriched information for the *boom* operators.
The indications enrichment/modification occurs on the *boom* Server before the indication correlation and de-duplication takes place. Server Policies are designed to change the incoming indication attributes.

More detailed information see in chapter Server Policies

5.14. Maintenance Window Management

Maintenance Window Management (formerly named Maintenance Management) means the handling of planned and unplanned non-availability for dedicated systems or applications for a definable time interval. *boom* Maintenance Windows are used to define regular or ad-hoc times during which all or parts of the monitoring is paused e.g. to implement maintenance windows for systems or applications. A similar effect can be reached by using the functionalities to disable policies or an agent, but by defining Maintenance Windows more flexibility and control is possible.

In general there are two types of Maintenances: Scheduled (Server based) and AdHoc (Agent initiated) Maintenance.

The scheduled Maintenance Windows are defined and triggered on the server side, while AdHoc maintenances are started and in most cases stopped by the agent. A typical use case for scheduled maintenances are regular intervals e.g. two hours every last friday each month to install operating system and application patches. A typical use case for an AdHoc maintenance are backups, where the backup startup script tells the agent to put all performance related alerts to Maintenance for approximately 2 hours and the post-backup script tells the agent to finish the maintenance.

The processing of an maintenance always happens on the server side. Incoming indications (active and "insert as closed") will be first passing the Server Modify Policies before the server decides if the indication matches an active Maintenance configuration and puts them to the outage stream or passes it on to the deduplication. AdHoc Maintenances take precedence over scheduled Maintenances.

AdHoc Maintenance:

Any agent has the possibility to trigger an Maintenance for itself. This Maintenance has an expected duration and can have a fixed time period (see "–du" options). A typical example is a "backup time" e.g. when the agent must inform the server that for some hours all indications must be suppressed or ignored. After finish of the backup e.g. the agent must inform the server to disable the Maintenance. Multiple overlapping Maintenances are supported. AdHoc based Maintenances will be processed before scheduled Maintenances, impacting to which Maintenance the incoming indications will be related. AdHoc Maintenances can be activated by scripty or users via the **boom_agent_cli.jar**.

The following Maintenance Modes/Actions are implemented:

- DROP: Indications will be discarded
- HIDE: Indications will come to a separate "Maintenance Browser"
  - Post Action NONE
    the indications will stay in the Maintenance browser after the Maintenance is ended
  - Post Action DROP
    the indications will be dropped from the Maintenance browser at the end of the maintenance
  - Post Action PUBLISH
    the indications will be moved to the active browser at the end of the maintenance

Scheduled Maintenace:

The following Maintenance Modes/Actions are implemented:

- DROP: Indications will be discarded
- HIDE: Indications will come to a separate "Maintenance Browser"
  - Post Action NONE
    the indications will stay in the Maintenance browser after the Maintenance is ended
  - Post Action DROP
    the indications will be dropped from the Maintenance browser at the end of the maintenance
  - Post Action PUBLISH the indications will be moved to the active browser at the end of the maintenance

Maintenance Processing:

- Will be performed after indication Enrichment/Modification
- Will be performed before indication De-duplication
- Affects only new active/closed indications
- Doesn’t affect indication updates (Duplicates)
- Automatic actions are not performed during Maintenance time. A corresponding annotation is added ("Remote Action can’t be processed during Maintenance") to the indication.
- Forward filters with Exec-Target will be not performed for indications that are arriving in a Maintenance time.

Maintenance in a Master/Slave environment:

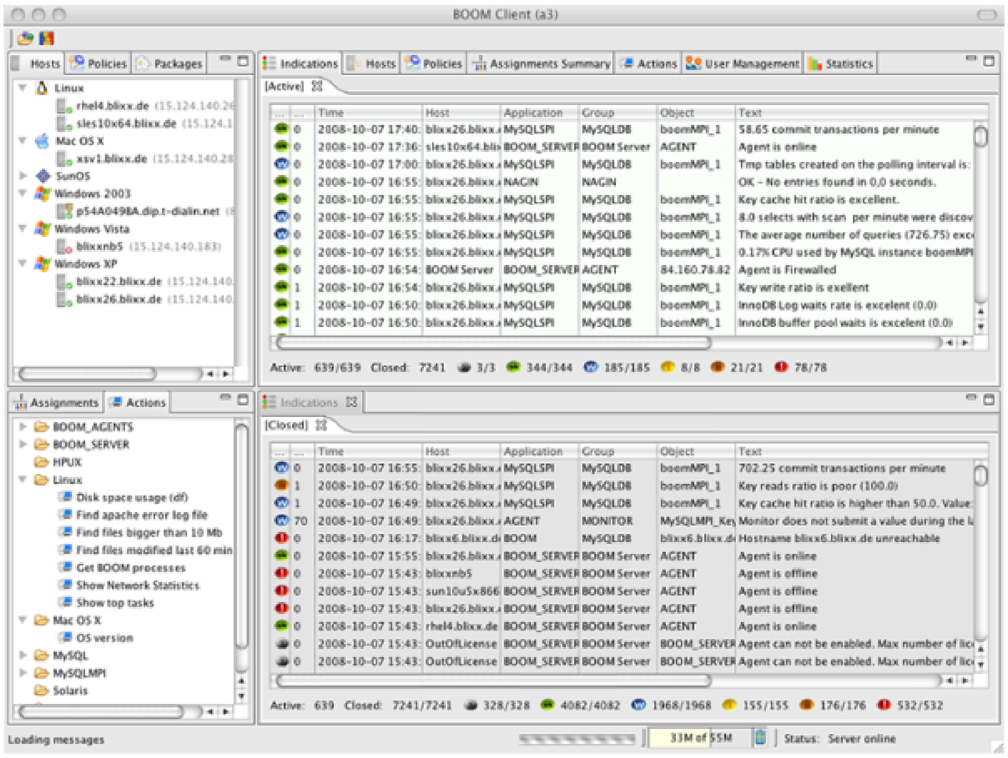
- Scheduled Maintenances can be configured for the **Master** and/or each Slave individually.
- Maintenance configuration is only available for the *boom* Server Mode1 (Proxy = Slave).
- Please note that for Primary-Backup (mode 8) maintenance windows are replicated from Primary to Backup server.
- Maintenance configuration is NOT available for Server Mode2 (Forward & Control) and Mode3 (Mirror).
- **Recommendation:** Each Scheduled Maintenance should be setup on the FIRST *boom* server receiving the corresponding indications (Slave).
    - Automatic Actions can only be suppressed on the FIRST *boom* server (Slave)
    - Due to Duplicates are created on the FIRST *boom* server:
    - If a Maintenance is activated on this FIRST server an indication update (Duplicate) is prevented by the Maintenance because the Maintenance is processed before De-duplication.
    - If a Maintenance isn't activated on this FIRST server but on the Master server then an indication update (Duplicate) isn't prevented by the Maintenance and therefore the indication is still visible and the Duplicate counter is increased.

Chapter 6. *boom* User Interface (Client)

boom GUI Workbench

When you login for the first time to the business open operations manager, you get the default boomworkbench. Depending on your operating system and your system settings the style of the *boom* workbench is slightly different. Here are some examples:

Apple



[Active]

...	...	Time	Host	Application	Group	Object	Text
●	0	2008-10-07 17:40:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	58.65 commit transactions per minute
●	0	2008-10-07 17:36:	sles10x64.blix	BOOM_SERVER	BOOM Server	AGENT	Agent is online
●	0	2008-10-07 17:00:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	Tmp tables created on the polling interval is:
●	0	2008-10-07 16:55:	blixx26.blixx.d	NAGIN	NAGIN		OK - No entries found in 0,0 seconds.
●	0	2008-10-07 16:55:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	Key cache hit ratio is excellent.
●	0	2008-10-07 16:55:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	8.0 selects with scan per minute were discov
●	0	2008-10-07 16:55:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	The average number of queries (726.75) exc
●	0	2008-10-07 16:55:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	0.17% CPU used by MySQL instance boomMPI
●	0	2008-10-07 16:54:	BOOM Server	BOOM_SERVER	AGENT	84.160.78.82	Agent is Firewalled
●	1	2008-10-07 16:54:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	Key write ratio is exellent
●	1	2008-10-07 16:50:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	InnoDB Log waits rate is excelent (0.0)
●	1	2008-10-07 16:50:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	InnoDB buffer pool waits is excelent (0.0)

Active: 639/639 Closed: 7241 ● 3/3 ● 344/344 ● 185/185 ● 8/8 ● 21/21 ● 78/78

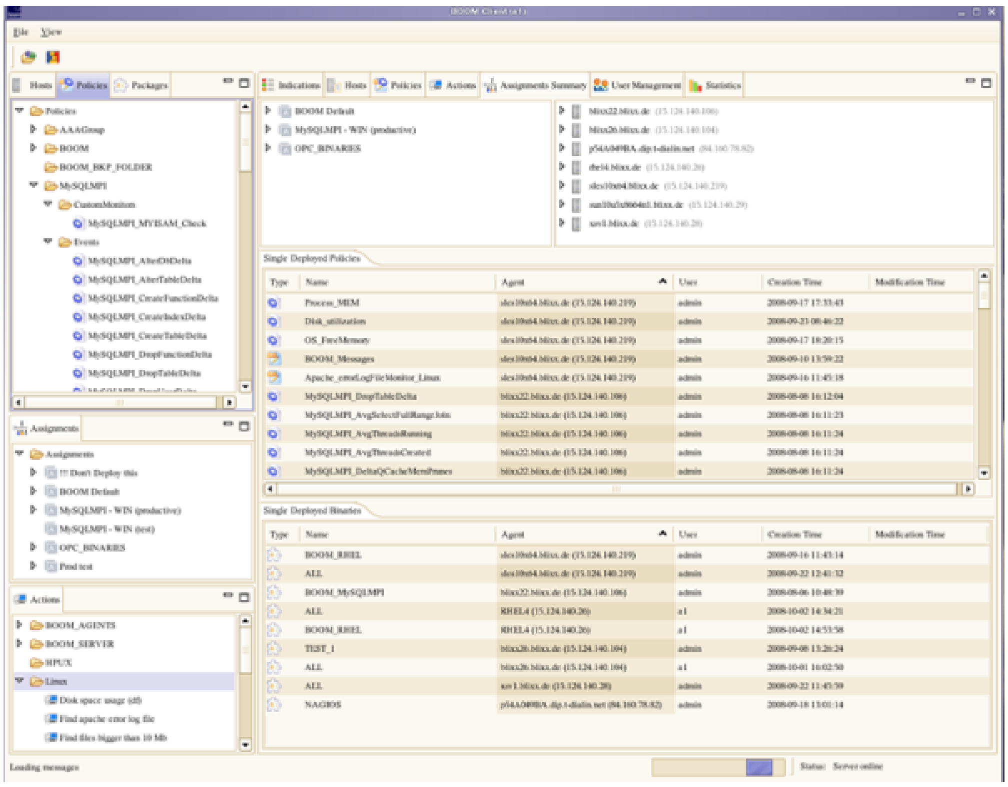
[Closed]

...	...	Time	Host	Application	Group	Object	Text
●	0	2008-10-07 16:55:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	702.25 commit transactions per minute
●	1	2008-10-07 16:50:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	Key reads ratio is poor (100.0)
●	1	2008-10-07 16:49:	blixx26.blixx.d	MySQLSPI	MySQLDB	boomMPI_1	Key cache hit ratio is higher than 50.0. Value:
●	70	2008-10-07 16:49:	blixx26.blixx.d	AGENT	MONITOR	MySQLMPI_Key	Monitor does not submit a value during the l
●	0	2008-10-07 16:17:	blixx6.blixx.de	BOOM	MySQLDB	blixx6.blixx.de	Hostname blixx6.blixx.de unreachable
●	0	2008-10-07 15:55:	blixx26.blixx.d	BOOM_SERVER	BOOM Server	AGENT	Agent is online
●	0	2008-10-07 15:43:	blixxnb5	BOOM_SERVER	BOOM Server	AGENT	Agent is offline
●	0	2008-10-07 15:43:	sun10u5x866	BOOM_SERVER	BOOM Server	AGENT	Agent is offline
●	0	2008-10-07 15:43:	blixx26.blixx.d	BOOM_SERVER	BOOM Server	AGENT	Agent is offline
●	0	2008-10-07 15:43:	rhel4.blixx.de	BOOM_SERVER	BOOM Server	AGENT	Agent is online
●	0	2008-10-07 15:43:	OutOfLicense	BOOM_SERVER	BOOM Server	BOOM_SERVER	Agent can not be enabled. Max number of lic
●	0	2008-10-07 15:43:	OutOfLicense	BOOM_SERVER	BOOM Server	BOOM_SERVER	Agent can not be enabled. Max number of lic

Active: 639 Closed: 7241/7241 ● 328/328 ● 4082/4082 ● 1968/1968 ● 155/155 ● 176/176 ● 532/532

33M of 55M Status: Server online

Linux


BOOM Client (a1)
File View
Hosts
Policies
Packages
Indications
Actions
Assignments Summary
User Management
Statistics
Policies
AAAGroup
BOOM
BOOM_BKP_FOLDER
MySQLMPI
CustomMonitors
MySQLMPI_MYISAM_Check
Events
MySQLMPI_AlterDbDelta
MySQLMPI_AlterTableDelta
MySQLMPI_CreateFunctionDelta
MySQLMPI_CreateIndexDelta
MySQLMPI_CreateTableDelta
MySQLMPI_DropFunctionDelta
MySQLMPI_DropTableDelta
BOOM Default
MySQLMPI - WIN (productive)
OPC_BINARIES
blixx22.blixx.de (15.124.140.106)
blixx26.blixx.de (15.124.140.104)
p54A049BA.dip.t-dialin.net (84.160.78.82)
rhel4.blixx.de (15.124.140.26)
sles10x64.blixx.de (15.124.140.219)
sun10x5x8664nl.blixx.de (15.124.140.29)
xsv1.blixx.de (15.124.140.28)
Assignments
!!! Don't Deploy this
MySQLMPI - WIN (test)
Prod test
BOOM_AGENTS
BOOM_SERVER
HPUX
Linux
Disk space usage (df)
Find apache error log file
Find files bigger than 10 Mb
Loading messages

Single Deployed Policies

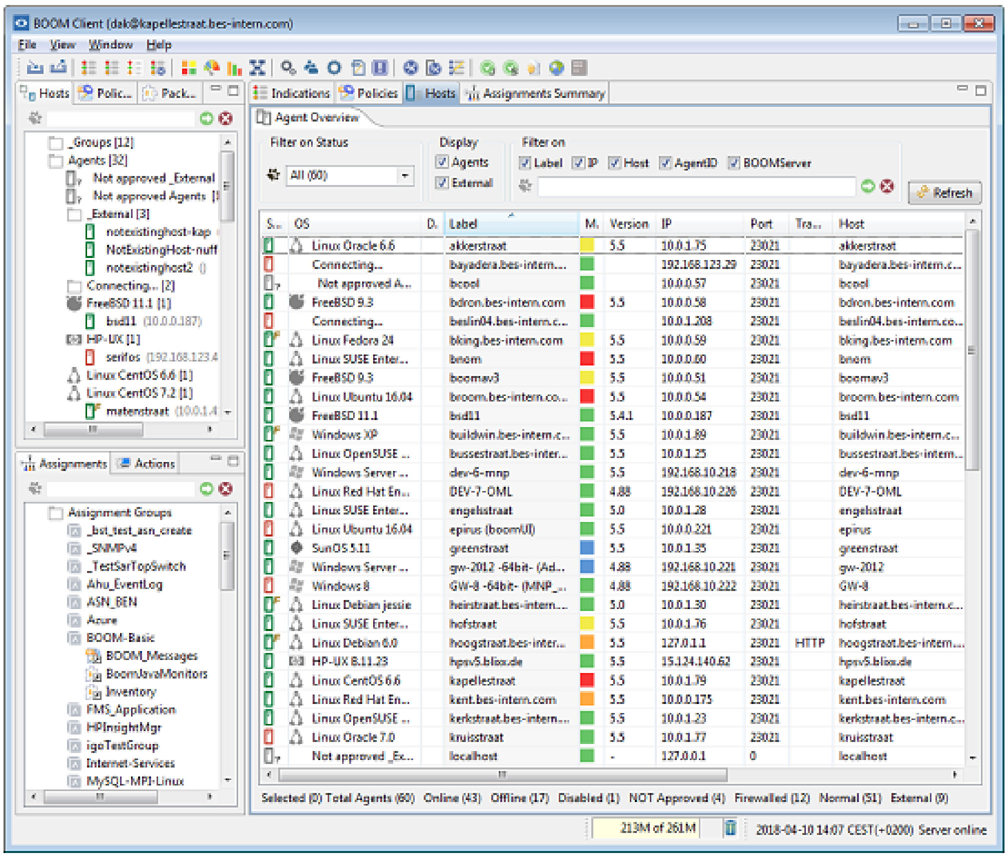
Type	Name	Agent	User	Creation Time	Modification Time
●	Process_MEM	sles10x64.blixx.de (15.124.140.219)	admin	2008-09-17 17:33:43	
●	Disk_utilization	sles10x64.blixx.de (15.124.140.219)	admin	2008-09-23 08:46:22	
●	OS_FreeMemory	sles10x64.blixx.de (15.124.140.219)	admin	2008-09-17 18:20:15	
●	BOOM_Messages	sles10x64.blixx.de (15.124.140.219)	admin	2008-09-10 13:59:22	
●	Apache_errorLogFileMonitor_Linux	sles10x64.blixx.de (15.124.140.219)	admin	2008-09-16 11:45:18	
●	MySQLMPI_DropTableDelta	blixx22.blixx.de (15.124.140.106)	admin	2008-08-08 16:12:04	
●	MySQLMPI_AvgSelectFullRangeJoin	blixx22.blixx.de (15.124.140.106)	admin	2008-08-08 16:11:23	
●	MySQLMPI_AvgThreadsRunning	blixx22.blixx.de (15.124.140.106)	admin	2008-08-08 16:11:24	
●	MySQLMPI_AvgThreadsCreated	blixx22.blixx.de (15.124.140.106)	admin	2008-08-08 16:11:24	
●	MySQLMPI_DeltaQCacheMemPrunes	blixx22.blixx.de (15.124.140.106)	admin	2008-08-08 16:11:24	

Single Deployed Binaries

Type	Name	Agent	User	Creation Time	Modification Time
●	BOOM_RHEL	sles10x64.blixx.de (15.124.140.219)	admin	2008-09-16 11:43:14	
●	ALL	sles10x64.blixx.de (15.124.140.219)	admin	2008-09-22 12:41:32	
●	BOOM_MySQLMPI	blixx22.blixx.de (15.124.140.106)	admin	2008-08-06 10:48:39	
●	ALL	RHEL4 (15.124.140.26)	a1	2008-10-02 14:34:21	
●	BOOM_RHEL	RHEL4 (15.124.140.26)	a1	2008-10-02 14:53:58	
●	TEST_1	blixx26.blixx.de (15.124.140.104)	admin	2008-09-08 13:26:24	
●	ALL	blixx26.blixx.de (15.124.140.104)	a1	2008-10-01 16:02:50	
●	ALL	xsv1.blixx.de (15.124.140.28)	admin	2008-09-22 11:45:59	
●	NAGIOS	p54A049BA.dip.t-dialin.net (84.160.78.82)	admin	2008-09-18 13:01:14	

Status: Server online

Windows



S..	OS	D..	Label	M..	Version	IP	Port	Tra...	Host
	Linux Oracle 6.6		akkerstraat	🟨	5.5	10.0.1.75	23021		akkerstraat
	Connecting...		bayadera.bes-intern....	🟩		192.168.123.29	23021		bayadera.bes-intern.c...
	Not approved A...		bcool	🟩		10.0.0.57	23021		bcool
	FreeBSD 9.3		bdron.bes-intern.com	🟥	5.5	10.0.0.58	23021		bdron.bes-intern.com
	Connecting...		beslin04.bes-intern.c...	🟩		10.0.1.208	23021		beslin04.bes-intern.co...
	Linux Fedora 24		bking.bes-intern.com	🟨	5.5	10.0.0.59	23021		bking.bes-intern.com
	Linux SUSE Enter...		bnom	🟥	5.5	10.0.0.60	23021		bnom
	FreeBSD 9.3		boomav3	🟨	5.5	10.0.0.51	23021		boomav3
	Linux Ubuntu 16.04		broom.bes-intern.co...	🟥	5.5	10.0.0.54	23021		broom.bes-intern.com
	FreeBSD 11.1		bsd11	🟩	5.4.1	10.0.0.187	23021		bsd11
	Windows XP		buildwin.bes-intern.c...	🟩	5.5	10.0.1.89	23021		buildwin.bes-intern.c...
	Linux OpenSUSE ...		bussestraat.bes-inter...	🟩	5.5	10.0.1.25	23021		bussestraat.bes-intern...
	Windows Server ...		dev-6-mnp	🟩	5.5	192.168.10.218	23021		dev-6-mnp
	Linux Red Hat En...		DEV-7-OML	🟩	4.88	192.168.10.226	23021		DEV-7-OML
	Linux SUSE Enter...		engelstraat	🟩	5.0	10.0.1.28	23021		engelstraat
	Linux Ubuntu 16.04		epirus (boomUI)	🟩	5.5	10.0.0.221	23021		epirus
	SunOS 5.11		greenstraat	🟦	5.5	10.0.1.35	23021		greenstraat
	Windows Server ...		gw-2012 -64bit- (Ad...	🟦	4.88	192.168.10.221	23021		gw-2012
	Windows 8		GW-8 -64bit- (MNP_...	🟩	4.88	192.168.10.222	23021		GW-8
	Linux Debian jessie		heirstraat.bes-intern....	🟩	5.0	10.0.1.30	23021		heirstraat.bes-intern.c...
	Linux SUSE Enter...		hofstraat	🟨	5.5	10.0.1.76	23021		hofstraat
	Linux Debian 6.0		hoogstraat.bes-inter...	🟧	5.5	127.0.1.1	23021	HTTP	hoogstraat.bes-intern....
	HP-UX B.11.23		hpsv5.blixx.de	🟩	5.5	15.124.140.62	23021		hpsv5.blixx.de
	Linux CentOS 6.6		kapellestraat	🟥	5.5	10.0.1.79	23021		kapellestraat
	Linux Red Hat En...		kent.bes-intern.com	🟧	5.5	10.0.0.175	23021		kent.bes-intern.com
	Linux OpenSUSE ...		kerkstraat.bes-intern....	🟩	5.5	10.0.1.23	23021		kerkstraat.bes-intern.c...
	Linux Oracle 7.0		kruisstraat	🟩	5.5	10.0.1.77	23021		kruisstraat
	Not approved _Ex...		localhost	🟩	-	127.0.0.1	0		localhost

Selected (0) Total Agents (60) Online (43) Offline (17) Disabled (1) NOT Approved (4) Firewalled (12) Normal (51) External (9)

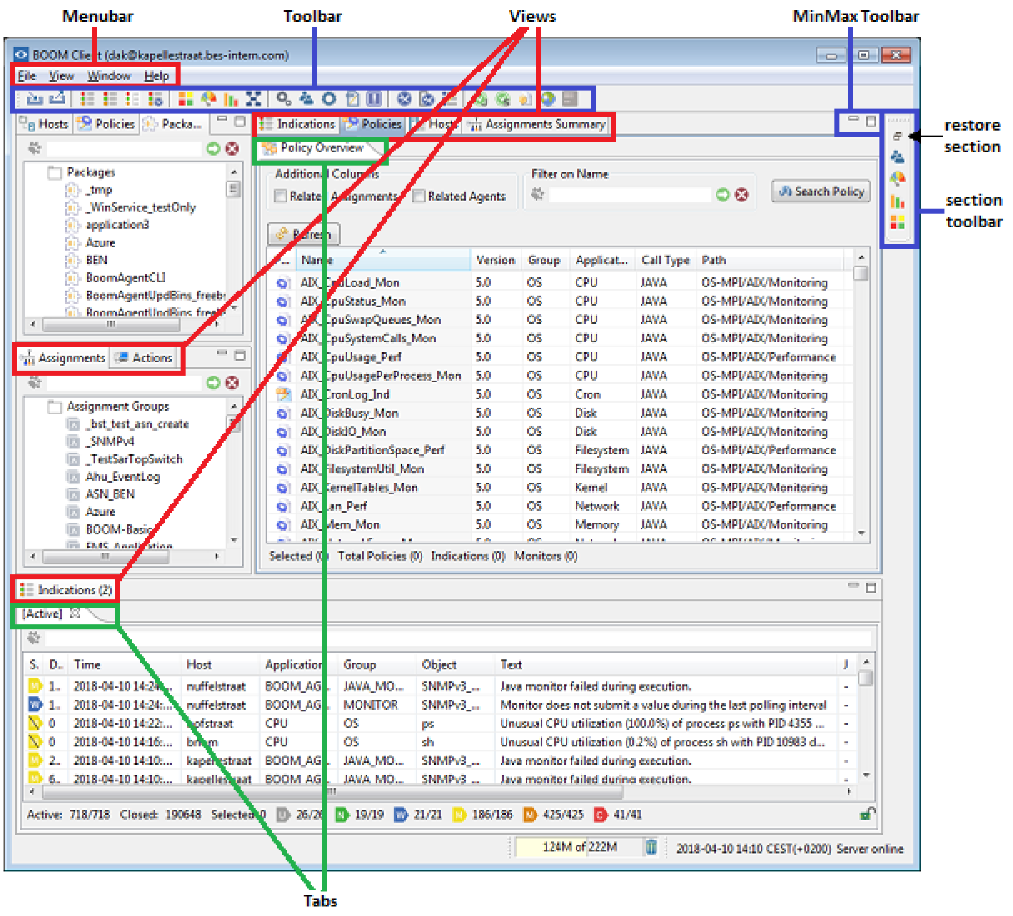
213M of 261M

2018-04-10 14:07 CEST(+0200) Server online

6.1. Overview and Basic Operations

6.1.1. General Information

The default *boom* workbench is split into multiple sections. Each section contains one or more views. The views are displayed depending on the user rights. The workbench itself can be reorganized for your individual need. All changes that are made to the *boom* GUI appearance will be saved inside the user profile.



...	Name	Version	Group	Applicat...	Call Type	Path
	AIX_CpuLoad_Mon	5.0	OS	CPU	JAVA	OS-MPI/AIX/Monitoring
	AIX_CpuStatus_Mon	5.0	OS	CPU	JAVA	OS-MPI/AIX/Monitoring
	AIX_CpuSwapQueues_Mon	5.0	OS	CPU	JAVA	OS-MPI/AIX/Monitoring
	AIX_CpuSystemCalls_Mon	5.0	OS	CPU	JAVA	OS-MPI/AIX/Monitoring
	AIX_CpuUsage_Perf	5.0	OS	CPU	JAVA	OS-MPI/AIX/Performance
	AIX_CpuUsagePerProcess_Mon	5.0	OS	CPU	JAVA	OS-MPI/AIX/Monitoring
	AIX_CronLog_Ind	5.0	OS	Cron	JAVA	OS-MPI/AIX/Monitoring
	AIX_DiskBusy_Mon	5.0	OS	Disk	JAVA	OS-MPI/AIX/Monitoring
	AIX_DiskIO_Mon	5.0	OS	Disk	JAVA	OS-MPI/AIX/Monitoring
	AIX_DiskPartitionSpace_Perf	5.0	OS	Filesystem	JAVA	OS-MPI/AIX/Performance
	AIX_FilesystemUtil_Mon	5.0	OS	Filesystem	JAVA	OS-MPI/AIX/Monitoring
	AIX_KernelTables_Mon	5.0	OS	Kernel	JAVA	OS-MPI/AIX/Monitoring
	AIX_Lan_Perf	5.0	OS	Network	JAVA	OS-MPI/AIX/Performance
	AIX_Mem_Mon	5.0	OS	Memory	JAVA	OS-MPI/AIX/Monitoring

Selected (0) Total Policies (0) Indications (0) Monitors (0)

S.	D..	Time	Host	Application	Group	Object	Text	J
M	1..	2018-04-10 14:24:...	nuffelstraat	BOOM_AG...	JAVA_MO...	SNMPv3_...	Java monitor failed during execution.	-
W	1..	2018-04-10 14:24:...	nuffelstraat	BOOM_AG...	MONITOR	SNMPv3_...	Monitor does not submit a value during the last polling interval	-
	0	2018-04-10 14:22:...	hofstraat	CPU	OS	ps	Unusual CPU utilization (100.0%) of process ps with PID 4355 ...	-
	0	2018-04-10 14:16:...	brum	CPU	OS	sh	Unusual CPU utilization (0.2%) of process sh with PID 10983 d...	-
M	2..	2018-04-10 14:10:...	kapellestraat	BOOM_AG...	JAVA_MO...	SNMPv3_...	Java monitor failed during execution.	-
M	6..	2018-04-10 14:10:...	kapellestraat	BOOM_AG...	JAVA_MO...	SNMPv3_...	Java monitor failed during execution.	-

Active: 718/718 Closed: 190648 Selected: 0 U 26/26 N 19/19 W 21/21 N 186/186 M 425/425 C 41/41

124M of 222M 2018-04-10 14:10 CEST(+0200) Server online

The **GUI workbench** includes the following sections:

- On the top of your screen you can see the following information:
  ***boom* Client (<user>@<boom-server>)**
  where <user> lists the username of the currently logged in user and <boom-server>lists the name of the *boom* server system you are connected to.
- **Menu bar:**
  The menu bar is a horizontal bar anchored to the top of the screen. It contains the following application-specific menus:
  File
  View
  Window
  Help
- **Tool bar:**
  The tool bar contains all optional views belonging to the user. These views can be reloaded to the view bar when the view was removed.
- **Views:**
  Your are able to reorder the view buttons. Select an view and move it to the favoured position.

  There are **two kind of views**:

  Fixed views

  which can't be removed from the workbench:

  1. Hosts
  2. Policies
  3. Packages
  4. Indications

5. Assignments
6. Actions

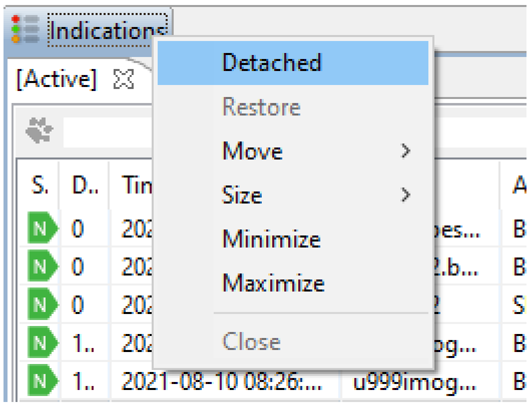
Removeable views
which can be removed from the workbench and reloaded:

1. AdHoc Maintenance
2. Browser
3. Configuration
4. Dashboard
5. Service Dashboard
6. Notifications
7. Scheduled Maintenance
8. Server Filters
9. Server Jobs
10. Server policies
11. Service Tree
12. SSH
13. Statistic
14. User Management
15. Export
16. Import

- **Min/Max Tool bar:**
  With the Minimize button the focused section can be sent to the background.
  With the Maximize button the focused section can be displayed in max size.

6.1.2. Basic Workbench Operations

Context Menu:



Right-click on the View header will open the following context menu:

Detached:
Opens the selected view as detached window. Closing the detached view reattaches the window.

Restore Sections:
The minimized section appear as a small section tool bar at the side of the workbench. The first icon in the section tool bar is always the restore icon to restore the complete section. All other icons represent the views that

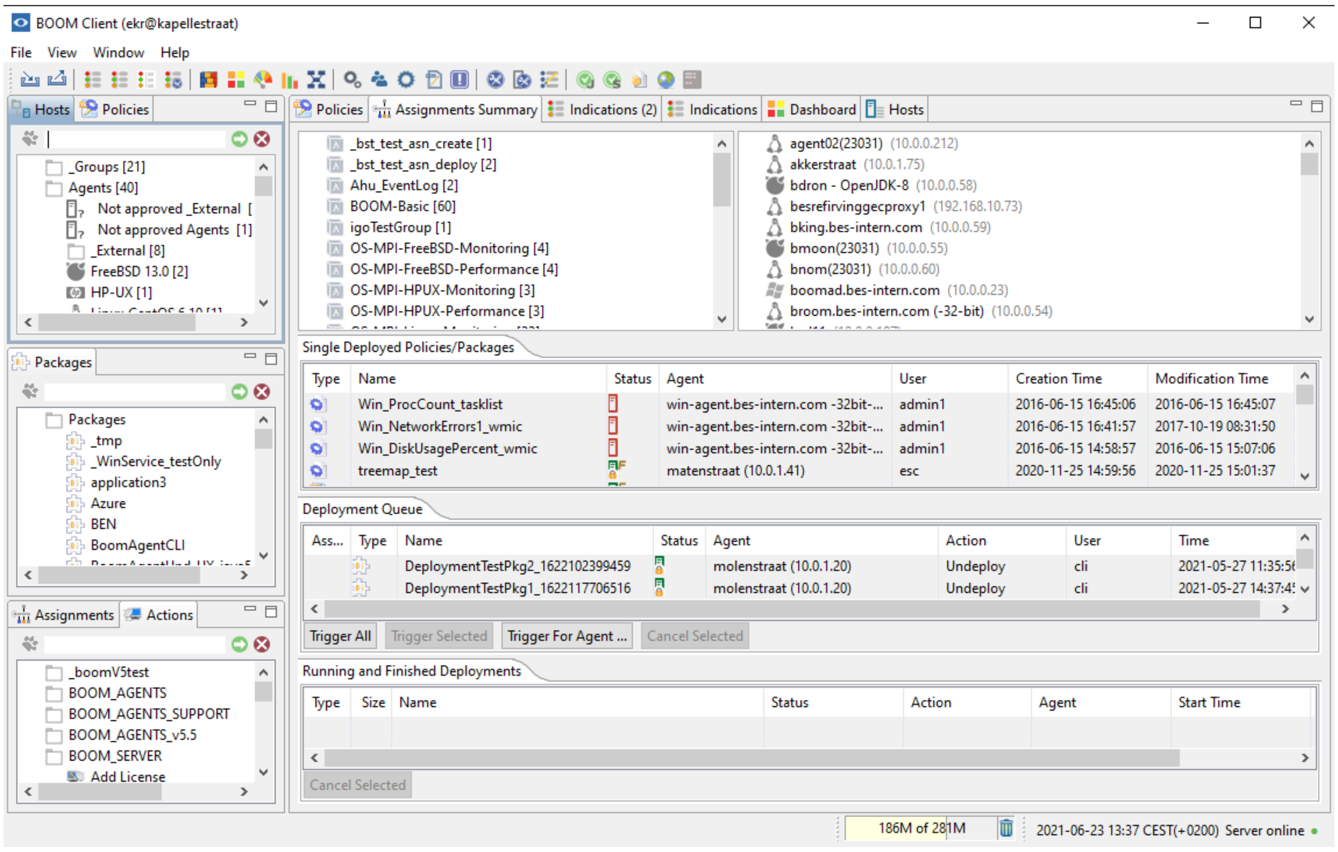
are located inside the minimized section. Clicking on one of this icon will restore a single view. The tool bar itself is movable and can be placed on the left, right, top or bottom of the workbench.

Move Sections and Views:

Sections and views can be moved to any place inside the workbench. Views can be moved either within a section (change the appearance order of the views in a section) or even to a different section. All sections and views can also be moved using drag&drop. Sections and views cannot be moved outside the *boom* workbench!

Minimize/Maximize Section:

To minimize or maximize a section you can also double-click on the section header.



_Groups [21]
Agents [40]
Not approved _External [
Not approved Agents [1]
_External [8]
FreeBSD 13.0 [2]
HP-UX [1]

Packages
_tmp
_WinService_testOnly
application3
Azure
BEN
BoomAgentCLI

_boomV5test
BOOM_AGENTS
BOOM_AGENTS_SUPPORT
BOOM_AGENTS_v5.5
BOOM_SERVER
Add License

_bst_test_asn_create [1]
_bst_test_asn_deploy [2]
Ahu_EventLog [2]
BOOM-Basic [60]
igoTestGroup [1]
OS-MPI-FreeBSD-Monitoring [4]
OS-MPI-FreeBSD-Performance [4]
OS-MPI-HPUX-Monitoring [3]
OS-MPI-HPUX-Performance [3]

agent02(23031) (10.0.0.212)
akkerstraat (10.0.1.75)
bdron - OpenJDK-8 (10.0.0.58)
besrefirvinggecproxy1 (192.168.10.73)
bking.bes-intern.com (10.0.0.59)
bmoon(23031) (10.0.0.55)
bnom(23031) (10.0.0.60)
boomad.bes-intern.com (10.0.0.23)
broom.bes-intern.com (-32-bit) (10.0.0.54)

Single Deployed Policies/Packages

Type	Name	Status	Agent	User	Creation Time	Modification Time
	Win_ProcCount_tasklist		win-agent.bes-intern.com -32bit-...	admin1	2016-06-15 16:45:06	2016-06-15 16:45:07
	Win_NetworkErrors1_wmic		win-agent.bes-intern.com -32bit-...	admin1	2016-06-15 16:41:57	2017-10-19 08:31:50
	Win_DiskUsagePercent_wmic		win-agent.bes-intern.com -32bit-...	admin1	2016-06-15 14:58:57	2016-06-15 15:07:06
	treemap_test		matenstraat (10.0.1.41)	esc	2020-11-25 14:59:56	2020-11-25 15:01:37

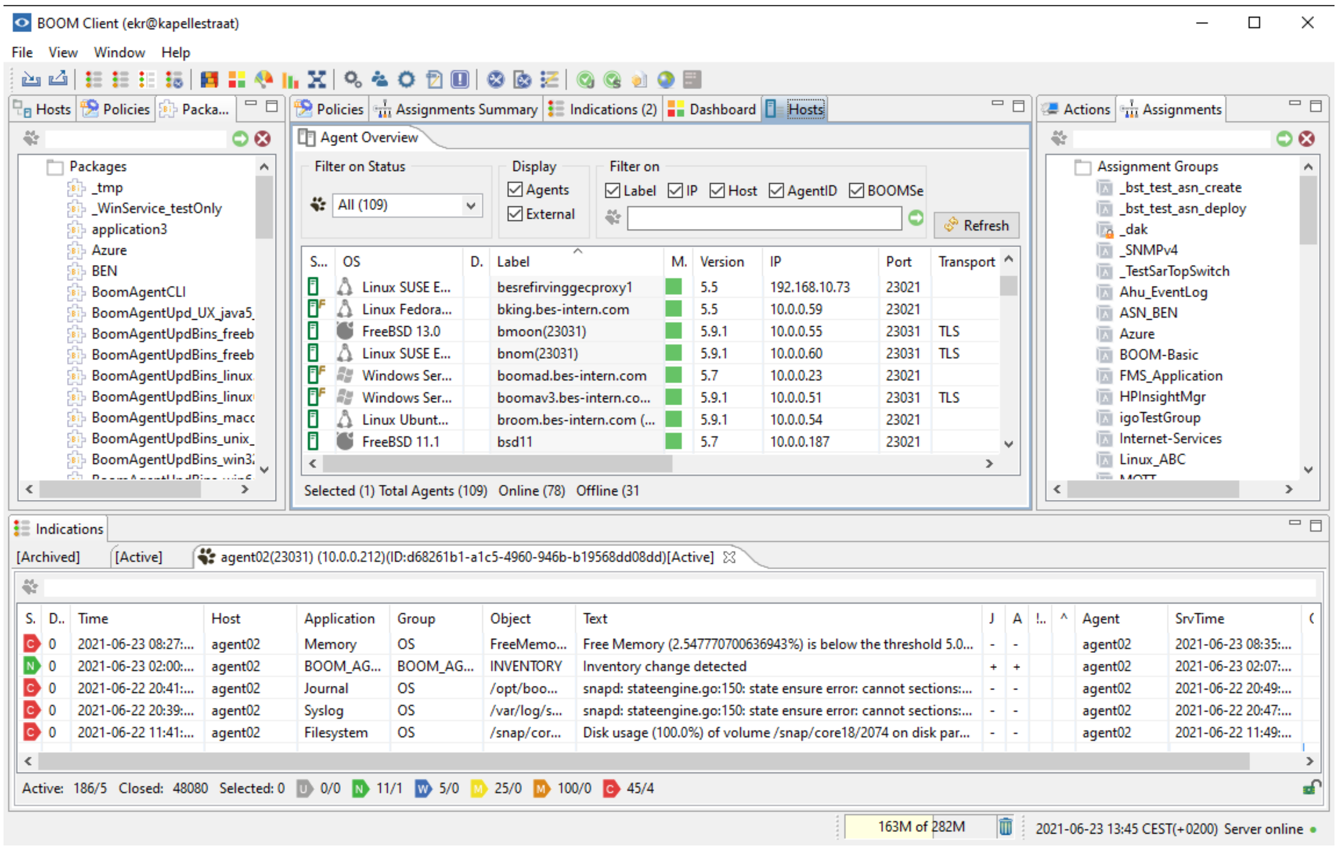
Deployment Queue

Ass...	Type	Name	Status	Agent	Action	User	Time
		DeploymentTestPkg2_1622102399459		molenstraat (10.0.1.20)	Undeploy	cli	2021-05-27 11:35:5
		DeploymentTestPkg1_1622117706516		molenstraat (10.0.1.20)	Undeploy	cli	2021-05-27 14:37:4

Running and Finished Deployments

Type	Size	Name	Status	Action	Agent	Start Time

186M of 281M
2021-06-23 13:37 CEST(+0200) Server online



S...	OS	D.	Label	M.	Version	IP	Port	Transport
	Linux SUSE E...		besrefirvinggecproxy1	🟩	5.5	192.168.10.73	23021	
F	Linux Fedora...		bking.bes-intern.com	🟩	5.5	10.0.0.59	23021	
	FreeBSD 13.0		bmoon(23031)	🟩	5.9.1	10.0.0.55	23031	TLS
	Linux SUSE E...		bnom(23031)	🟩	5.9.1	10.0.0.60	23031	TLS
F	Windows Ser...		boomad.bes-intern.com	🟩	5.7	10.0.0.23	23021	
F	Windows Ser...		boomav3.bes-intern.co...	🟩	5.9.1	10.0.0.51	23031	TLS
	Linux Ubunt...		broom.bes-intern.com (...	🟩	5.9.1	10.0.0.54	23021	
	FreeBSD 11.1		bsd11	🟩	5.7	10.0.0.187	23021	

Selected (1) Total Agents (109) Online (78) Offline (31

S.	D..	Time	Host	Application	Group	Object	Text	J	A	!..	^	Agent	SrvTime
C	0	2021-06-23 08:27:...	agent02	Memory	OS	FreeMemo...	Free Memory (2.547770700636943%) is below the threshold 5.0...	-	-			agent02	2021-06-23 08:35:...
N	0	2021-06-23 02:00:...	agent02	BOOM_AG...	BOOM_AG...	INVENTORY	Inventory change detected	+	+			agent02	2021-06-23 02:07:...
C	0	2021-06-22 20:41:...	agent02	Journal	OS	/opt/boo...	snapd: stateengine.go:150: state ensure error: cannot sections:...	-	-			agent02	2021-06-22 20:49:...
C	0	2021-06-22 20:39:...	agent02	Syslog	OS	/var/log/s...	snapd: stateengine.go:150: state ensure error: cannot sections:...	-	-			agent02	2021-06-22 20:47:...
C	0	2021-06-22 11:41:...	agent02	Filesystem	OS	/snap/cor...	Disk usage (100.0%) of volume /snap/core18/2074 on disk par...	-	-			agent02	2021-06-22 11:49:...

Active: 186/5 Closed: 48080 Selected: 0 U 0/0 N 11/1 W 5/0 M 25/0 M 100/0 C 45/4

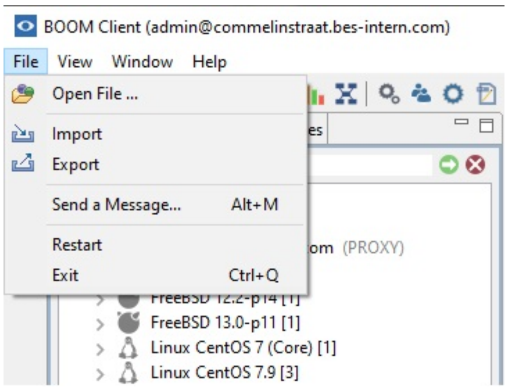
163M of 282M 2021-06-23 13:45 CEST(+0200) Server online

6.1.3. Menu Bar

The following dropdown menus are located in the menu bar:

- File Menu
- View Menu
- Window Menu
- Help Menu

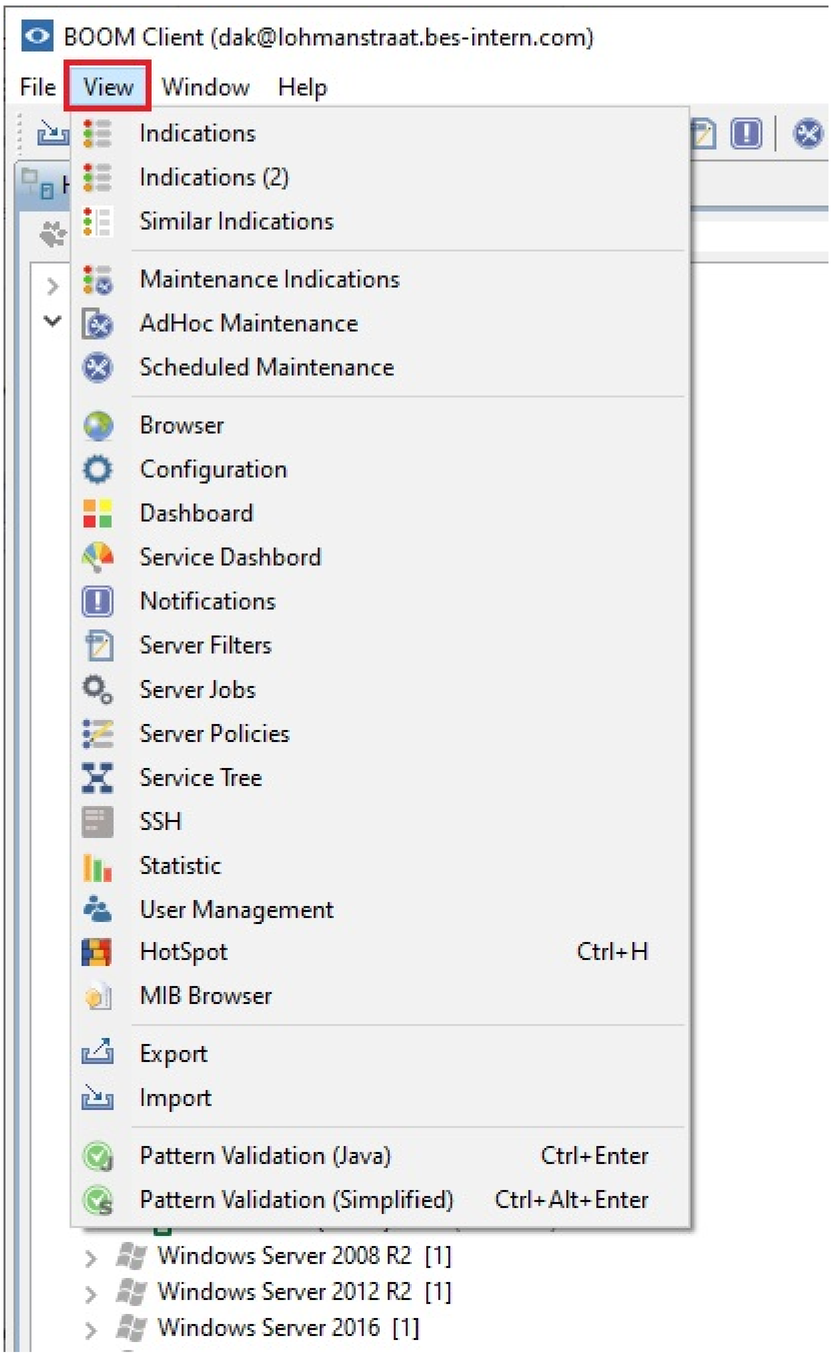
File Menu: The file menu allows standard file operations and helps to quickly perform tasks like import and export MPIs (Management Plug In's), broadcasting a message, restarting and exiting the GUI.



Open File: Standard file editor function
Show Import MPI View: Allows you to import policies, packages, assignments and actions.
Show Export MPI View: Allows you to export policies, packages, assignments and actions.
Send a message: Allows administrators to broadcast a message to all logged on *boom* users.

Restart: To restart the *boom* GUI client
Exit: This will exit the *boom* GUI client.

View Menu:



The view menu shows several views depending on your user role and rights.

Indications:

Opens the Indications Browser View. See chapter Indications.

Indications2:

Opens the Indications2 Browser View. See chapter Indications.

Similar Indications:

Opens the Indication Similarity View. See chapter Indication Similarity View.

Maintenance Indications:

Opens the Maintenance Indication View. See chapter Indication Maintenance.

AdHoc maintenance:

Opens the AdHoc Maintance View. See chapter AdHoc Maintenance.

Scheduled Maintenance:

Opens the Scheduled Maintenance View. See chapter Maintenance Window Management.

Browser:

Opens the web browser view. See chapter Browser.

Configuration:

Opens the Configuration view. See chapter Configuration View.

Dashboard:

For more information about the "Operator Dashboard" see chapter Operator Dashboard.

Service Dashboard:

For more information about the "Dashboard Services" see chapter Service Dashboard.

Notifications:

Opens the notification view, see chapter Notifications.

Server Filters:

For more information about "Server Filters" see chapter Server Filters.

Server Jobs:

For more information about "Server Jobs" see chapter Server Jobs.

Server Policies:

For more information about "Server Policies" see chapter Server Policies.

Service Tree:

For more information about the "Service Tree" see chapter Service Tree.

SSH:

For more information about the SSH terminal see chapter SSH Terminal.

Statistic:

Opens the browser with the *boom* server statistic page. For more information about "Statistics" see chapter Statistics.

User Management:

For more information about "User Management" see chapter User Management.

HotSpot:

For more information about the "HotSpot Dialog" see chapter HotSpot.

MIB Browser:

For more information about the "MIB Browser" see chapter MIB Browser.

Export MPI:

For more information about "Export MPI" see chapter Import/Export Management Plug-Ins.

Import MPI:

For more information about "Import MPI" see chapter Import/Export Management Plug-Ins.

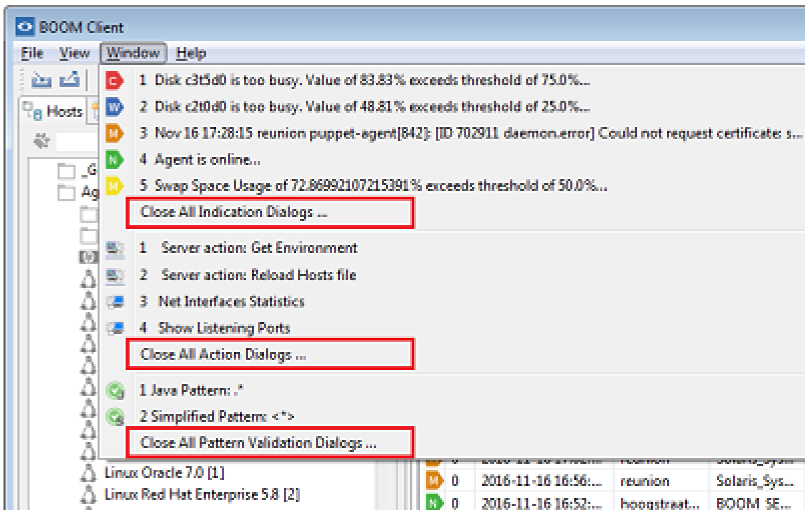
Pattern Validation Dialog (Java):

For more information about "Pattern Validation" see chapter Pattern Validation.

Pattern Validation Dialog (Simplified Java):

For more information about "Pattern Validation" see chapter Pattern Validation.

Window Menu:



1 Disk c3t5d0 is too busy. Value of 83.83% exceeds threshold of 75.0%...
2 Disk c2t0d0 is too busy. Value of 48.81% exceeds threshold of 25.0%...
3 Nov 16 17:28:15 reunion puppet-agent[842]: [ID 702911 daemon.error] Could not request certificate: s...
4 Agent is online...
5 Swap Space Usage of 72.86992107215391% exceeds threshold of 50.0%...
Close All Indication Dialogs ...
1 Server action: Get Environment
2 Server action: Reload Hosts file
3 Net Interfaces Statistics
4 Show Listening Ports
Close All Action Dialogs ...
1 Java Pattern: .*
2 Simplified Pattern: <*>
Close All Pattern Validation Dialogs ...

```
0  2016-11-16 16:56:...  reunion      Solaris_Sys...
0  2016-11-16 16:52:...  hoogstraat...  BOOM SE...
```

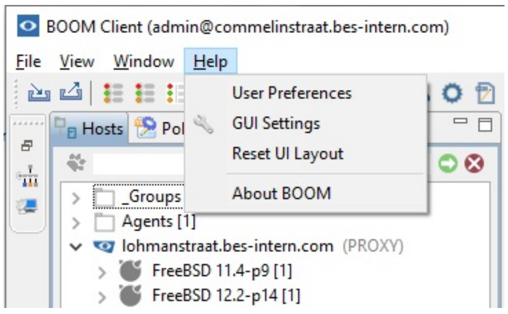
All open indication and action dialogs are managed in the window menu. The dialogs can be closed in groups with the following functions:

Close All Indication Dialogs ...
This function will become active if at least one "Indication Details Dialog" is open.

Close All Action Dialogs ...
This function will become active if at least one "Action Dialog" is open.

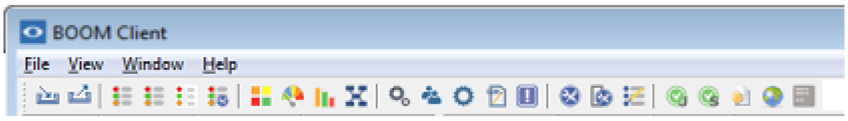
Close All Pattern Validation Dialogs ...
This function will become active if at least one "Pattern Validation Dialog" is open.

Help Menu:



6.1.4. Tool Bar

The tool bar contains icons for all optional views belonging to the user. The icons allow to quickly open or navigate to the according view. For the general overview about these views refer to the chapter Menu Bar.



Import MPI:

For more information about "Import MPI" see chapter Import/Export Management Plug-Ins.

Export MPI:

For more information about "Export MPI" see chapter Import/Export Management Plug-Ins.

Indications:

Opens the Indications Browser View. See chapter Indications.

Indications2:

Opens the Indications2 Browser View. See chapter Indications.

Similar Indications:

Opens the Indication Similarity View. See chapter Indication Similarity View.

Maintenance Indications:

Opens the Maintenance Indication View. See chapter Indication Maintenance.

Dashboard:

For more information about the "Operator Dashboard" see chapter Operator Dashboard.

Service Dashboard:

For more information about the "Dashboard Services" see chapter Service Dashboard.

Statistic:

Opens the browser with the *boom* server statistic page. For more information about "Statistics" see chapter Statistics.

Service Tree:

For more information about the "Service Tree" see chapter Service Tree.

Server Jobs:

For more information about "Server Jobs" see chapter Server Jobs.

User Management:

For more information about "User Management" see chapter User Management.

Configuration:

Opens the Configuration view. See chapter Configuration View.

Server Filters:

For more information about "Server Filters" see chapter Server Filters.

Notifications:

Opens the notification view, see chapter Notifications.

Scheduled Maintenance:

Opens the Scheduled Maintenance View. See chapter Maintenance Window Management.

AdHoc maintenance:

Opens the AdHoc Maintance View. See chapter AdHoc Maintenance.

Server Policies:

For more information about "Server Policies" see chapter Server Policies.

Pattern Validation Dialog (Java):

For more information about "Pattern Validation" see chapter Pattern Validation.

Pattern Validation Dialog (Simplified Java):

For more information about "Pattern Validation" see chapter Pattern Validation.

MIB Browser:

For more information about the "MIB Browser" see chapter MIB Browser.

Browser:

Opens the web browser view. See chapter Browser.

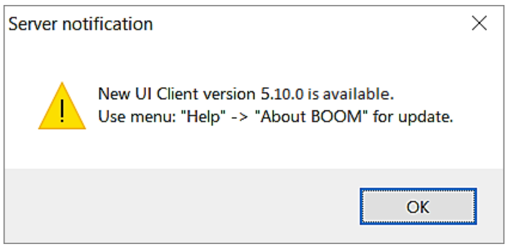
SSH:

For more information about the SSH terminal see chapter SSH Terminal.

6.1.5. *boom* UI update

Your *boom* Client will be automatically informed once a new client version is available.

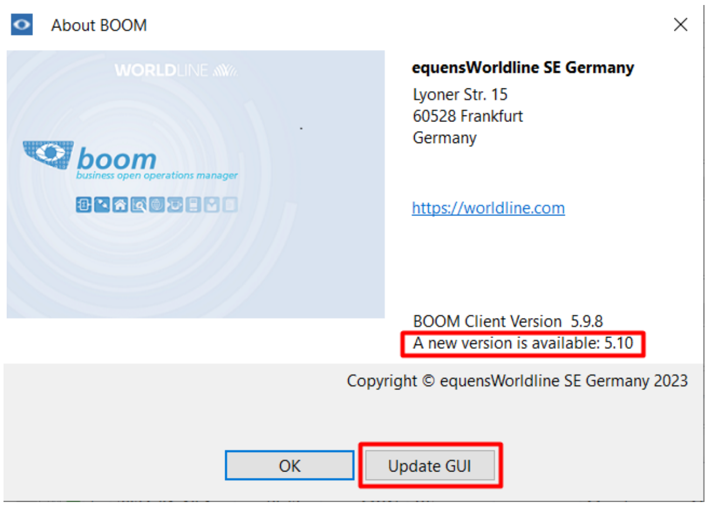
During startup, you get the following information:



Server notification

New UI Client version 5.10.0 is available.
Use menu: "Help" -> "About BOOM" for update.

In order to find out if there is a new version available on the server, open the **About *boom*** dialog in the **Help menu**.



About BOOM

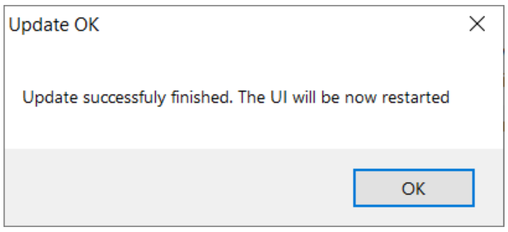
equensWorldline SE Germany
Lyoner Str. 15
60528 Frankfurt
Germany

https://worldline.com

BOOM Client Version 5.9.8
A new version is available: 5.10

Copyright © equensWorldline SE Germany 2023

Select **Update GUI** and the new version will be automatically installed from the *boom* server. The GUI will be restarted.

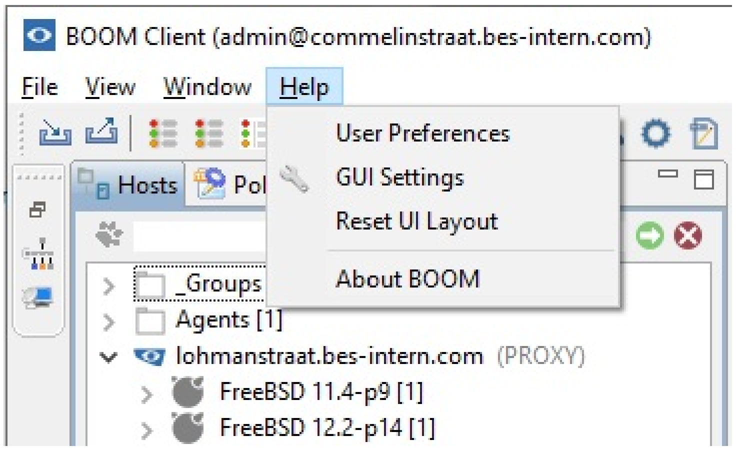


Update OK

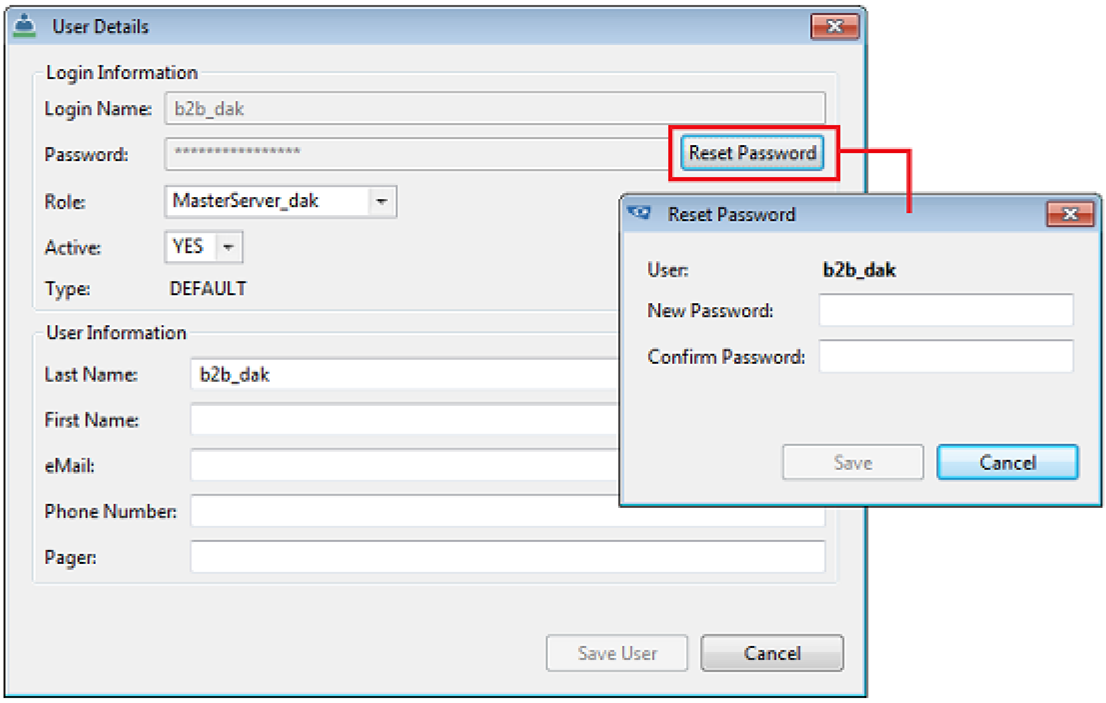
Update successfuly finished. The UI will be now restarted

6.1.6. Change UI Password

Please select **Help** and **User Preferences** to change your individual password.



The **user preferences** window will open. Select **Change password** and provide the new password.


User Details

Login Information

Login Name:	b2b_dak
Password:	****************
Role:	MasterServer_dak
Active:	YES
Type:	DEFAULT

User Information

Last Name:	b2b_dak
First Name:	
eMail:	
Phone Number:	
Pager:	

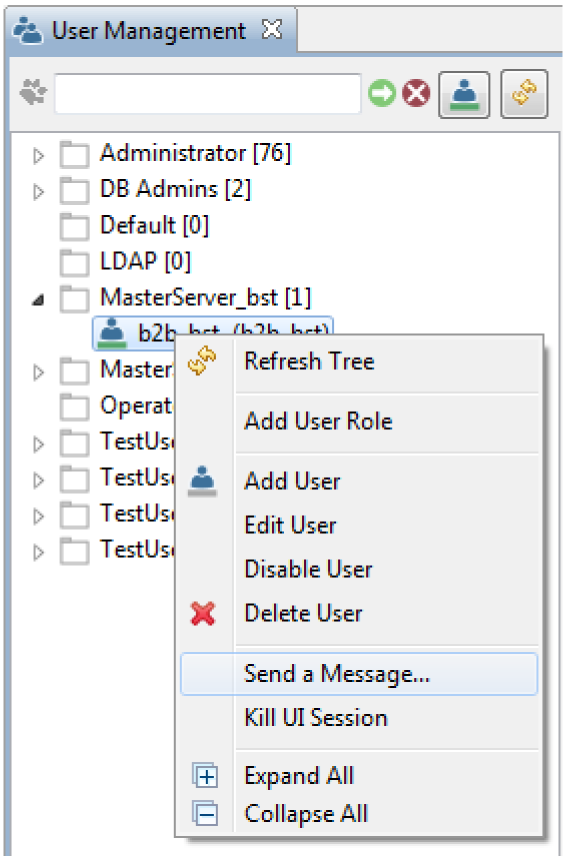
Reset Password

User:	b2b_dak
New Password:	
Confirm Password:	

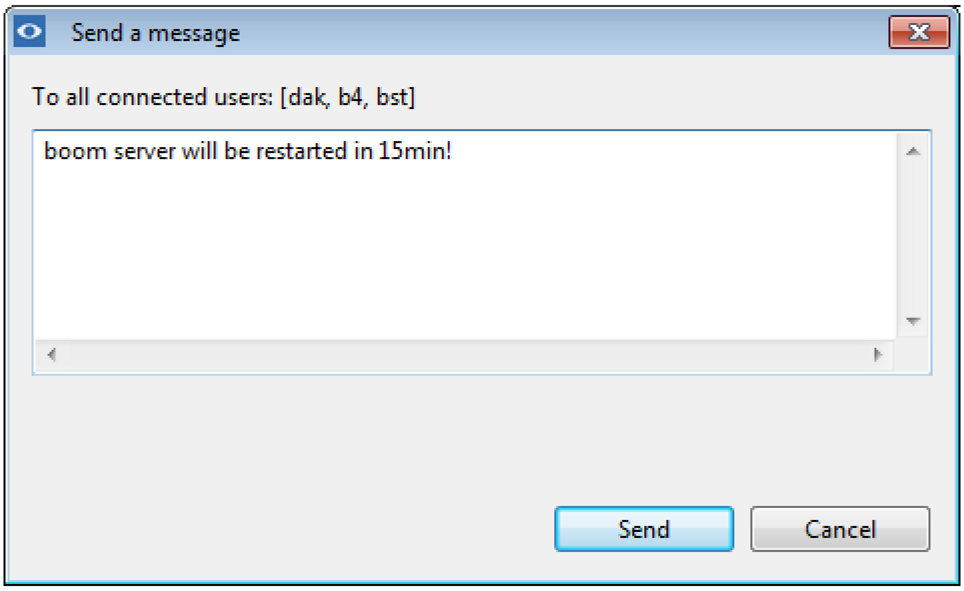
Select **Save** for saving the new password.

6.1.7. Admin Broadcasts & User Messaging

Users can selectively send messages by right clicking the destined logged on *boom* user under the user management tab and choose "Send a message..." for notification purposes.



Administrators can also broadcast a message to all logged on *boom* users via **ALT+M** shortcut (or selecting "Send a message..." under the File menu) for maintenance/notification purposes.

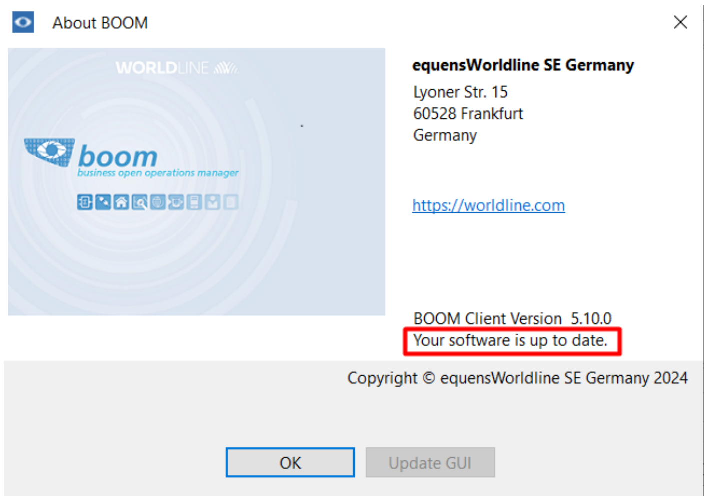


To all connected users: [dak, b4, bst]

boom server will be restarted in 15min!

6.1.8. *boom* Client Version

General information about the *boom* client can be found in the **About BOOM** dialog in the **Help menu**, including the version of the *boom* GUI client.



About BOOM

equensWorldline SE Germany
Lyoner Str. 15
60528 Frankfurt
Germany

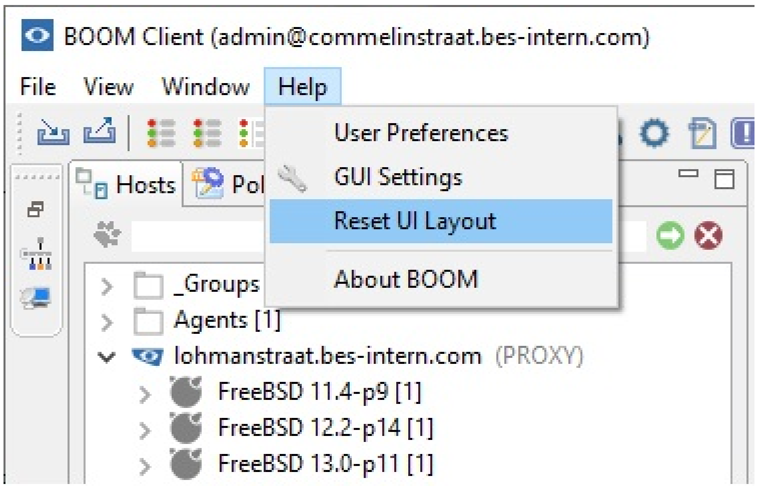
https://worldline.com

BOOM Client Version 5.10.0
Your software is up to date.

Copyright © equensWorldline SE Germany 2024

6.1.9. Reset UI Layout

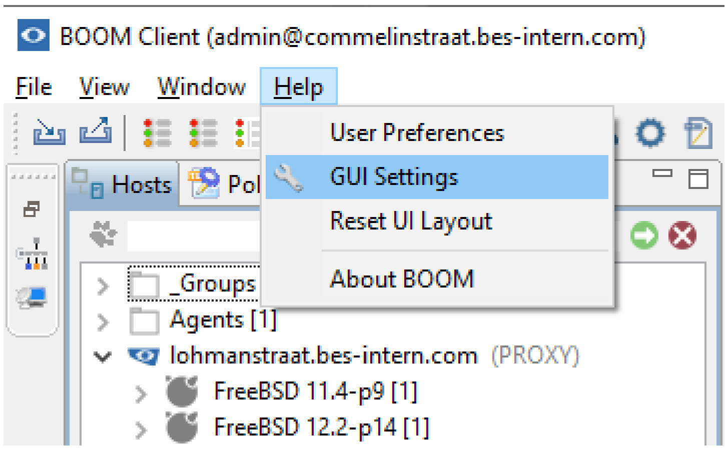
Please select **Help** and **Reset UI Layout** to reset the User Interface layout.



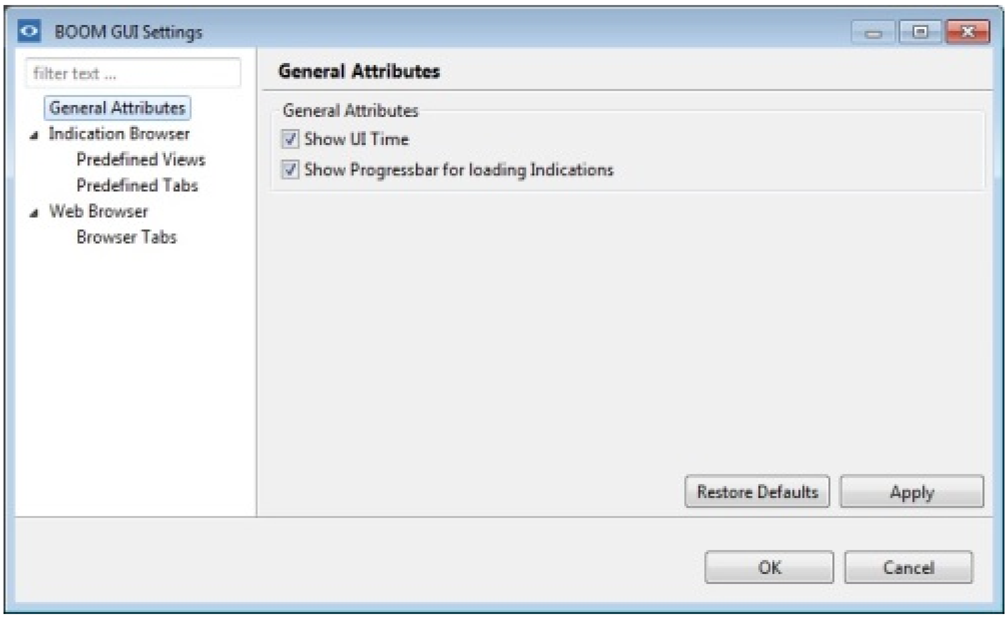
The UI will be restarted with the default values.

6.1.10. General UI Settings

Select **Help** and **GUI Setting** to customize your UI layout.



The ***boom* GUI Settings** window opens:

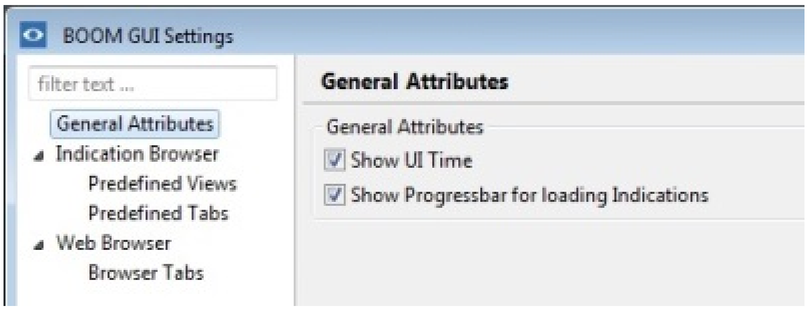


There are 3 sections for customizing the GUI:

- General Attributes
- Indication Browser
- Web Browser

6.1.10.1. General Attributes

Select General Attributes to change common attributes. No UI restart is necessary.

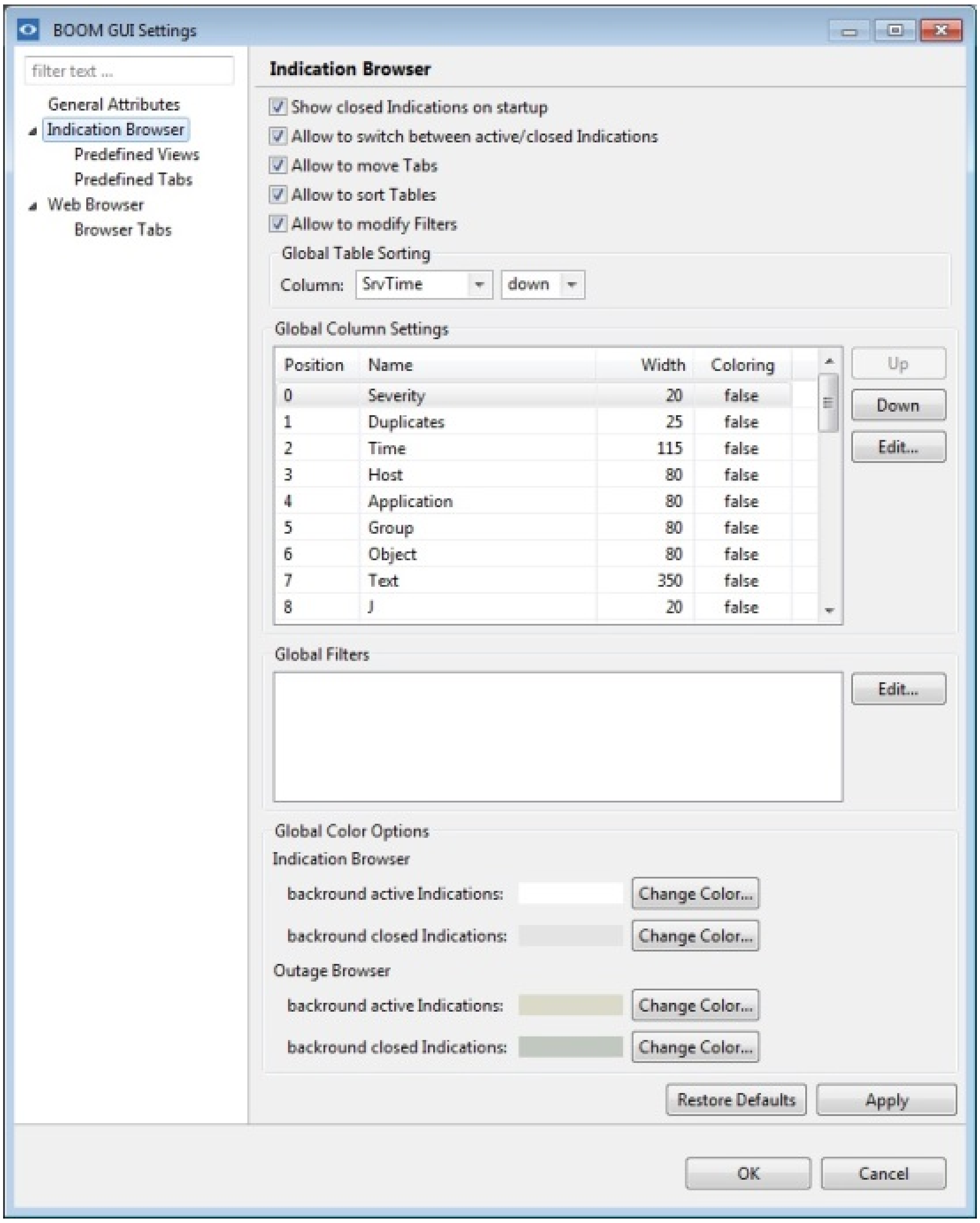


Show UI Time:
: Check mark to add date and time to the bottom status line

Show Progressbar for loading Indications:
: check mark to add the progressbar to the bottom status line

6.1.10.2. Indication Browser

Select **Indication Browser** to change general attributes concerning the Indication browser. The configuration parameters are valid for all Indications tabs.



Indication Browser

- ☑ Show closed Indications on startup
- ☑ Allow to switch between active/closed Indications
- ☑ Allow to move Tabs
- ☑ Allow to sort Tables
- ☑ Allow to modify Filters

Global Table Sorting

Column: SrvTime down

Global Column Settings

Position	Name	Width	Coloring
0	Severity	20	false
1	Duplicates	25	false
2	Time	115	false
3	Host	80	false
4	Application	80	false
5	Group	80	false
6	Object	80	false
7	Text	350	false
8	J	20	false

Global Filters

Global Color Options

Indication Browser

backround active Indications:

backround closed Indications:

Outage Browser

backround active Indications:

backround closed Indications:

Show closed Indications on startup:

Check mark to show the [Closed] Indications view per default

Allow to switch between active/closed Indications:

Check mark to enable the switching between [Active] and [Closed] Indication view in one view (Switch to Closed/Switch to Active)

Allow to move Tabs:

Check mark to allow to move indication tabs within the indication window

Allow to sort Tables:

Check mark to allow to sort an attribute table up-and-down

Allow to modify Filters:

Check mark to allow the modification of filters

Global Column Settings:

Allows to reorganize the attribute tables in the Indication views. Allows to activate coloring for every single attribute table in the Indication views.

Up/Down:

Change the order of the attribute tables in the indication views

Edit:

open the "Column Settings" window -→ de-/activate background coloring; modify the width of the column

Global Filters:

Edit:

add/modify predefined filters in the "Predefined Filter Dialog" window.

Global Color Options:

Gives the possibility to change the background colors of the following views:

Indication Browser

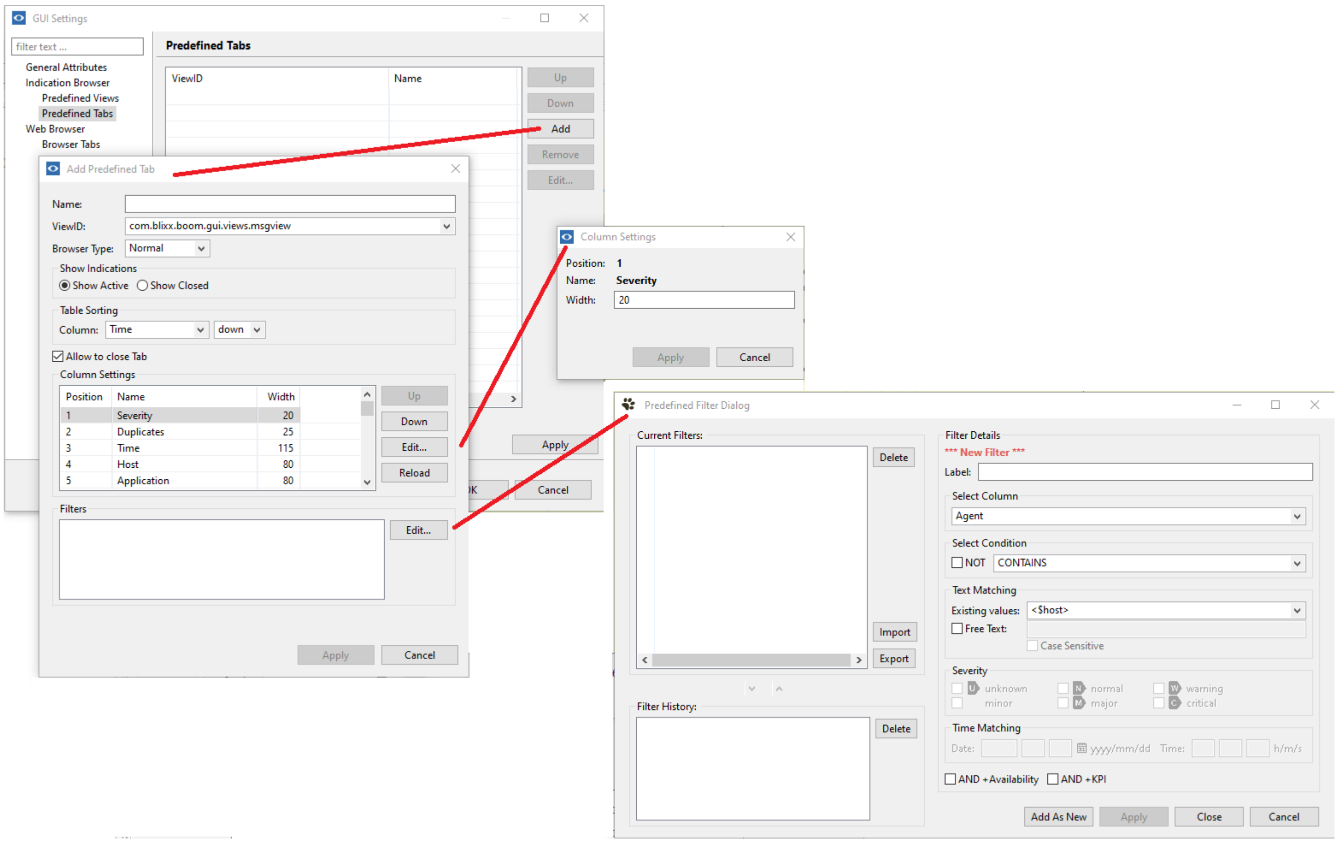
- background active Indications
- background closed Indications

Maintenance browser

- background active Indications
- background closed Indications

6.1.10.3. Predefined Tabs

The **Predefined Tabs** configuration can be used in combination with server based user profiles as described in *boom* User Interface Administration to define how many tabs will be opened as indication browser views for the user role.



Add Predefined Tab

Name:
ViewID: com.blixx.boom.gui.views.msgview
Browser Type: Normal
Show Indications: [✓ Show Active] Show Closed
Table Sorting – Column: Time down
☑ Allow to close Tab

Column Settings

Position	Name	Width
1	Severity	20
2	Duplicates	25
3	Time	115
4	Host	80
5	Application	80

Filters

Column Settings

Position: 1
Name: **Severity**
Width: 20

Predefined Filter Dialog

Filter Details
*** New Filter ***
Label:
Select Column: Agent
Select Condition: ☐ NOT CONTAINS
Text Matching – Existing values: <$host>
☐ Free Text:
☐ Case Sensitive
Severity: ☐ unknown ☐ normal ☐ warning ☐ minor ☐ major ☐ critical
Time Matching – Date: yyyy/mm/dd Time: h/m/s
☐ AND +Availability ☐ AND +KPI

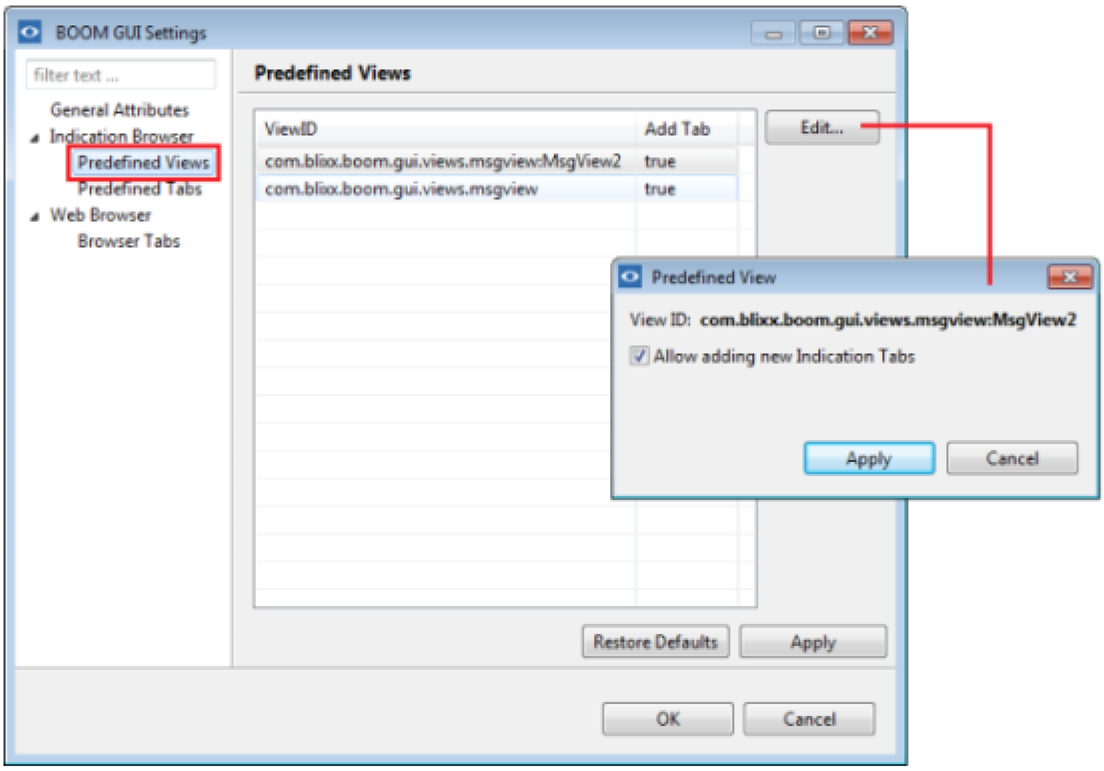
Predefined Tabs will always be opened at any start of the *boom* UI independent whether the Tab was manually closed by the user in the previous session or not.

The following parameters have to be defined:

- Define a tab name.
- Select the viewID of the Indication View where the tab should be opened. With the default *boom* installation this allows to open the tab in the main indication view or the indication view 2
- Define wether the tab should display active or closed indications.
- Select the column that is used for the default table sort order.
- Allow or disallow the user to close this tab. Please note that the tab will be opened again after a restart of the *boom* UI
- Arrange the column order and the column width.
- Add Filter conditions to restrict the Indications that are presented in the Tab.

6.1.10.4. Predefined Views

The **Predefined Views** configuration can be used in combination with server based user profiles as described in *boom* User Interface Administration to define the capabilities a user group will have on the indication browser views.



ViewID	Add Tab
com.blixx.boom.gui.views.msgview:MsgView2	true
com.blixx.boom.gui.views.msgview	true

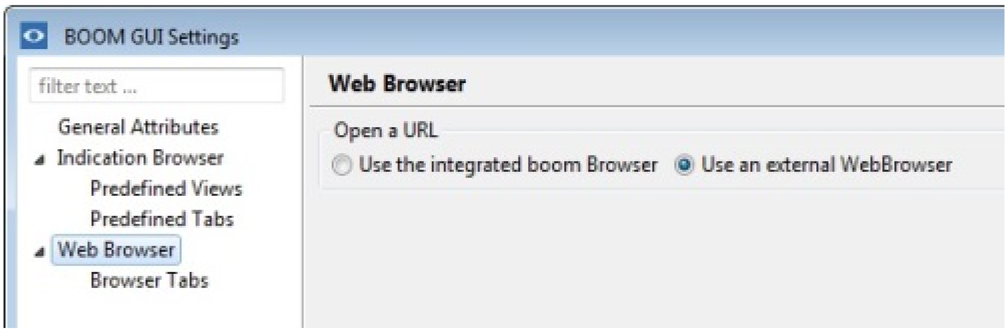
View ID: **com.blixx.boom.gui.views.msgview:MsgView2**

☑ Allow adding new Indication Tabs

With the *boom* default installation there are two Predefined Views, the Indication View (the main indication browser) and Indication View2 (typically used for the indication browser for closed indications).
These default views can not be deleted.
The **Add Tab** attribute defines if this user is allowed to add new Indication Tabs inside the view.

6.1.10.5. Web Browser

Defines if the integrated *boom* browser or an external WebBrowser is used for e.g. for advices and instruction URL.


BOOM GUI Settings

filter text ...

General Attributes
Indication Browser
Predefined Views
Predefined Tabs
Web Browser
Browser Tabs

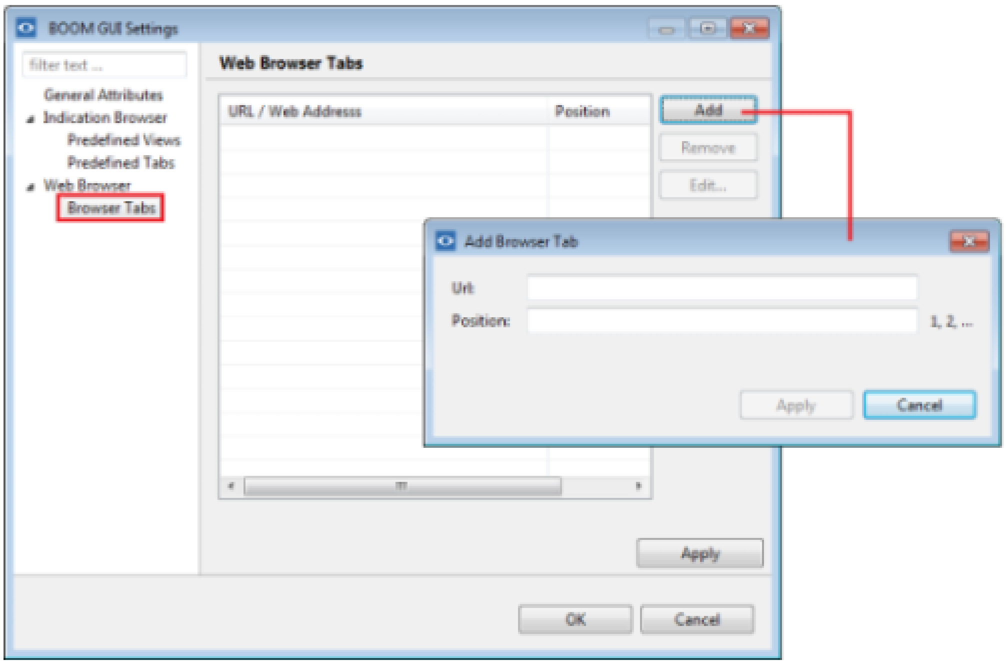
Web Browser

Open a URL

○ Use the integrated boom Browser [✓ ● Use an external WebBrowser]

6.1.10.6. Web Browser Tabs

The **Browser Tabs** configuration can be used in combination with server based user profiles as described in *boom* User Interface Administration to define how many tabs and which pages will be opened in the web browser view as tabs for the user role.



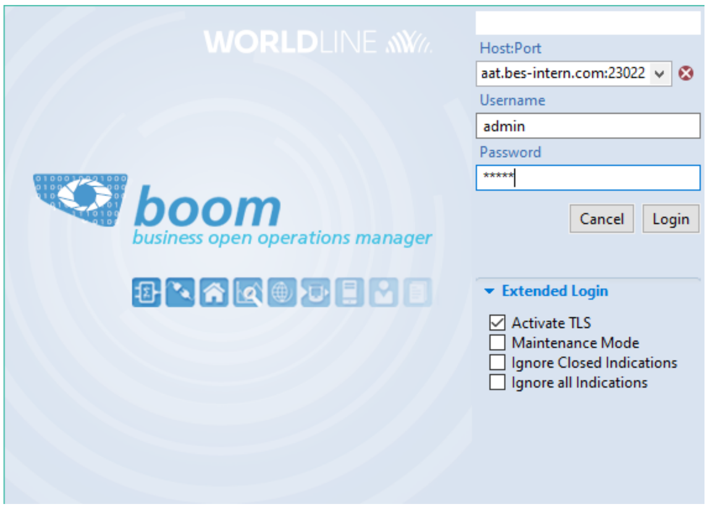
All Browser Tabs that are defined will be opened after any start oft he *boom* UI. You can add a new Browser Tab by defining the URL and the desired tab position.

6.2. Access and Login

6.2.1. Login to the *boom* Workbench

To access the web interface of the *boom* GUI, please follow the steps below:

1. Start the *boom* graphical user interface by executing the boomgui program.



2. Enter the *boom* server host name and port, your username and password.

Host:
mandatory - host name or IPv4/IPv6 address

Port:
optional
If no port is provided, the standard port will be used. If a port is provided, hostname and port have to be separated by a colon ":". E.g. localhost:23022, boom-dev01:23022

Username:
is case-sensitive

Password:
is case-sensitive

Extended Login:
optional

Initial Account is: admin/admin

Multiple logins of the same user are not possible.

Extended Login

Activate TLS:
Switch to enable/disable TLS while communicating with *boom* server. By default, the TLS is activated. If you want to use latest UI with older *boom* server version - please deactivate TLS.

Also, there are three special modes (select extended login) to start the GUI in large environments or as "Recovery Console".

Maintenance Mode:

The "Maintenance Mode" gives the possibility to start the GUI without loading any indications. The "Source Groups" view shows the number of active/closed indication for every host/source. There is **no automatic update** of this view. The context menu of a selected host/source lets you perform options such as cleaning (close/archive) indications of a selected host/source e.g. in case of a message storm.

In the **"Source Groups" view** select a source. Right-click provides the following options:

Load Indications (Selected):

Loads all indications of the selected group into the active indication view

Close Indications (Selected):

Closes all indications of the selected group

Archive Closed Indications (selected):

Archives all closed indications

Reload All Indications:

Refreshes/updates the "Source Groups" view

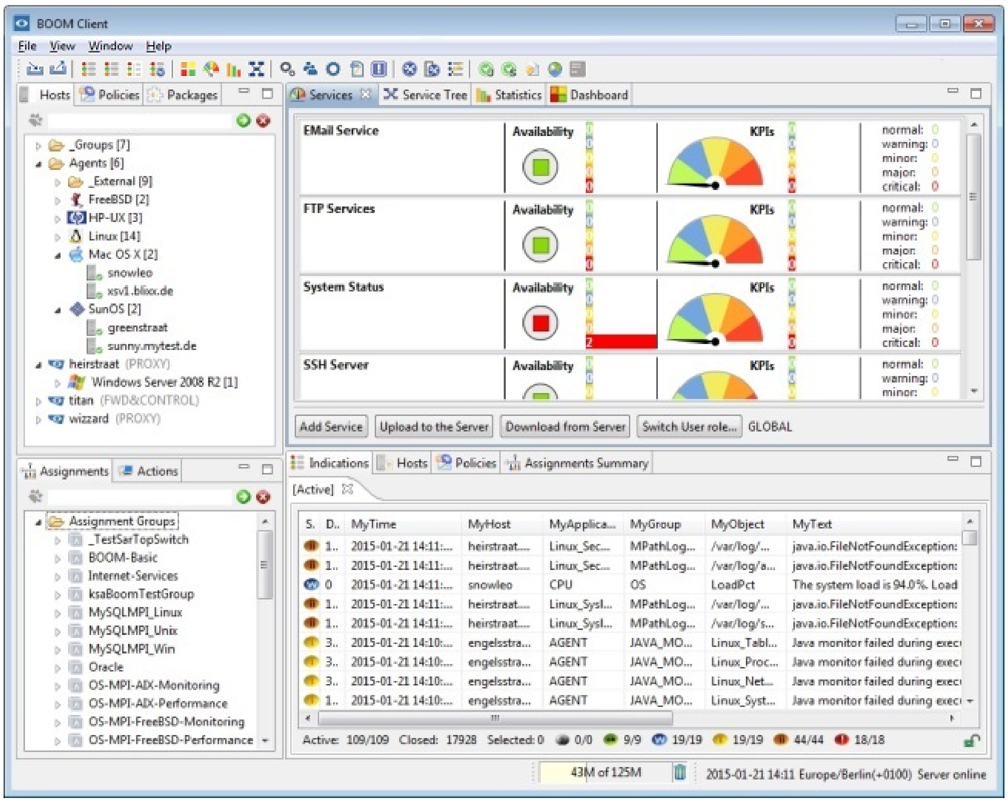
Ignore Closed Indications:

This mode allows you to load only active indications (i.e. history functions like graphs won’t work).

Ignore all Indications:

This mode starts the GUI without loading any indications.

3. After a successful login the *boom* workbench will open.



S.	D..	MyTime	MyHost	MyApplica...	MyGroup	MyObject	MyText
	1..	2015-01-21 14:11:...	heirstraat....	Linux_Sec...	MPathLog...	/var/log/...	java.io.FileNotFoundException:
	1..	2015-01-21 14:11:...	heirstraat....	Linux_Sec...	MPathLog...	/var/log/a...	java.io.FileNotFoundException:
	0	2015-01-21 14:11:...	snowleo	CPU	OS	LoadPct	The system load is 94.0%. Load
	1..	2015-01-21 14:11:...	heirstraat....	Linux_Sysl...	MPathLog...	/var/log/...	java.io.FileNotFoundException:
	1..	2015-01-21 14:11:...	heirstraat....	Linux_Sysl...	MPathLog...	/var/log/s...	java.io.FileNotFoundException:
	3..	2015-01-21 14:10:...	engelsstra...	AGENT	JAVA_MO...	Linux_Tabl...	Java monitor failed during exec
	3..	2015-01-21 14:10:...	engelsstra...	AGENT	JAVA_MO...	Linux_Proc...	Java monitor failed during exec
	3..	2015-01-21 14:10:...	engelsstra...	AGENT	JAVA_MO...	Linux_Net...	Java monitor failed during exec
	1..	2015-01-21 14:10:...	engelsstra...	AGENT	JAVA_MO...	Linux_Syst...	Java monitor failed during exec

Active: 109/109 Closed: 17928 Selected: 0 0/0 9/9 19/19 19/19 44/44 18/18

43M of 125M 2015-01-21 14:11 Europe/Berlin(+0100) Server online

To configure the default login parameters see chapter Setting Default Login Parameters

6.2.2. Login Problems

General error messages:

"Connection refused: connect"

Server is not available or server isn’t running.

"login/password invalid"

Incorrect login and/or password.

"Rejected. 'user' already connected from 'IPAddress'"

User is already connected from another system.

"Max number of clients reached."

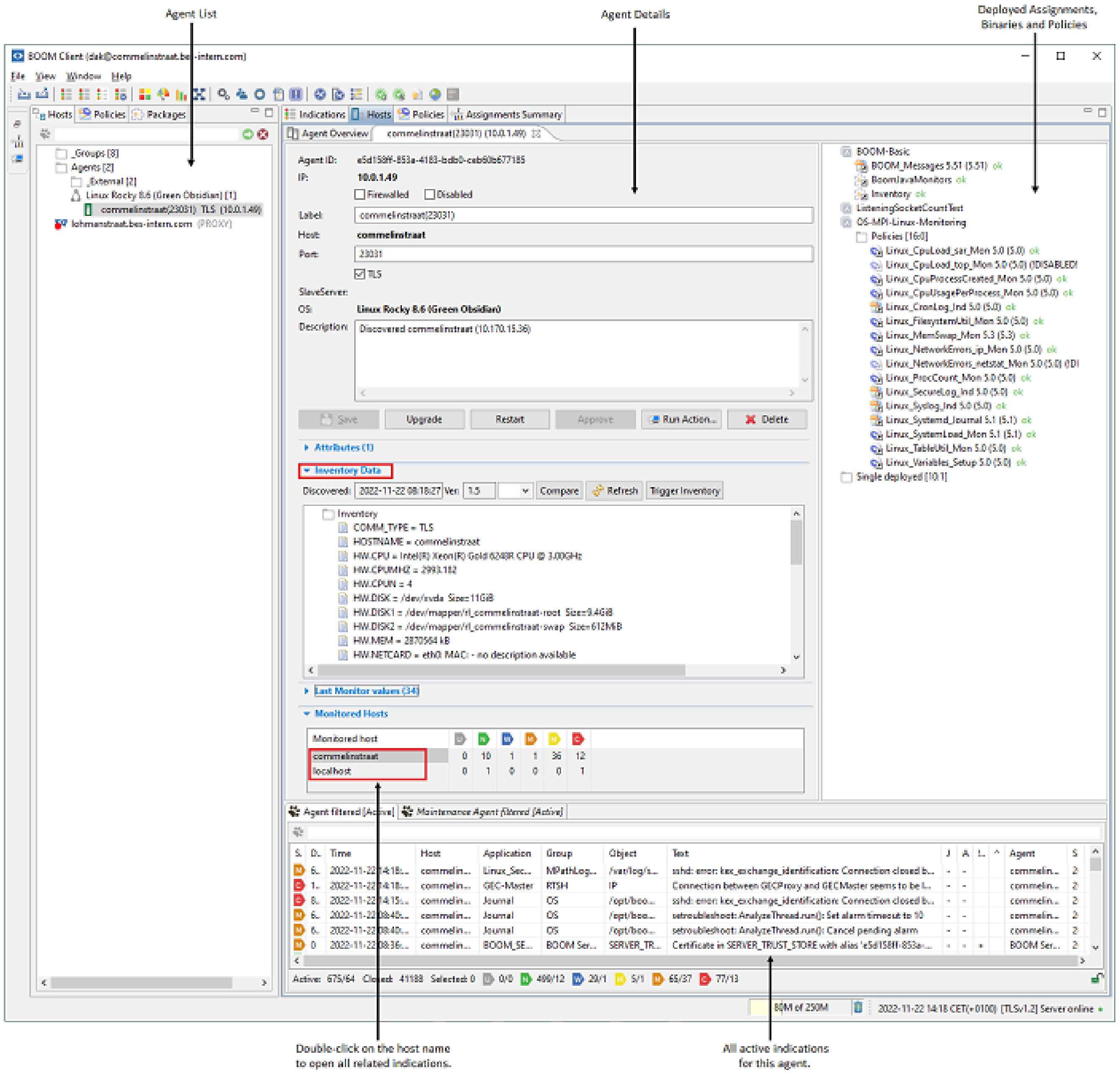
Maximum number of concurrent licensed users is reached.

ℹ	The concurrent user licensing was removed so this message will no longer show up.

6.3. Hosts / Agents

6.3.1. General Information

There are two different views belonging to the "Host/Agents" section. One view displays a list of all known agents and the second view contains all available agent details. The "Host List View" will be automatically opened after starting the *boom* workbench. Double-click on a single host in the tree to open the "Agent Details View":


Agent List
Agent Details
Deployed Assignments, Binaries and Policies
BOOM Client (dak@commelinstraat.bes-intern.com)
File View Window Help
Hosts Policies Packages
Indications Hosts Policies Assignments Summary
Agent Overview commelinstraat(23031) (10.0.1.49)
Save Upgrade Restart Approve Run Action... Delete
Attributes (1)
Inventory Data
Compare Refresh Trigger Inventory
Last Monitor values (34)
Monitored Hosts
Agent filtered [Active] Maintenance Agent filtered [Active]
Double-click on the host name to open all related indications.
All active indications for this agent.

_Groups [8]
Agents [2]
_External [2]
Linux Rocky 8.6 (Green Obsidian) [1]
commelinstraat(23031) TLS (10.0.1.49)
lohmanstraat.bes-intern.com (PROXY)

Agent ID: e5d158ff-853a-4183-bdb0-ceb60b677185
IP: 10.0.1.49
☐ Firewalled ☐ Disabled
Label: commelinstraat(23031)
Host: commelinstraat
Port: 23031
☑ TLS
SlaveServer:
OS: Linux Rocky 8.6 (Green Obsidian)
Description: Discovered commelinstraat (10.170.15.36)

Discovered: 2022-11-22 08:18:27 Ver: 1.5

```
Inventory
    COMM_TYPE = TLS
    HOSTNAME = commelinstraat
    HW.CPU = Intel(R) Xeon(R) Gold 6248R CPU @ 3.00GHz
    HW.CPUMHZ = 2993.182
    HW.CPUN = 4
    HW.DISK = /dev/vda  Size=11GiB
    HW.DISK1 = /dev/mapper/rl_commelinstraat-root  Size=9.4GiB
    HW.DISK2 = /dev/mapper/rl_commelinstraat-swap  Size=612MiB
    HW.MEM = 2870564 kB
    HW.NETCARD = eth0 MAC: - no description available
```

Monitored host	U	N	W	m	M	C
commelinstraat	0	10	1	1	36	12
localhost	0	1	0	0	0	1

```
BOOM-Basic
    BOOM_Messages 5.51 (5.51) ok
    BoomJavaMonitors ok
    Inventory ok
ListeningSocketCountTest
OS-MPI-Linux-Monitoring
    Policies [16:0]
        Linux_CpuLoad_sar_Mon 5.0 (5.0) ok
        Linux_CpuLoad_top_Mon 5.0 (5.0) (DISABLED!
        Linux_CpuProcessCreated_Mon 5.0 (5.0) ok
        Linux_CpuUsagePerProcess_Mon 5.0 (5.0) ok
        Linux_CronLog_Ind 5.0 (5.0) ok
        Linux_FilesystemUtil_Mon 5.0 (5.0) ok
        Linux_MemSwap_Mon 5.3 (5.3) ok
        Linux_NetworkErrors_ip_Mon 5.0 (5.0) ok
        Linux_NetworkErrors_netstat_Mon 5.0 (5.0) (DI
        Linux_ProcCount_Mon 5.0 (5.0) ok
        Linux_SecureLog_Ind 5.0 (5.0) ok
        Linux_Syslog_Ind 5.0 (5.0) ok
        Linux_Systemd_Journal 5.1 (5.1) ok
        Linux_SystemLoad_Mon 5.1 (5.1) ok
        Linux_TableUtil_Mon 5.0 (5.0) ok
        Linux_Variables_Setup 5.0 (5.0) ok
Single deployed [10:1]
```

S.	D..	Time	Host	Application	Group	Object	Text	J	A	I..	^	Agent	S
M	6..	2022-11-22 14:18:...	commelin...	Linux_Sec...	MPathLog...	/var/log/s...	sshd: error: kex_exchange_identification: Connection closed b...	-	-			commelin...	2
C	1..	2022-11-22 14:18:...	commelin...	GEC-Master	RTSH	IP	Connection between GECProxy and GECMaster seems to be l...	-	-			commelin...	2
C	8..	2022-11-22 14:15:...	commelin...	Journal	OS	/opt/boo...	sshd: error: kex_exchange_identification: Connection closed b...	-	-			commelin...	2
M	6..	2022-11-22 08:40:...	commelin...	Journal	OS	/opt/boo...	setroubleshoot: AnalyzeThread.run(): Set alarm timeout to 10	-	-			commelin...	2
M	6..	2022-11-22 08:40:...	commelin...	Journal	OS	/opt/boo...	setroubleshoot: AnalyzeThread.run(): Cancel pending alarm	-	-			commelin...	2
M	0	2022-11-22 08:36:...	commelin...	BOOM_SE...	BOOM Ser...	SERVER_TR...	Certificate in SERVER_TRUST_STORE with alias 'e5d158ff-853a-...	-	-	•		BOOM Ser...	2

Active: 675/64 Closed: 41188 Selected: 0 U 0/0 N 498/12 W 29/1 m 5/1 M 65/37 C 77/13

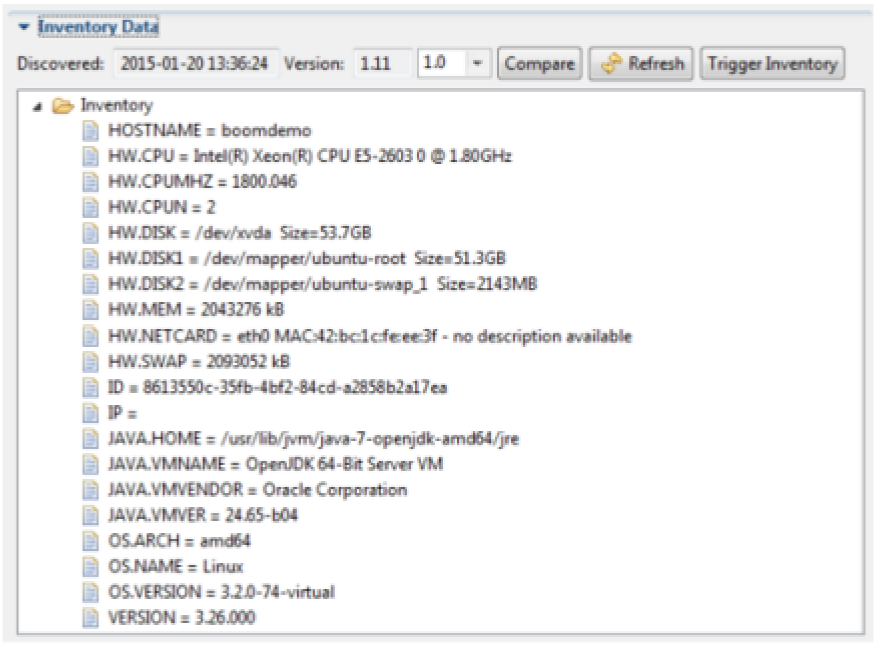
80M of 250M 2022-11-22 14:18 CET(+0100) [TLSv1.2] Server online

6.3.2. Agent Details and Agent Operations

Description of the agent details:

Agent ID:	The unique ID. Usually generated by the Agent during first start.
IP:	The IP address of the Agent. The IP will be automatically detected by the *boom* server, each time the agent connects to the server. It can be changed if the Agent is using DHCP or IP was changes manually.
Firewalled:	The flag is set when the *boom* server recognized this host as firewalled. It can be also set manually.
Disabled:	This will set the agent mode to 'disabled'. If the agent is disabled, it will only be able to run actions. The agent will not send any indications to the server. All policies are still available and in state "enabled". The agent will be marked with an (X).
Label:	This field contains the host label which appears on the hosts view and on the name field of the "Agent Overview" in the GUI. (Editable attribute)
Host:	The hostname provided by the Agent.
Port:	The agent port. (Editable attribute)
SlaveServer:	Name of slave *boom* server if available.
OS:	The operating system that is installed on the host. For virtual Agents (hosts without an installed agent) the OS is specified as *External*.
Description:	A description of the host. (Editable attribute)
Save:	Save changes that have been made to the "Agent Details".
Upgrade:	Updates the agent jar file on the remote system and restarts the agent.
Restart:	Restarts the agent on the remote system.
Approve:	Any new connected agent must be approved by the administrator to become operative. It's possible to configure AUTO_APPROVAL=true flag in boom.props configuration file of the server to activate automatic approval for all new Agents.
Run Action:	Run a remote action on the Agent.
Delete:	Deletes this Agent entry, all assignments and all links to node groups from the server.

Inventory Data:



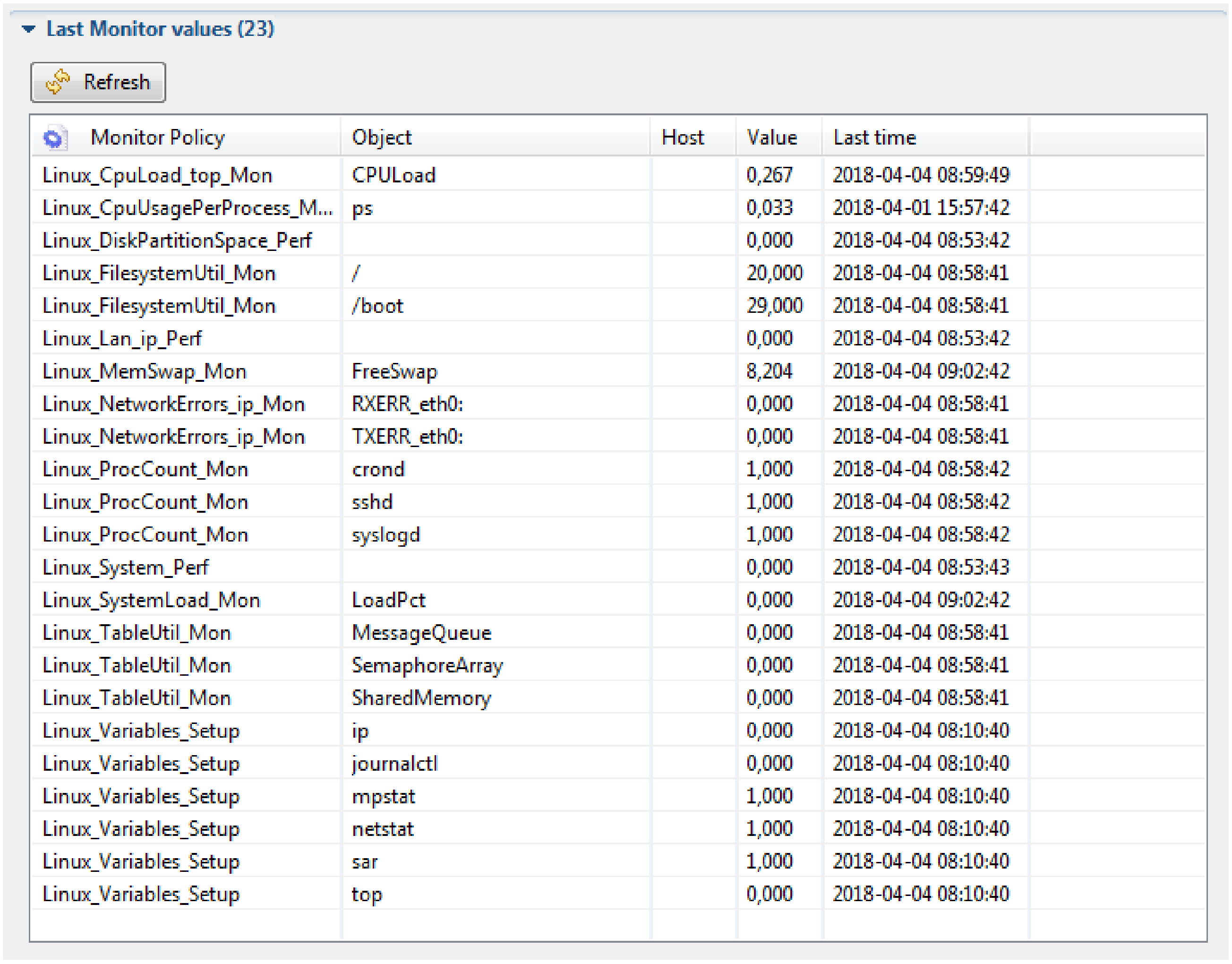
```
Discovered: 2015-01-20 13:36:24  Version: 1.11  1.0
Inventory
    HOSTNAME = boomdemo
    HW.CPU = Intel(R) Xeon(R) CPU E5-2603 0 @ 1.80GHz
    HW.CPUMHZ = 1800.046
    HW.CPUN = 2
    HW.DISK = /dev/xvda  Size=53.7GB
    HW.DISK1 = /dev/mapper/ubuntu-root  Size=51.3GB
    HW.DISK2 = /dev/mapper/ubuntu-swap_1  Size=2143MB
    HW.MEM = 2043276 kB
    HW.NETCARD = eth0 MAC:42:bc:1c:fe:ee:3f - no description available
    HW.SWAP = 2093052 kB
    ID = 8613550c-35fb-4bf2-84cd-a2858b2a17ea
    IP =
    JAVA.HOME = /usr/lib/jvm/java-7-openjdk-amd64/jre
    JAVA.VMNAME = OpenJDK 64-Bit Server VM
    JAVA.VMVENDOR = Oracle Corporation
    JAVA.VMVER = 24.65-b04
    OS.ARCH = amd64
    OS.NAME = Linux
    OS.VERSION = 3.2.0-74-virtual
    VERSION = 3.26.000
```

:: **Trigger Inventory**: Triggers the inventory discovery on the agent.

New version of Inventory data will be displayed only if any change was detected. Each Inventory version is stored on the server as separate file in srv/inventory/ folder.

If the "Inventory" binary package is not deployed to the Agent, discovered data does not contain most of hardware related information. The "Inventory" package contains platform specific scripts that help to the Agent get extended inventory information. It is possible to change these scripts to modify discovery data if necessary. NOTE: The "Inventory" binary package is part of the "BOOM-Basic" assignment group.

Last Monitor values:



Last Monitor values (23)

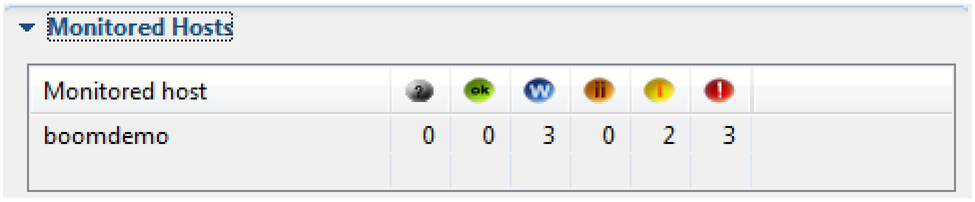
Refresh

Monitor Policy	Object	Host	Value	Last time
Linux_CpuLoad_top_Mon	CPULoad		0,267	2018-04-04 08:59:49
Linux_CpuUsagePerProcess_M...	ps		0,033	2018-04-01 15:57:42
Linux_DiskPartitionSpace_Perf			0,000	2018-04-04 08:53:42
Linux_FilesystemUtil_Mon	/		20,000	2018-04-04 08:58:41
Linux_FilesystemUtil_Mon	/boot		29,000	2018-04-04 08:58:41
Linux_Lan_ip_Perf			0,000	2018-04-04 08:53:42
Linux_MemSwap_Mon	FreeSwap		8,204	2018-04-04 09:02:42
Linux_NetworkErrors_ip_Mon	RXERR_eth0:		0,000	2018-04-04 08:58:41
Linux_NetworkErrors_ip_Mon	TXERR_eth0:		0,000	2018-04-04 08:58:41
Linux_ProcCount_Mon	crond		1,000	2018-04-04 08:58:42
Linux_ProcCount_Mon	sshd		1,000	2018-04-04 08:58:42
Linux_ProcCount_Mon	syslogd		1,000	2018-04-04 08:58:42
Linux_System_Perf			0,000	2018-04-04 08:53:43
Linux_SystemLoad_Mon	LoadPct		0,000	2018-04-04 09:02:42
Linux_TableUtil_Mon	MessageQueue		0,000	2018-04-04 08:58:41
Linux_TableUtil_Mon	SemaphoreArray		0,000	2018-04-04 08:58:41
Linux_TableUtil_Mon	SharedMemory		0,000	2018-04-04 08:58:41
Linux_Variables_Setup	ip		0,000	2018-04-04 08:10:40
Linux_Variables_Setup	journalctl		0,000	2018-04-04 08:10:40
Linux_Variables_Setup	mpstat		1,000	2018-04-04 08:10:40
Linux_Variables_Setup	netstat		1,000	2018-04-04 08:10:40
Linux_Variables_Setup	sar		1,000	2018-04-04 08:10:40
Linux_Variables_Setup	top		0,000	2018-04-04 08:10:40

Last values from all Monitor policies deployed on the Agent. A running Agent remembers last received value for each policy+object+value unique combinations. This information can be requested from Agent via remote action "Show Last Monitor Values". Similar to that the *boom* GUI makes a request to online Agent to get displayed data. Click on Refresh button will trigger a request to the Agent to get fresh dataset. An automatic request triggered by opening Agent details tab.

Only values and objects that match one of the Monitor policy condition will be displayed here.

Monitored Hosts:



Monitored Hosts

Monitored host	?	ok	W	ii	!	!
boomdemo	0	0	3	0	2	3

A list of all monitored hosts. Double-click on a host name to open all related indication in the "Indication View". This information is calculated by UI based on visible indications.

Attributes:

The following attributes are available and can be used:

Name	Type	Description
Email	varchar(255)	predefined attribute
Contact Person	varchar(1024)	predefined attribute

Name	Type	Description
Related Agent ID	varchar(255)	predefined attribute
External ID	varchar(255)	This can only be added for external hosts!
Long_1, Long_2 ... Long_10	bigint(20)	up to 10 attributes are available
Double_1, Double_2 ... Double_10	double	up to 10 attributes are available
Integer_1, Integer_2 ... Integer_10	Int(11)	up to 10 attributes are available
Timestamp_1, Timestamp_2 ... Timestamp_5	timestamp yyyy-[m]m-[d]d hh:mm:ss	up to 5 attributes are available
Text1024_1, Text1024_2 ... Text1024_15	varchar(1024)	up to 15 attributes are available

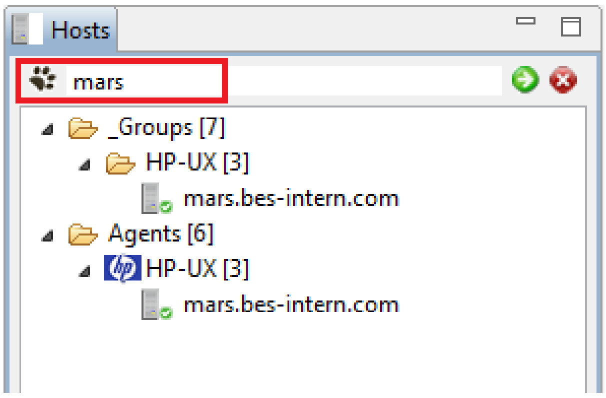
All custom attributes are stored in the *boom* database. The attribute labels can be modified by changing the value of the column ***label*** in the database table ***agents_ext_labels***. It is possible to reload Attribute labels and values without restarting server using server action "Reload Agent Attributes" (server action: RELOAD_AGENT_EXT_FROM_DB)

6.3.3. Agent List

Agent Status Symbols

Agent is running.
F Agent is running and Agent is firewalled.
Agent is not running.
F Agent is not running and Agent is firewalled.
(X) Agent is disabled.
? Agent is waiting for approval.

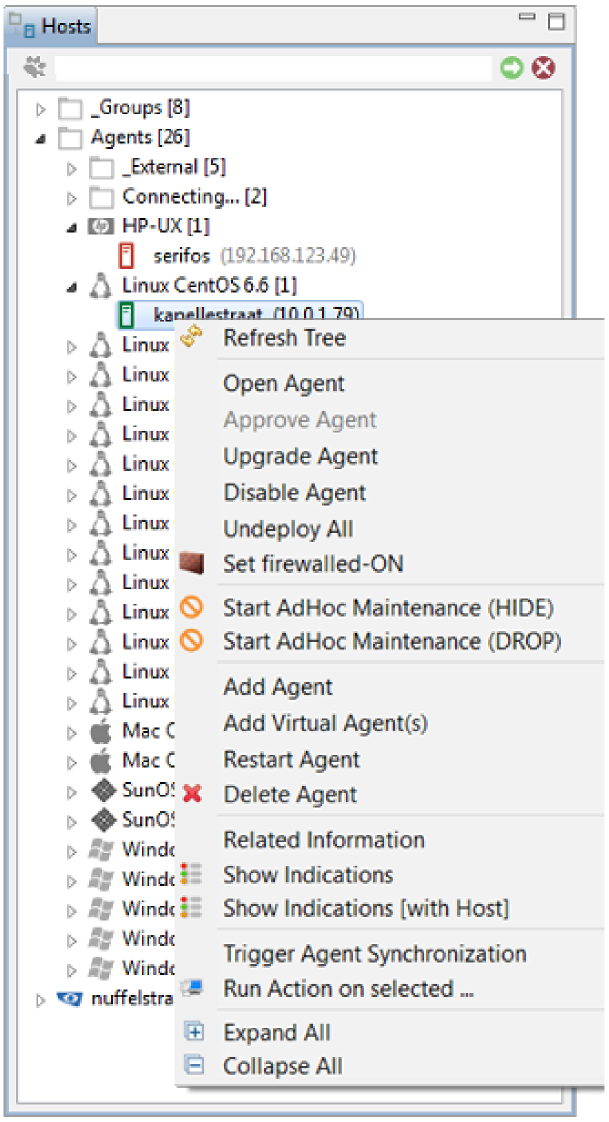
Search Function



Search text:

Specify the full host name or part of the name. The host tree opens with the search results.

Right-click to open the context menu:



Open Agent:

Opens the "Agent Details View".

Approve Agent:

Any automatically discovered agents must be approved by administrator before they are becoming functional.

Upgrade Agent:

Updates the agent jar file on the remote system and restarts the agent.

Disable Agent:

This will set the agent mode to 'disabled'. If the agent is disabled, it will only be able to run actions. The agent will not run any monitors or submit any messages. The agent will be marked with an (X).

Undeploy All:

Undeploy all known assignments from the agent.
Any deployed Assignment groups, policies and binary packages will be undepoyed from the agent.

Set firewalled-OFF/ON:

The flag is set when the *boom* Server recognized this host as firewalled. It can be also set manually.

Start AdHoc Maintenance (HIDE):

AdHoc Maintenance will be created with immediate start time, infinite duration and activated promptly. All incoming indications within this period will be moved to the Maintenance browser.

Start AdHoc Maintenance (DROP):

AdHoc Maintenance will be created with immediate start time, infinite duration and activated promptly. All incoming indications within this period will be dropped.

Add Agent:

By default the *boom* Server is able to discover and display any new installed *boom* agent automatically. When an agent has no access to the server due to the firewall settings (OUT only mode), it is necessary to manually add this system to the host list by providing the agent IP address. The server will then try to connect to the specified IP address and obtain the agent details.

Add Virtual Agent(s):

This will add one/more "virtual" agent(s) which are "external" host(s) without *boom* Agent.

Restart Agent:

This will restart the agent on the remote system.

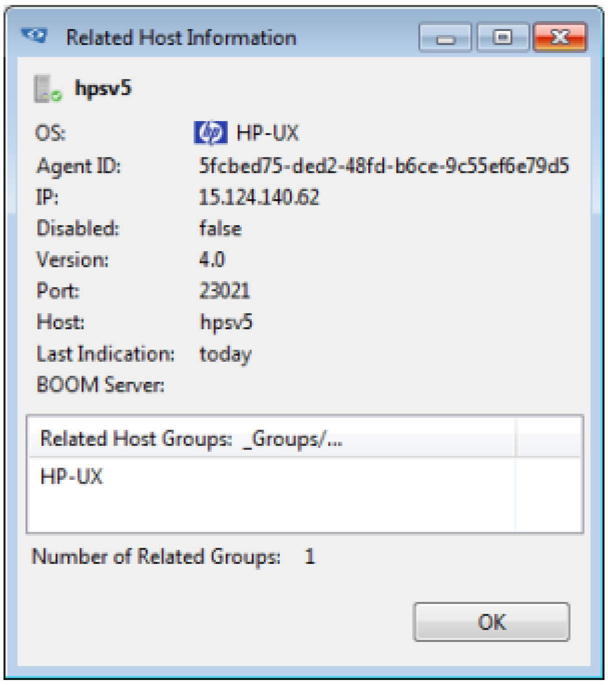
Delete Agent:

To delete an agent entry from the server.

Show Related Information:

This will display related information of the selected agent.

For example: status, operating system, IP, version, port, date of last indication and a list of all groups to which this agent is linked to.


Related Host Information

hpsv5

OS:	HP-UX
Agent ID:	5fcbed75-ded2-48fd-b6ce-9c55ef6e79d5
IP:	15.124.140.62
Disabled:	false
Version:	4.0
Port:	23021
Host:	hpsv5
Last Indication:	today
BOOM Server:	

Related Host Groups: _Groups/...
HP-UX

Number of Related Groups: 1

Show Indications:

Opens a new "Indication View" and displays all active indications delivered by selected agent.

All other context menu functionalities are already described in the "Agent Details" section above!

Trigger synchronization:

Allows a single agent or host group to synchronize its configuration with the actual *boom* server configuration if its configuration concerning policies and packages is no longer consistent with the server configuration. An undeploy/deploy is started automatically to synchronize the configurations.

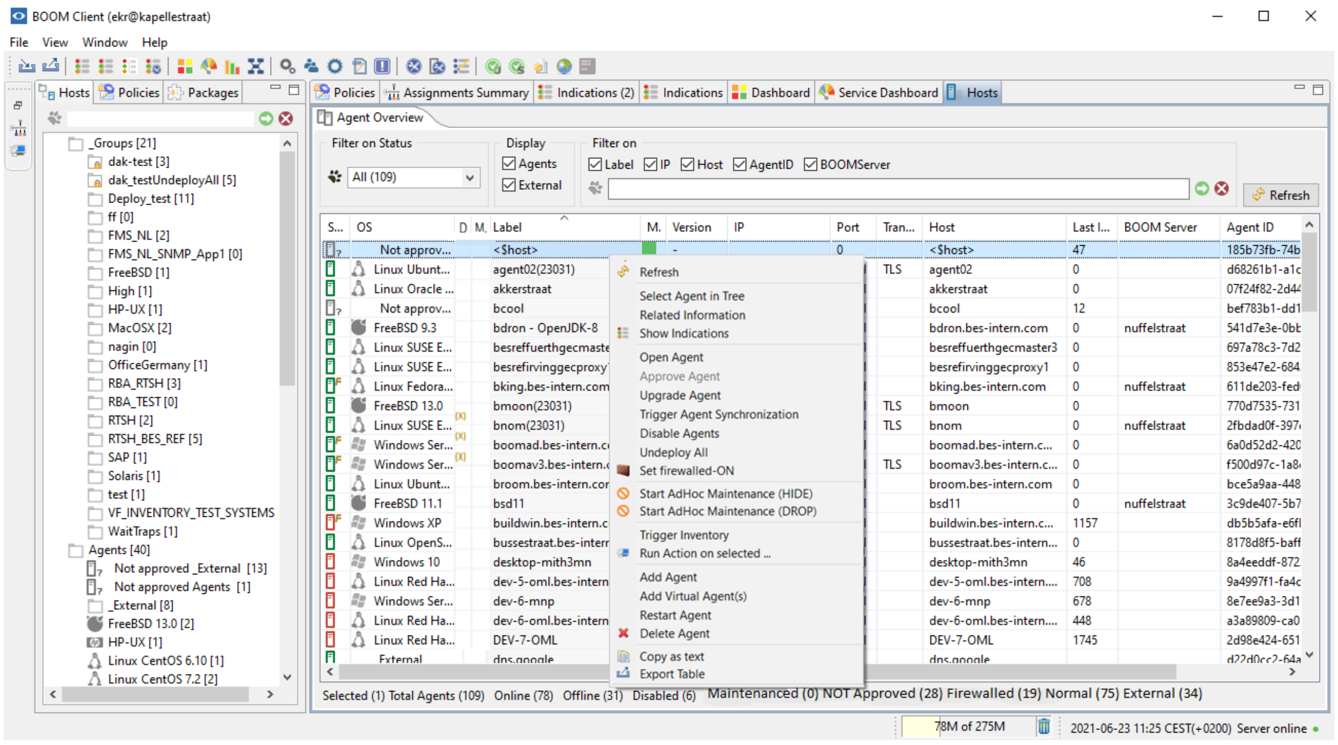
Run Action on selected ...

Opens the **Select Agent Action** view to select an action which should be executed on the specified agent.

Multiple agents can be selected.

6.3.4. Agent Overview

The **Agent Overview** view lists all agents. Right-click on a selected agent opens the context menu. See above section **agent list** for an explanation.



Filter on Status: All (109)

Display: [✓] Agents [✓] External

Filter on: [✓] Label [✓] IP [✓] Host [✓] AgentID [✓] BOOMServer

S...	OS	D	M.	Label	M.	Version	IP	Port	Tran...	Host	Last I...	BOOM Server	Agent ID
	Not approv...			<$host>	🟩	-		0		<$host>	47		185b73fb-74b
	Linux Ubunt...			agent02(23031)					TLS	agent02	0		d68261b1-a1c
	Linux Oracle ...			akkerstraat						akkerstraat	0		07f24f82-2d44
	Not approv...			bcool						bcool	12		bef783b1-dd1
	FreeBSD 9.3			bdron - OpenJDK-8						bdron.bes-intern.com	0	nuffelstraat	541d7e3e-0bb
	Linux SUSE E...			besreffuerthgecmaste						besreffuerthgecmaster3	0		697a78c3-7d2
	Linux SUSE E...			besrefirvinggecproxy						besrefirvinggecproxy1	0		853e47e2-684
	Linux Fedora...			bking.bes-intern.com						bking.bes-intern.com	0	nuffelstraat	611de203-fed
	FreeBSD 13.0			bmoon(23031)					TLS	bmoon	0		770d7535-731
	Linux SUSE E...	(X)		bnom(23031)					TLS	bnom	0	nuffelstraat	2fbdad0f-397
	Windows Ser...	(X)		boomad.bes-intern.c						boomad.bes-intern.c...	0		6a0d52d2-420
	Windows Ser...	(X)		boomav3.bes-intern.					TLS	boomav3.bes-intern.c...	0		f500d97c-1a8
	Linux Ubunt...			broom.bes-intern.cor						broom.bes-intern.com	0		bce5a9aa-448
	FreeBSD 11.1			bsd11						bsd11	0	nuffelstraat	3c9de407-5b7
	Windows XP			buildwin.bes-intern.c						buildwin.bes-intern.c...	1157		db5b5afa-e6f
	Linux OpenS...			bussestraat.bes-interr						bussestraat.bes-intern...	0		8178d8f5-baff
	Windows 10			desktop-mith3mn						desktop-mith3mn	46		8a4eeddf-872
	Linux Red Ha...			dev-5-oml.bes-intern						dev-5-oml.bes-intern....	708		9a4997f1-fa4c
	Windows Ser...			dev-6-mnp						dev-6-mnp	678		8e7ee9a3-3d1
	Linux Red Ha...			dev-6-oml.bes-intern						dev-6-oml.bes-intern....	448		a3a89809-ca0
	Linux Red Ha...			DEV-7-OML						DEV-7-OML	1745		2d98e424-651
	External			dns.google						dns.google			d22d0cc2-64a

Selected (1) Total Agents (109) Online (78) Offline (31) Disabled (6) Maintenanced (0) NOT Approved (28) Firewalled (19) Normal (75) External (34)

78M of 275M 2021-06-23 11:25 CEST(+0200) Server online

The above view offers multiple filter alternatives:

Filter on status:

All Agents(n)
Online Agents(n)
Offline Agents(n)
Disabled Agents(n)
Maintenanced Agents(n)
Not Approved Agents(n)
Firewalled Agents(n)
Not Firewalled Agents(n)

(n) specifies the number of matching hosts

Display:

Agents
External

Allows to view only agents, agent-less hosts or both

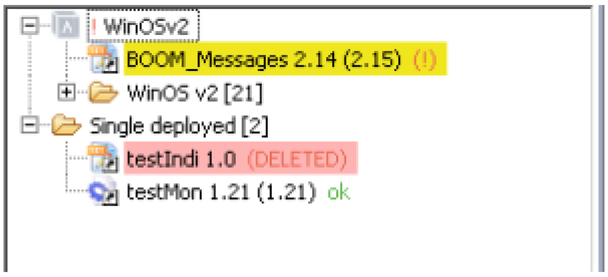
Filter on:

Label
IP
Host
AgentID
BOOMServer

Checkmark all fields that the filter will search. Press the arrow symbol to apply the filter or the cross to remove the filter.

6.3.5. Deployed Assignments and single deployed Policies

Assignment Status



! ***Conflict:***

The red exclamation mark indicates that there are some conflicts inside this assignment.

(!) ***Policy version conflict:***

Any changes to a policy will be saved with a new version. In this case the "BOOM_Messages" policy has been changed and then saved with the next version 2.15 but has not yet been redeployed to the agent. Right-click on the policy and deploy this policy. Do not forget to refresh the tree (right-click and choose "Refresh" from the content menu)

(DELETED) ***Policy has been deleted:***

Policy no longer exists on the *boom* server. To remove the policy from the deployed assignment tree right-click on the policy and undeploy it. Do not forget to refresh the tree (right-click and choose "Refresh" from the content menu)

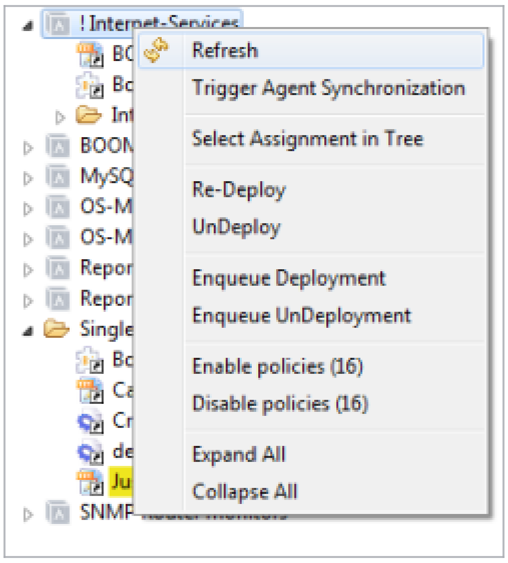
(! Unexpected by server !) Unexpected by server:

The policy is on the agent but the *boom* server does not have an entry that this policy should be on the agent

(X) Not deployed:

The policy is not deployed on the agent, but the *boom* server does have an entry that this policy should be on the agent

Context menu on Assignment level



Trigger synchronization:

Allows the agent to synchronize its configuration with the actual *boom* server configuration if its configuration concerning policies and packages is no longer consistent with the server configuration. An undeploy/deploy is started automatically to synchronize the configurations.

Select Assignment in Tree:

Points you directly to the selected assignment in the assignment view.

Re-Deploy:

Re-deploys the selected assignments to the appropriate *boom* agent.

UnDeploy:

Un-deploys the selected assignments from the appropriate *boom* agent.

Enqueue Deployment/Enqueue UnDeployment:

All "Enqueue Deployment" actions are exactly the same actions like the normal "Deployment" actions (described above) except that the "Enqueue Deployments" will NOT be started but added to the "Deployment Queue". For more details see chapter Assignments Summary.

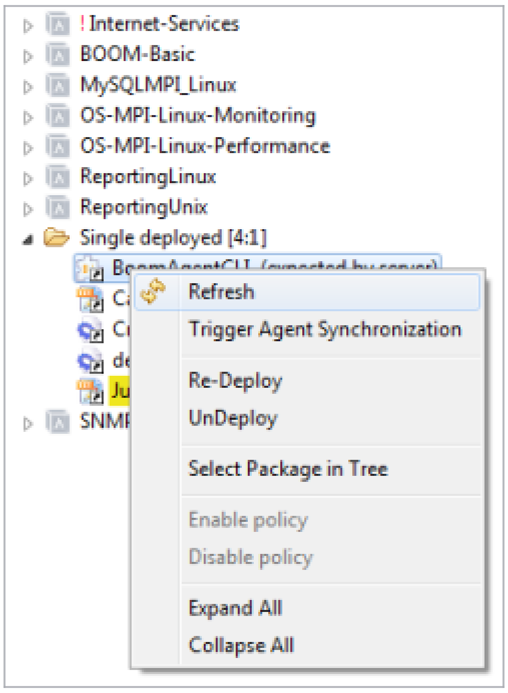
Enable policies:

Enables all policies of the selected assignment group. Select a single policy to enable only a single policy.

Disable policies:

Disables all policies of the selected assignment group. Select a single policy to disable only a single policy.

Context menu on single deployed level



Trigger synchronization:

Allows the agent to synchronize its configuration with the actual *boom* server configuration if its configuration concerning policies and packages is no longer consistent with the server configuration. An undeploy/deploy is started automatically to synchronize the configurations.

UnDeploy:

Un-deploys the selected single deployed policy or binary package from the appropriate *boom* agent.

Re-Deploy:

Re-deploys the selected single deployed policy or binary package to the appropriate *boom* agent.

Enqueue Deployment/Enqueue UnDeployment:

All "Enqueue Deployment" actions are exactly the same actions like the normal "Deployment" actions (described above) except that the "Enqueue Deployments" will **NOT** be started but added to the "Deployment Queue". For more details see the Chapter Assignments Summary.

Enable policy:

Enables a single policy.

Disable policy:

Disables a single policy.

6.3.6. Agent Auto-Inventory

During startup or runtime the agent checks the **Inventory Data** automatically every 24 hours and in case of differences from the previous state it increases the version of the Inventory Data and sends a message to the server. The *boom* server automatically fetches the newest inventory data from the agent. If the **inventory package** is not deployed, the agent reports back only a **minimal set of inventory data** like:

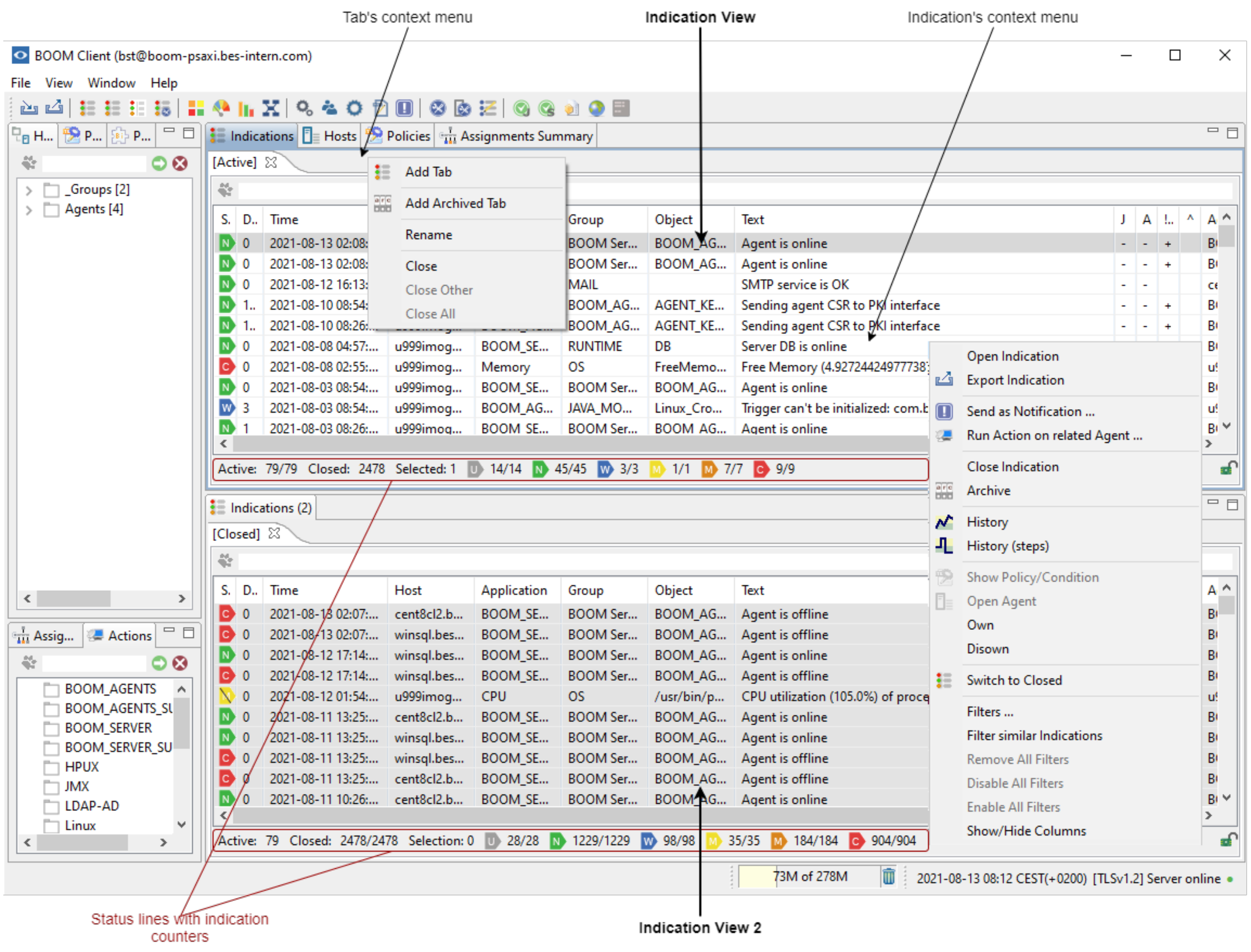
ID	agent ID
VERSION	agent version
HOSTNAME	agent hostname
IP	agent IP address
OS.NAME	OS name (i.e."Windows 7")
OS.VERSION	OS version (i.e."6.1")
OS.ARCH	Architecture of the system (i.e. "amd64")
HW.CPU	CPU Modell
HW.CPUMHZ	configured SPU Speed
HW.CPUN	number of CPUs
HW.DISK	Disks installed in the system
HW.MEM	physical Memory reported by the system
HW.SWAP	configured SWAP Space
JAVA.HOME	home directory of JVM running *boom* agent
JAVA.VMNAME	Java virtual machine implementation name
JAVA.VMVENDOR	JVM vendor name
JAVA.VMVER	JVM version

Deploy the inventory package only if you need a more detailed information about the system e.g. hard disks, network cards...

6.4. Indications

6.4.1. General Information

The *boom* workbench provides three Indication Views. By default the first view contains **all active Indications [Active]** and the second view all **closed Indications [Closed]**. Multiple browser tabs can be added for each indication view (open context menu by right-click on the 'View' header). Closing the last browser tab of indication view 2 (closed)) will close the complete section. The section can be reopened via the context menu. The third view shows all archived indications and isn't shown by default. Select "View Archived" in the context menu to open the archived view.



S.	D..	Time			Group	Object	Text
N	0	2021-08-13 02:08:			BOOM Ser...	BOOM_AG...	Agent is online
N	0	2021-08-13 02:08:			BOOM Ser...	BOOM_AG...	Agent is online
N	0	2021-08-12 16:13:			MAIL		SMTP service is OK
N	1..	2021-08-10 08:54:			BOOM_AG...	AGENT_KE...	Sending agent CSR to PKI interface
N	1..	2021-08-10 08:26:...			BOOM_AG...	AGENT_KE...	Sending agent CSR to PKI interface
N	0	2021-08-08 04:57:...	u999imog...	BOOM_SE...	RUNTIME	DB	Server DB is online
C	0	2021-08-08 02:55:...	u999imog...	Memory	OS	FreeMemo...	Free Memory (4.92724424977738
N	0	2021-08-03 08:54:...	u999imog...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online
W	3	2021-08-03 08:54:...	u999imog...	BOOM_AG...	JAVA_MO...	Linux_Cro...	Trigger can't be initialized: com.b
N	1	2021-08-03 08:26:...	u999imog...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online

Active: 79/79 Closed: 2478 Selected: 1 U 14/14 N 45/45 W 3/3 M 1/1 M 7/7 C 9/9

S.	D..	Time	Host	Application	Group	Object	Text
C	0	2021-08-13 02:07:...	cent8cl2.b...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is offline
C	0	2021-08-13 02:07:...	winsql.bes...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is offline
N	0	2021-08-12 17:14:...	winsql.bes...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online
C	0	2021-08-12 17:14:...	winsql.bes...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is offline
N	0	2021-08-12 01:54:...	u999imog...	CPU	OS	/usr/bin/p...	CPU utilization (105.0%) of proce
N	0	2021-08-11 13:25:...	cent8cl2.b...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online
N	0	2021-08-11 13:25:...	winsql.bes...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online
C	0	2021-08-11 13:25:...	winsql.bes...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is offline
C	0	2021-08-11 13:25:...	cent8cl2.b...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is offline
N	0	2021-08-11 10:26:...	cent8cl2.b...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online

Active: 79 Closed: 2478/2478 Selection: 0 U 28/28 N 1229/1229 W 98/98 M 35/35 M 184/184 C 904/904

73M of 278M 2021-08-13 08:12 CEST(+0200) [TLSv1.2] Server online

Tab's Context Menu

Add Tab:

Multiple indication tabs can be opened inside a View.

Rename:

Give the indication tab a new name.

Add Archived Tab:

Display all archived indications (see screenshot below).

Close:

Close tab.

Close Other:

Close tabs except active.

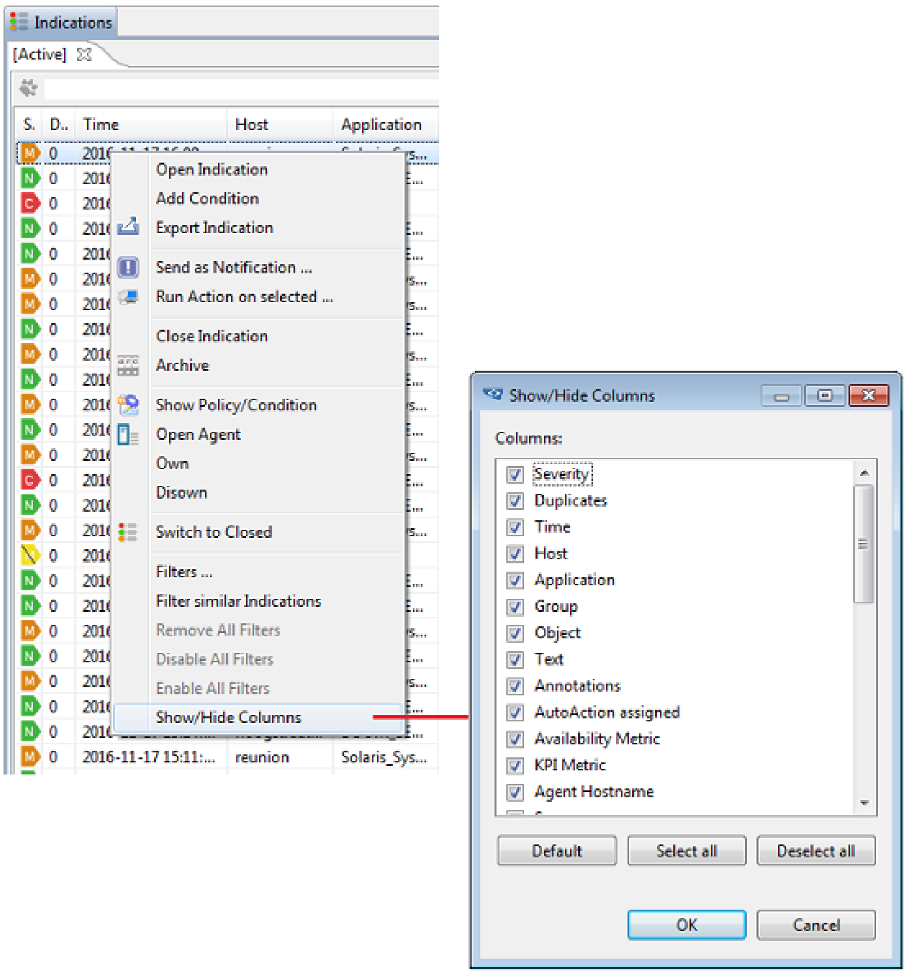
Close All:

Close all tabs.

6.4.2. Indication View Layout

You can reorganize the indication view for your individual needs. Right-click on the column header line of the indication view or any indication itself opens the context menu. Mark (see)/unmark (hide) the column attributes you want to show/hide in your individual layout.

Additionally you are able to reorder the attribute columns. Select an attribute column and move it to the favoured position.



Depending on the setup of the indication view the following information is displayed:

Annotations

The annotations column **J** is marked if the indication has annotations

Automatic actions

The Automatic actions column **AA** is marked if an automatic action is configured for the indication

Availability Metric

The availability column **!AV** is marked if the indication has an impact on a service availability

KPI

The KPI column **^KPI** is marked if the indication has an impact on a service level objective or is a Key

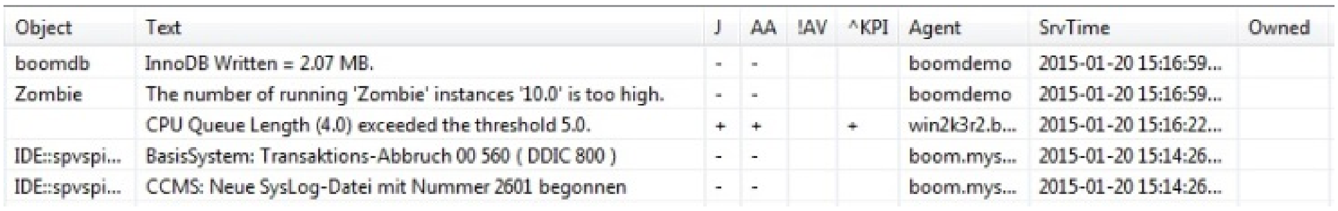
Performance Indicator

Owned

The Owned column will be marked and display the operator name if the indication is owned by an user.

Operator actions

are **not** separately marked in the indication view window.

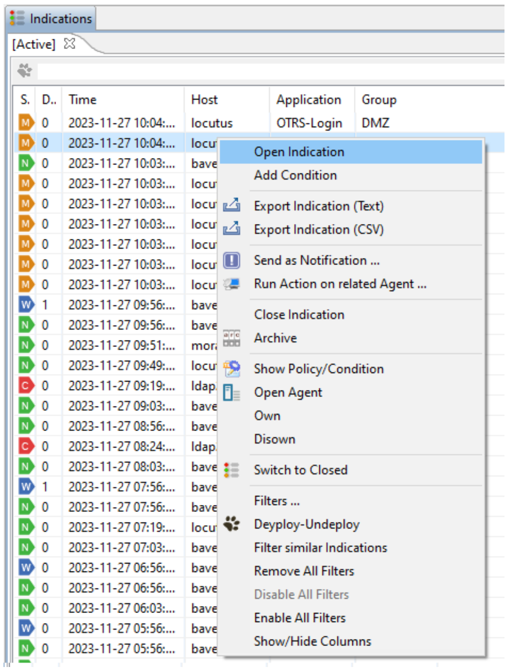


Object	Text	J	AA	!AV	^KPI	Agent	SrvTime	Owned
boomdb	InnoDB Written = 2.07 MB.	-	-			boomdemo	2015-01-20 15:16:59...	
Zombie	The number of running 'Zombie' instances '10.0' is too high.	-	-			boomdemo	2015-01-20 15:16:59...	
	CPU Queue Length (4.0) exceeded the threshold 5.0.	+	+		+	win2k3r2.b...	2015-01-20 15:16:22...	
IDE::spvspi...	BasisSystem: Transaktions-Abbruch 00 560 (DDIC 800)	-	-			boom.mys...	2015-01-20 15:14:26...	
IDE::spvspi...	CCMS: Neue SysLog-Datei mit Nummer 2601 begonnen	-	-			boom.mys...	2015-01-20 15:14:26...	

6.4.3. Indication Operations

6.4.3.1. Active Indication

Right-click on any ACTIVE Indication to open the Indication context menu:



S.	D..	Time	Host	Application	Group
M	0	2023-11-27 10:04:...	locutus	OTRS-Login	DMZ
M	0	2023-11-27 10:04:...	locu		
N	0	2023-11-27 10:03:...	bave		
M	0	2023-11-27 10:03:...	locu		
M	0	2023-11-27 10:03:...	locu		
M	0	2023-11-27 10:03:...	locu		
M	0	2023-11-27 10:03:...	locu		
M	0	2023-11-27 10:03:...	locu		
M	0	2023-11-27 10:03:...	locu		
W	1	2023-11-27 09:56:...	bave		
N	0	2023-11-27 09:56:...	bave		
N	0	2023-11-27 09:51:...	mor		
N	0	2023-11-27 09:49:...	locu		
C	0	2023-11-27 09:19:...	ldap		
N	0	2023-11-27 09:03:...	bave		
N	0	2023-11-27 08:56:...	bave		
C	0	2023-11-27 08:24:...	ldap		
N	0	2023-11-27 08:03:...	bave		
W	1	2023-11-27 07:56:...	bave		
N	0	2023-11-27 07:56:...	bave		
N	0	2023-11-27 07:19:...	locu		
N	0	2023-11-27 07:03:...	bave		
W	0	2023-11-27 06:56:...	bave		
N	0	2023-11-27 06:56:...	bave		
N	0	2023-11-27 06:03:...	bave		
W	0	2023-11-27 05:56:...	bave		
N	0	2023-11-27 05:56:...	bave		

Open Indication:

Opens the "Indications Details Dialog". For more information see chapter Indication Details.

Add Condition:

Only valid for indications coming from an text based (indication) policy. Allows you to add a condition to the policy that created this indication on the fly.

Export Indication:

One or more indications can be exported to a simple text or CSV file. For more information see chapter Indication Export.

Send as Notification:

allows to manually forward this indication to the notification service that is selected in the popup dialog.

Run Action on selected:

allows to select and run an action from the popup dialog on the agent that created the indication.

Close:

Closing one ore more indications will mark them as closed and also move them to the closed indications. To close one or more indications you can also use the hotkey 'c'.

Archive:

Move one or more active indications directly to the archived indications.

Show History/(Steps):

If the selected indication is created by a monitor policy the history data for this monitor can be displayed in a "History Chart". For more information about the "History Chart" see chapter History Chart.

Show Policy/Condition:

This opens the policy and selects the appropriate condition which triggered this indication.

Own:

Allows selected indications to be owned by the operator. This is only a visual indicator, anyone can work on the indication.

Disown:

Allows selected indications to be disowned by the operator. This is only a visual indicator, anyone can work on the indication.

Switch to Closed:

Displays all closed indications. You can also use the hotkey 's' to switch between active and closed indications.

Filters:

Click on filters will open the 'Filter Dialog'. For more details about filtering see chapter Indication Filtering.

Filter similar Indications:

Lists all indications containing the same application, group and object attribute as the selected indication.

Remove ALL Filters:

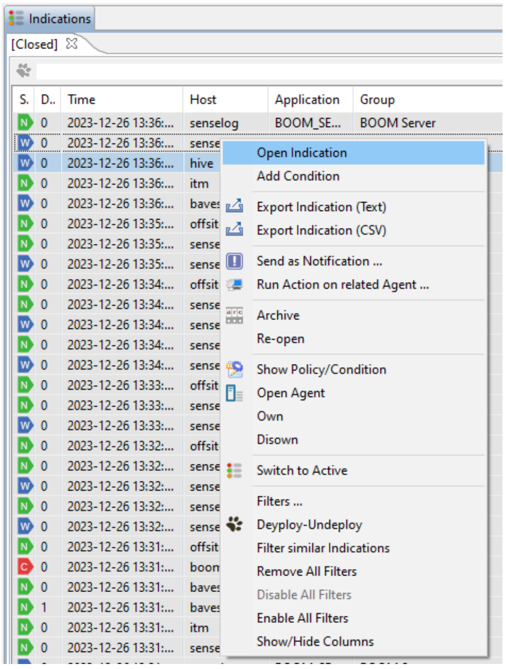
Removes all set filters and reloads all indications.

Show/Hide Columns:

Mark the attributes you want to see in the indication view layout.

6.4.3.2. Closed Indication

Right-click on any CLOSED Indication to open the Indication context menu:



S.	D..	Time	Host	Application	Group
N	0	2023-12-26 13:36:...	senselog	BOOM_SE...	BOOM Server
W	0	2023-12-26 13:36:...	sense		
W	0	2023-12-26 13:36:...	hive		
N	0	2023-12-26 13:36:...	itm		
W	0	2023-12-26 13:36:...	baves		
N	0	2023-12-26 13:35:...	offsit		
N	0	2023-12-26 13:35:...	sense		
W	0	2023-12-26 13:35:...	sense		
N	0	2023-12-26 13:34:...	offsit		
N	0	2023-12-26 13:34:...	sense		
W	0	2023-12-26 13:34:...	sense		
N	0	2023-12-26 13:34:...	sense		
W	0	2023-12-26 13:34:...	sense		
N	0	2023-12-26 13:33:...	offsit		
N	0	2023-12-26 13:33:...	sense		
W	0	2023-12-26 13:33:...	sense		
N	0	2023-12-26 13:32:...	offsit		
N	0	2023-12-26 13:32:...	sense		
W	0	2023-12-26 13:32:...	sense		
N	0	2023-12-26 13:32:...	sense		
W	0	2023-12-26 13:32:...	sense		
N	0	2023-12-26 13:31:...	offsit		
C	0	2023-12-26 13:31:...	boon		
N	0	2023-12-26 13:31:...	baves		
N	1	2023-12-26 13:31:...	baves		
N	0	2023-12-26 13:31:...	itm		
N	0	2023-12-26 13:31:...	sense		

Open Indication:

Opens the "Indications Details Dialog". For more information see chapter Indication Details.

Export Indication:

One or more indications can be exported to a simple txt file. For more information see chapter simple txt file. For more information see chapter Indication Export.

Send as Notification:

allows to manually forward this indication to the notification service that is selected in the popup dialog.

Run Action on selected:

allows to select and run an action from the popup dialog on the agent that created the indication.

Archive:

Archived indications will be removed from the indication browser and from the "History Chart".

Re-open:

Reopen the indication as active. An annotation "re-opened by" is added to the indication.

Show History/(Steps):

If there are any history data available for this Indication, they will be shown in a "History Chart". For more information about the "History Chart" see chapter History Chart.

Show Policy/Condition:

This opens the policy and selects the appropriate condition which is related to this indication.

Switch to Active:

Display all active indications. You can also use the hotkey 's' to switch between active and closed indications.

Filters:

Click on filters will open the 'Filter Dialog'. For more details about filtering see chapter Indication Filtering.

Filter similar Indications:

Lists all indications containing the same application, group and object attribute as the selected indication.

Remove ALL Filters:

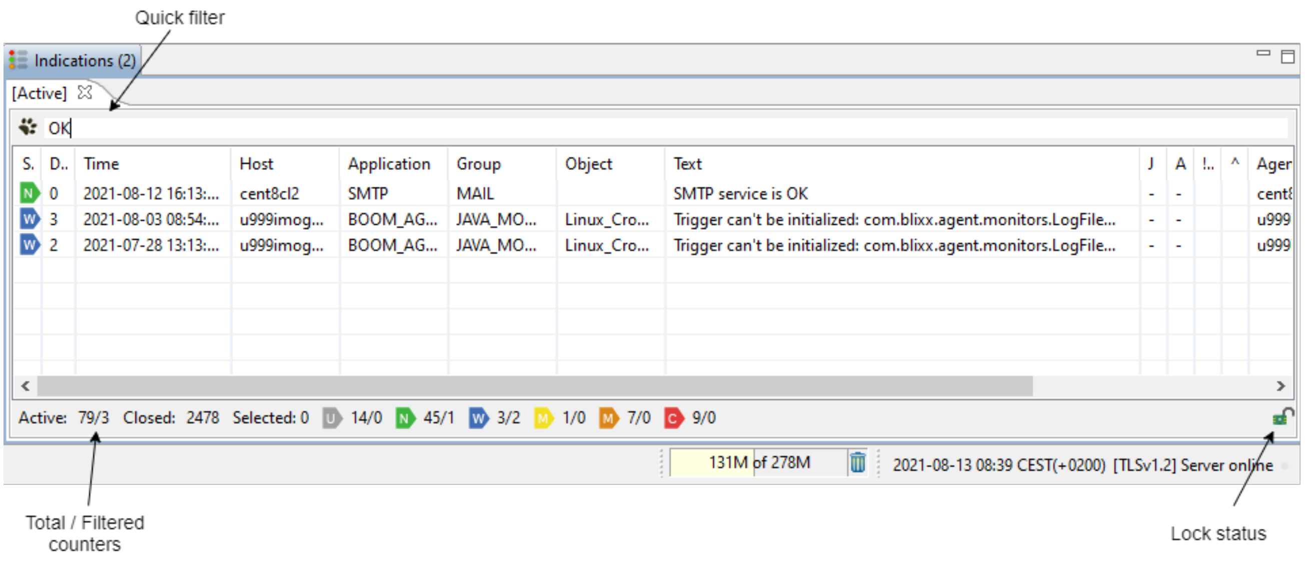
Removes all set filters and reloads all indications.

Show/Hide Columns:

Mark the attributes you want to see in the indication view layout.

6.4.4. Status Line

The indication status line provides the following information:



OK

S.	D..	Time	Host	Application	Group	Object	Text	J	A	!..	^	Ager
N	0	2021-08-12 16:13:...	cent8cl2	SMTP	MAIL		SMTP service is OK	-	-			cent8
W	3	2021-08-03 08:54:...	u999imog...	BOOM_AG...	JAVA_MO...	Linux_Cro...	Trigger can't be initialized: com.blixx.agent.monitors.LogFile...	-	-			u999
W	2	2021-07-28 13:13:...	u999imog...	BOOM_AG...	JAVA_MO...	Linux_Cro...	Trigger can't be initialized: com.blixx.agent.monitors.LogFile...	-	-			u999

```
Active: 79/3  Closed: 2478  Selected: 0  U 14/0  N 45/1  W 3/2  M 1/0  M 7/0  C 9/0
131M of 278M    2021-08-13 08:39 CEST(+0200)  [TLSv1.2] Server online
```

- Number of active indications.
- Number of closed indications.
- Number of selected indications.
- Number of indications belonging to the different severity levels.
- It shows if the indication table is locked.

The first number of each group indicates the total number of all existing indications, the second number is the amount of indications that are currently displayed in the table. The two numbers can differ i.e. if a filter is added to the table. Double-click on a severity icon or on the words 'Active' and 'Closed' will open a new indication table containing all appropriate indications.

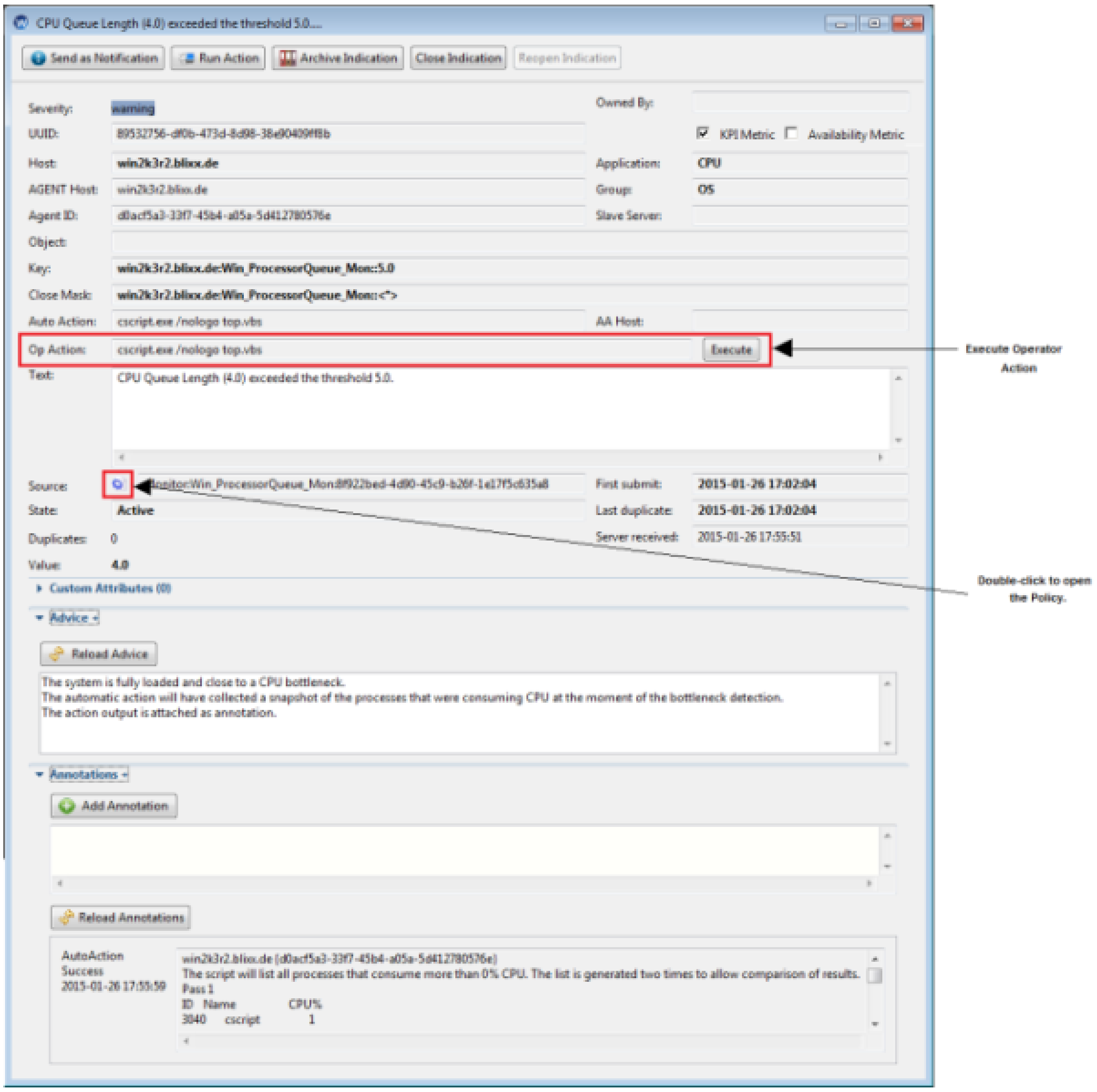
Lock Status Symbols

All 'Indication Tables' are refreshing when updates are coming.

🔒	A locked table will not be refreshed until the lock remains active. There are two situation when a table can be locked: (1) If the table selection exceeds 5000 elements. (2)The user manually locks the table by double-click on the unlocked icon or via the content menu (right click on the unlocked icon). This function can be helpful i.e. if the table is sorted other than by time and you want to keep a certain selection.
🔓	An unlocked table will be refreshed periodically.

6.4.5. Indication Details

Double-click on a single indication or use the 'Open Indication' action from the content menu to open the 'Indication Details' window. The 'Indication Details' window contains all available indication details. Please note: The indication details are not editable (except annotations)! Double-click on the policy icon opens the policy details and selects the condition that triggered the indication.



CPU Queue Length (4.0) exceeded the threshold 5.0....

Severity:	warning	Owned By:	
UUID:	89532756-df0b-473d-8d98-38e90409ff8b	☑ KPI Metric	☐ Availability Metric
Host:	win2k3r2.blixx.de	Application:	CPU
AGENT Host:	win2k3r2.blixx.de	Group:	OS
Agent ID:	d0acf5a3-33f7-45b4-a05a-5d412780576e	Slave Server:	
Object:			
Key:	win2k3r2.blixx.de:Win_ProcessorQueue_Mon::5.0		
Close Mask:	win2k3r2.blixx.de:Win_ProcessorQueue_Mon::<*>		
Auto Action:	cscript.exe /nologo top.vbs	AA Host:	
Op Action:	cscript.exe /nologo top.vbs		
Text:	CPU Queue Length (4.0) exceeded the threshold 5.0.		
Source:	Monitor:Win_ProcessorQueue_Mon:8f922bed-4d80-45c9-b26f-1a17f5c635a8	First submit:	2015-01-26 17:02:04
State:	Active	Last duplicate:	2015-01-26 17:02:04
Duplicates:	0	Server received:	2015-01-26 17:55:51
Value:	4.0		

The system is fully loaded and close to a CPU bottleneck.
The automatic action will have collected a snapshot of the processes that were consuming CPU at the moment of the bottleneck detection.
The action output is attached as annotation.

AutoAction
Success
2015-01-26 17:55:59

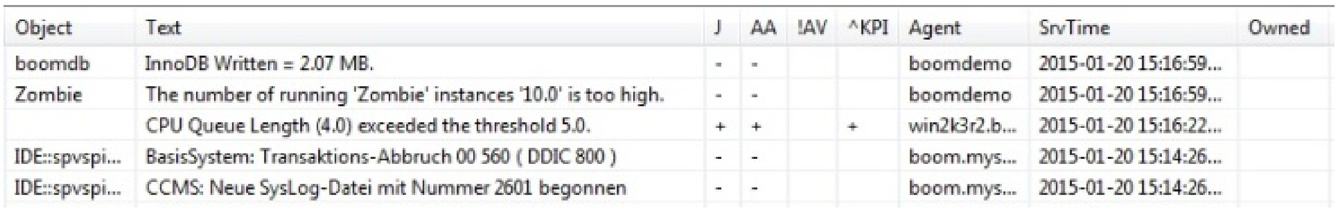
```
win2k3r2.blixx.de (d0acf5a3-33f7-45b4-a05a-5d412780576e)
The script will list all processes that consume more than 0% CPU. The list is generated two times to allow comparison of results.
Pass 1
ID   Name        CPU%
3040    cscript        1
```

Severity:	Severity of the current indication
UUID:	Unique number to identify the indication
Host:	Node from where the indication has been issued
Agent Host:	Host name of the *boom* agent
Agent ID:	Unique number to identify the *boom* agent
Object:	Specific object that detected or caused the indication
Key:	A unique key which differentiates an indication from other indications. The indication key contains a list of supported attributes/variables that will be resolved during the processing of the incoming value. The result string is used for the correlation with the following indication for the auto acknowledgment and the duplicate suppression.
Close Mask:	Pattern that is used to search and close previously submitted indications

Application:	Name of the application that detected the problem
Group:	Indication group the message belongs to
Slave Server:	Name of the *boom* slave server
Auto Action:	Specifies an action that will automatically be performed on an incoming indication on the configured node. An automatic action will be triggered from the *boom* server. The result of this action will be stored as annotation in the 'Annotation Field' of the indication.
Op Action:	Recommended operator action. The operator action allows you to trigger a specified action manually. Press "execute" to trigger the action.
Text:	Detailed description of the problem
Source:	Policy condition that triggered the indication. Double-click on the policy icon opens the policy details windows and selects the condition that triggered the indication.
State:	Displays the state of the indication e.g. active, closed by operator, auto close
Duplicates:	The *boom* server detects duplicates if one of the De-Duplication flags is set in the policy. If duplicate indications are received the *boom* server increases the "duplicate count" and updates the "last duplicate" time of the first received indication.
Value:	Monitored value
First submit:	Creation time of the indication
Last duplicate:	If duplicate indications are received the *boom* server increases the "duplicate count" and updates the "last duplicate" time of the first received indication.
Server Received:	Date and time the message received on the management server
Custom Attributes (x):	x shows the number of specified customized attributes. Displays all specified customized attributes.
Advice:	+ sign indicates that an advice is available Provides instructions for the operator.
Annotation:	+ sign indicates that annotations are available Annotations are stored till the indication is deleted from the archive. Annotations are added by an automatic action. The result and output will be added to the annotation field of the indication. Annotations are added via the re-open function. Annotations can be added by the operator. Annotations are added for indications which are closed via message correlation (Auto closed indications). The closing date, closing time and the ID of the closing alarm is added. Select 'Add Annotation' to add a new annotation.

Depending on the setup of the indication view the following information is shown:

Annotations	are marked in one the columns **(J)** of the indication view window.
Automatic actions	are marked in one column **(AA)** of the indication view window.
Operator actions	are **not** marked in the indication view window.

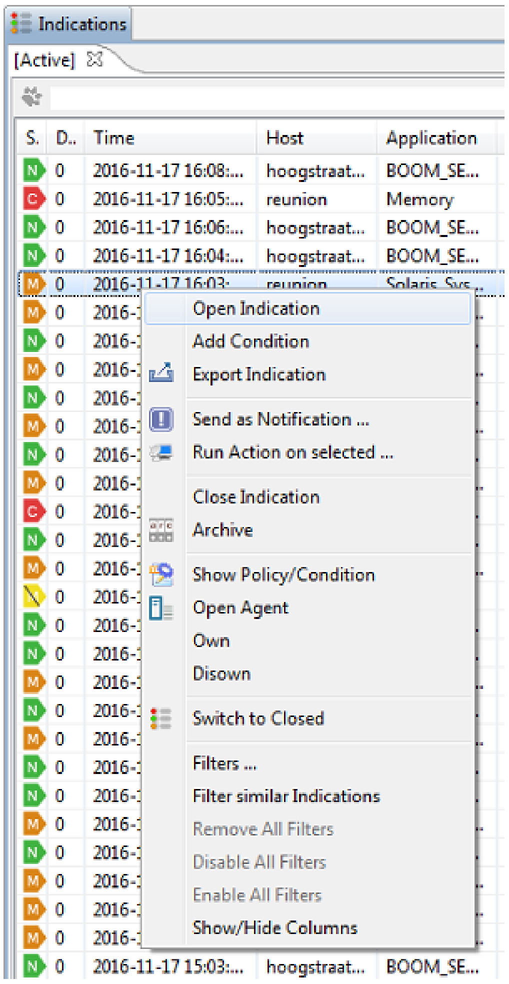


Object	Text	J	AA	!AV	^KPI	Agent	SrvTime	Owned
boomdb	InnoDB Written = 2.07 MB.	-	-			boomdemo	2015-01-20 15:16:59...	
Zombie	The number of running 'Zombie' instances '10.0' is too high.	-	-			boomdemo	2015-01-20 15:16:59...	
	CPU Queue Length (4.0) exceeded the threshold 5.0.	+	+		+	win2k3r2.b...	2015-01-20 15:16:22...	
IDE::spvspi...	BasisSystem: Transaktions-Abbruch 00 560 (DDIC 800)	-	-			boom.mys...	2015-01-20 15:14:26...	
IDE::spvspi...	CCMS: Neue SysLog-Datei mit Nummer 2601 begonnen	-	-			boom.mys...	2015-01-20 15:14:26...	

6.4.6. Indication Filtering

All incoming indications are displayed in the 'Indication Browser'. The 'Indication Filtering' gives you a possibility to display only the relevant indications by adding one or more filters to the table. Please note: The filters are not automatically saved when closing the *boom* workbench! If you want to reuse the filters you have to add them to the 'Filter History' (see below).

Right-click on any Indication to open the Indication context menu:



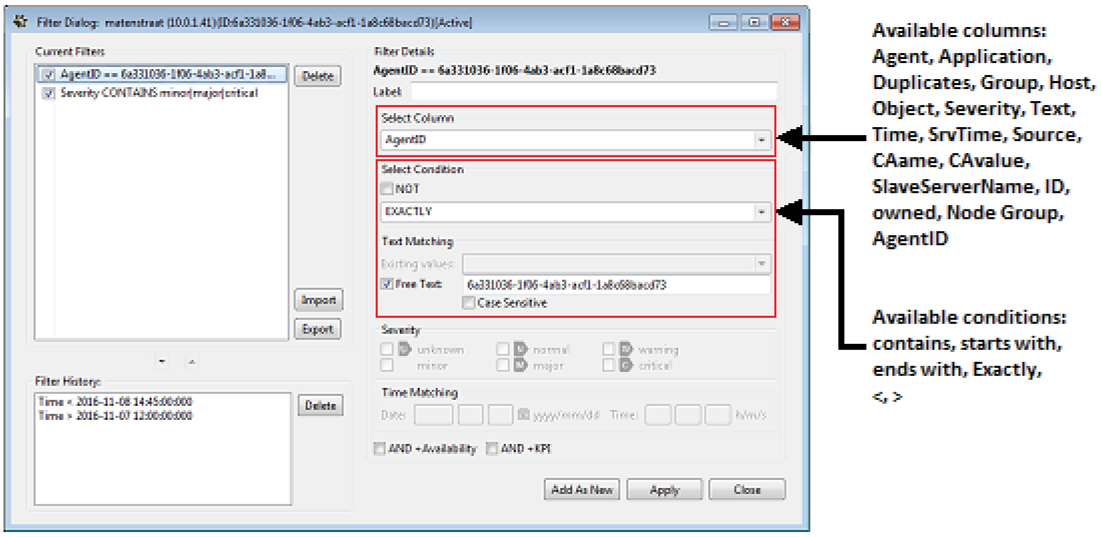
S.	D..	Time	Host	Application
N	0	2016-11-17 16:08:...	hoogstraat...	BOOM_SE...
C	0	2016-11-17 16:05:...	reunion	Memory
N	0	2016-11-17 16:06:...	hoogstraat...	BOOM_SE...
N	0	2016-11-17 16:04:...	hoogstraat...	BOOM_SE...
M	0	2016-11-17 16:03:...	reunion	Solaris Sys...
N	0	2016-11-17 15:03:...	hoogstraat...	BOOM_SE...

Filters :	Click on 'Filters ...' will open the 'Filter Dialog'. All filters that have been added to an indication table are listed in the context menu. The check mark in front of a filter indicates an active filter. No check mark means the filter is not active.

All filters that have been added to an Indication table are listed in the context menu.

✓	The check mark in front of a filter indicates an active filter.
	No check mark means the filter is not active

To add, delete or change an existing filter you have to select 'Filters ...' from the context menu above. The following dialog will be opened:



Filter Dialog: matenstraat (10.0.1.41)(ID:6a331036-1f06-4ab3-acf1-1a8c68bacd73)(Active)

Current Filters:
- [✓] AgentID == 6a331036-1f06-4ab3-acf1-1a8...
- [✓] Severity CONTAINS minor|major|critical

Filter History:
- Time < 2016-11-08 14:45:00:000
- Time > 2016-11-07 12:00:00:000

Filter Details
AgentID == 6a331036-1f06-4ab3-acf1-1a8c68bacd73
Label:
Select Column: AgentID
Select Condition: NOT [ ] EXACTLY
Text Matching: Free Text [✓]: 6a331036-1f06-4ab3-acf1-1a8c68bacd73; Case Sensitive [ ]

Available columns:
Agent, Application, Duplicates, Group, Host, Object, Severity, Text, Time, SrvTime, Source, CAame, CAvalue, SlaveServerName, ID, owned, Node Group, AgentID

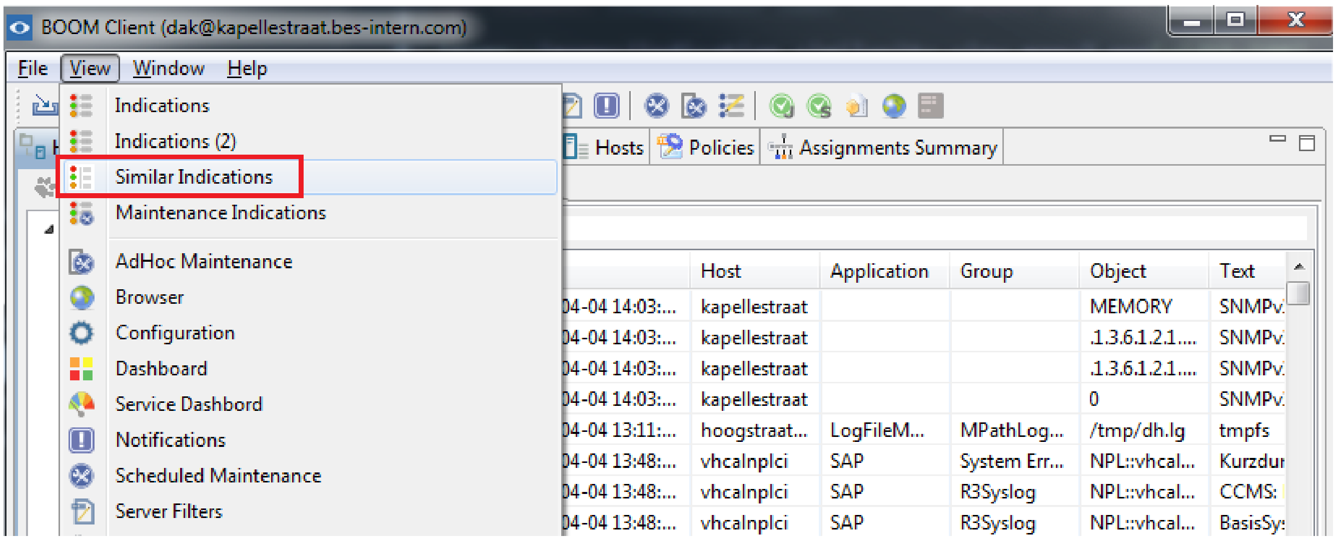
Available conditions: contains, starts with, ends with, Exactly, <, >

Activate/Deactivate Filter:	If a filter is not needed it can be simply deactivated. The filter exists until it will be deleted or the workbench will be closed. To save a filter permanently you must add the filter to the 'Filter History'.
Import:	Import filters from an XML file.
Export:	Export filters to an XML file.
Filter History:	All filters that are moved to the 'Filter History' will be saved automatically to an XML file (filterHistory.xml). The filters are available after a restart.

ℹ	Any changes that are made inside the 'Filter Dialog' will automatically reflect on the 'Indication Browser' without closing the dialog.

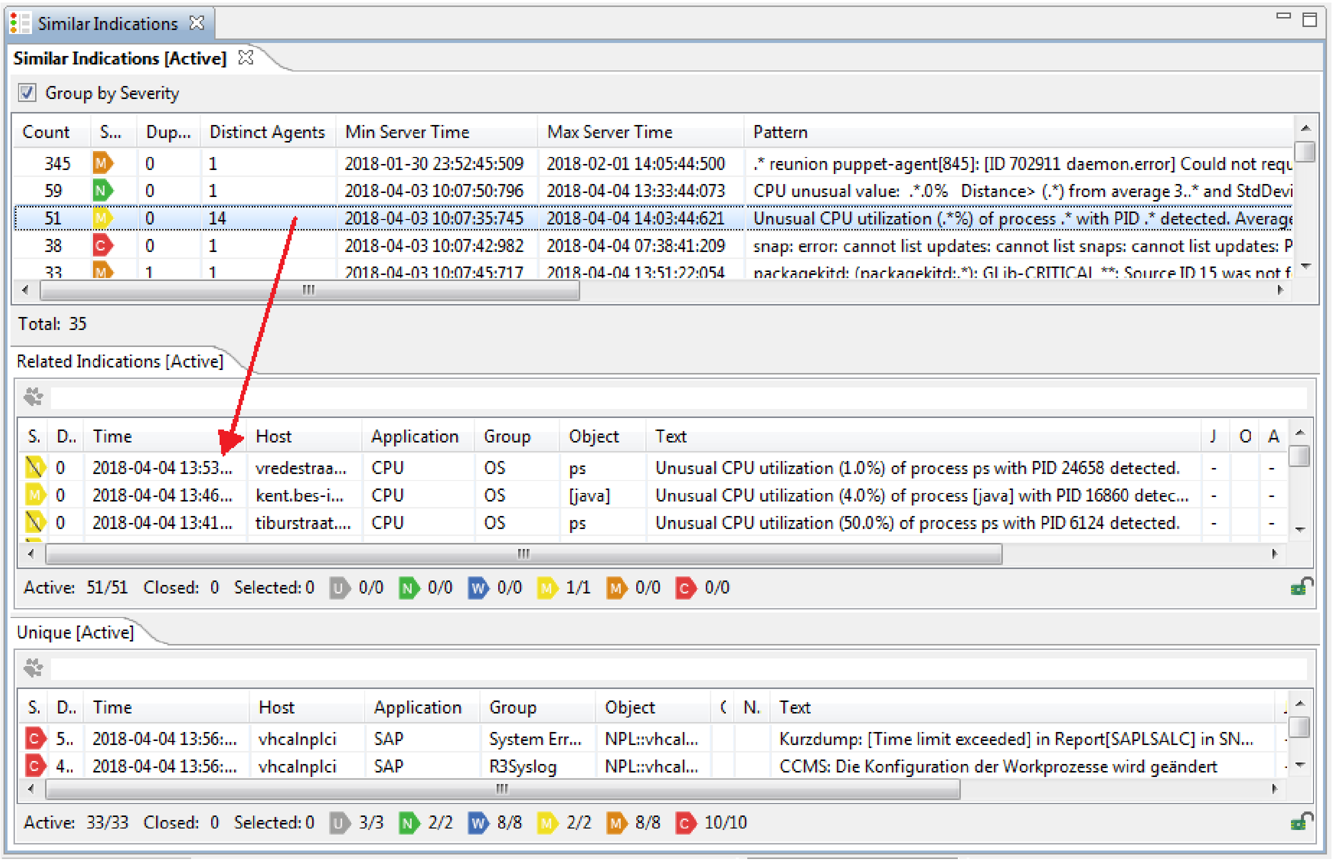
6.4.7. Indication Similarity View

The Similar Indications View can be opened via the View Menu by choosing **Similar Indications**.



	Host	Application	Group	Object	Text
04-04 14:03:...	kapellestraat			MEMORY	SNMPv.
04-04 14:03:...	kapellestraat			.1.3.6.1.2.1....	SNMPv.
04-04 14:03:...	kapellestraat			.1.3.6.1.2.1....	SNMPv.
04-04 14:03:...	kapellestraat			0	SNMPv.
04-04 13:11:...	hoogstraat...	LogFileM...	MPathLog...	/tmp/dh.lg	tmpfs
04-04 13:48:...	vhcalnplci	SAP	System Err...	NPL::vhcal...	Kurzdur
04-04 13:48:...	vhcalnplci	SAP	R3Syslog	NPL::vhcal...	CCMS:
04-04 13:48:...	vhcalnplci	SAP	R3Syslog	NPL::vhcal...	BasisSy:

The Similar Indications View groups indications by their similarity regarding their attributes enabling to quickly identify common problem patterns or related indications.



Count	S...	Dup...	Distinct Agents	Min Server Time	Max Server Time	Pattern
345	M	0	1	2018-01-30 23:52:45:509	2018-02-01 14:05:44:500	.* reunion puppet-agent[845]: [ID 702911 daemon.error] Could not requ
59	N	0	1	2018-04-03 10:07:50:796	2018-04-04 13:33:44:073	CPU unusual value: .*.0% Distance> (.*) from average 3..* and StdDevi
51	M	0	14	2018-04-03 10:07:35:745	2018-04-04 14:03:44:621	Unusual CPU utilization (.*%) of process .* with PID .* detected. Average
38	C	0	1	2018-04-03 10:07:42:982	2018-04-04 07:38:41:209	snap: error: cannot list updates: cannot list snaps: cannot list updates: P
33	M	1	1	2018-04-03 10:07:45:717	2018-04-04 13:51:22:054	packagekitd: (packagekitd:.*): GLib-CRITICAL **: Source ID 15 was not f

S.	D..	Time	Host	Application	Group	Object	Text	J	O	A
	0	2018-04-04 13:53...	vredestraa...	CPU	OS	ps	Unusual CPU utilization (1.0%) of process ps with PID 24658 detected.	-		-
M	0	2018-04-04 13:46...	kent.bes-i...	CPU	OS	[java]	Unusual CPU utilization (4.0%) of process [java] with PID 16860 detec...	-		-
	0	2018-04-04 13:41...	tiburstraat....	CPU	OS	ps	Unusual CPU utilization (50.0%) of process ps with PID 6124 detected.	-		-

Active: 51/51 Closed: 0 Selected: 0 U 0/0 N 0/0 W 0/0 M 1/1 M 0/0 C 0/0

S.	D..	Time	Host	Application	Group	Object	(	N.	Text
C	5..	2018-04-04 13:56:...	vhcalnplci	SAP	System Err...	NPL::vhcal...			Kurzdump: [Time limit exceeded] in Report[SAPLSALC] in SN...
C	4..	2018-04-04 13:56:...	vhcalnplci	SAP	R3Syslog	NPL::vhcal...			CCMS: Die Konfiguration der Workprozesse wird geändert

Active: 33/33 Closed: 0 Selected: 0 U 3/3 N 2/2 W 8/8 M 2/2 M 8/8 C 10/10

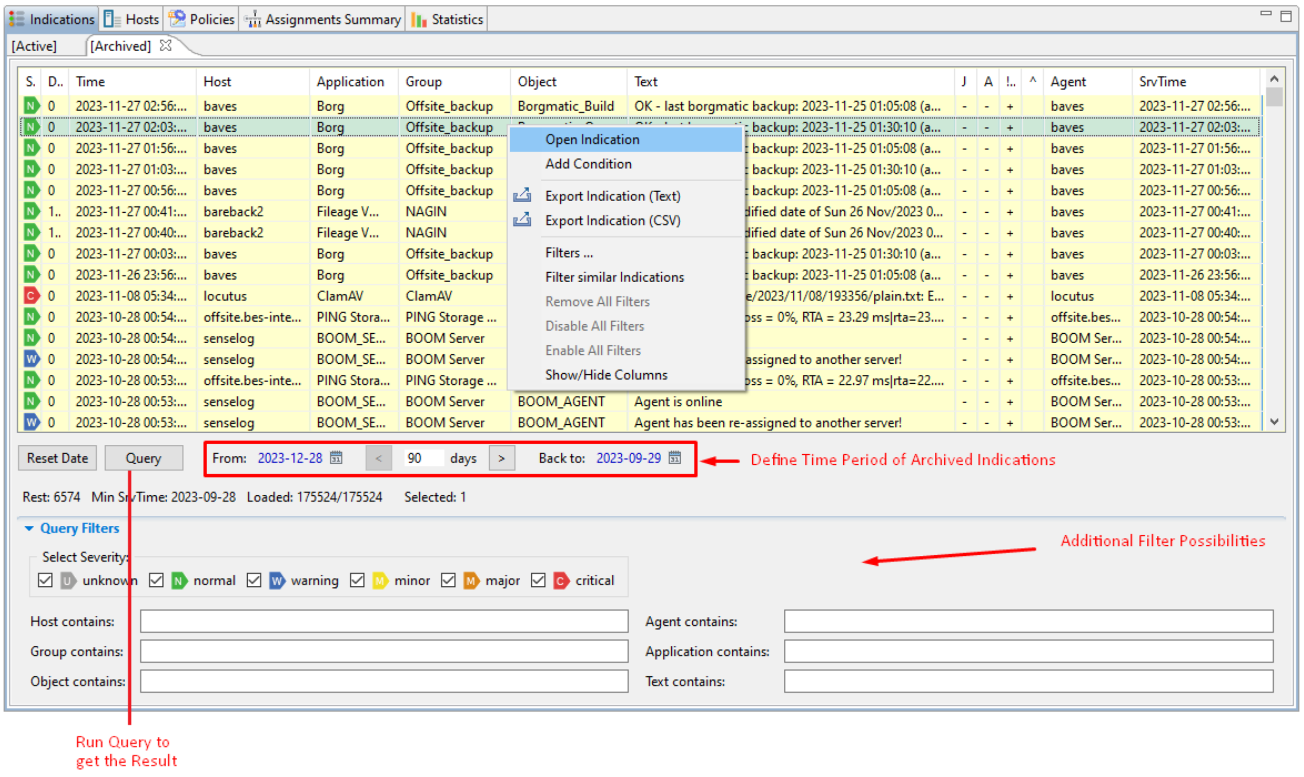
The upper list shows grouped indications with the total amount of related indications and number of originating agents. Selecting an indication in this list will display all related indications in the middle section named "Filtered Groups".
The lower list shows all indications that did not have enough similarity to another active indication and are therefore considered to be a unique indications.

6.4.8. Archived Indications

The *boom* workbench provides three indication views. By default the first view contains all active indications [Active] and the second view all closed indications [Closed]. The third view shows all archived indications and isn't shown by default. Select "View Archived" in the indication context menu to open the archived view.

Only "closed indications" can be archived.

Archived Indication View:



Indications | Hosts | Policies | Assignments Summary | Statistics

[Active] | [Archived]

S.	D..	Time	Host	Application	Group	Object	Text	J	A	!..	^	Agent	SrvTime
N	0	2023-11-27 02:56:...	baves	Borg	Offsite_backup	Borgmatic_Build	OK - last borgmatic backup: 2023-11-25 01:05:08 (a...	-	-	+		baves	2023-11-27 02:56:...
N	0	2023-11-27 02:03:...	baves	Borg	Offsite_backup		backup: 2023-11-25 01:30:10 (a...	-	-	+		baves	2023-11-27 02:03:...
N	0	2023-11-27 01:56:...	baves	Borg	Offsite_backup		backup: 2023-11-25 01:05:08 (a...	-	-	+		baves	2023-11-27 01:56:...
N	0	2023-11-27 01:03:...	baves	Borg	Offsite_backup		backup: 2023-11-25 01:30:10 (a...	-	-	+		baves	2023-11-27 01:03:...
N	0	2023-11-27 00:56:...	baves	Borg	Offsite_backup		backup: 2023-11-25 01:05:08 (a...	-	-	+		baves	2023-11-27 00:56:...
N	1..	2023-11-27 00:41:...	bareback2	Fileage V...	NAGIN		dified date of Sun 26 Nov/2023 0...	-	-	+		baves	2023-11-27 00:41:...
N	1..	2023-11-27 00:40:...	bareback2	Fileage V...	NAGIN		dified date of Sun 26 Nov/2023 0...	-	-	+		baves	2023-11-27 00:40:...
N	0	2023-11-27 00:03:...	baves	Borg	Offsite_backup		backup: 2023-11-25 01:30:10 (a...	-	-	+		baves	2023-11-27 00:03:...
N	0	2023-11-26 23:56:...	baves	Borg	Offsite_backup		backup: 2023-11-25 01:05:08 (a...	-	-	+		baves	2023-11-26 23:56:...
C	0	2023-11-08 05:34:...	locutus	ClamAV	ClamAV		e/2023/11/08/193356/plain.txt: E...	-	-	+		locutus	2023-11-08 05:34:...
N	0	2023-10-28 00:54:...	offsite.bes-inte...	PING Stora...	PING Storage ...		oss = 0%, RTA = 23.29 ms|rta=23....	-	-	+		offsite.bes...	2023-10-28 00:54:...
N	0	2023-10-28 00:54:...	senselog	BOOM_SE...	BOOM Server			-	-	+		BOOM Ser...	2023-10-28 00:54:...
W	0	2023-10-28 00:54:...	senselog	BOOM_SE...	BOOM Server		assigned to another server!	-	-	+		BOOM Ser...	2023-10-28 00:54:...
N	0	2023-10-28 00:53:...	offsite.bes-inte...	PING Stora...	PING Storage ...		oss = 0%, RTA = 22.97 ms|rta=22....	-	-	+		offsite.bes...	2023-10-28 00:53:...
N	0	2023-10-28 00:53:...	senselog	BOOM_SE...	BOOM Server	BOOM_AGENT	Agent is online	-	-	+		BOOM Ser...	2023-10-28 00:53:...
W	0	2023-10-28 00:53:...	senselog	BOOM_SE...	BOOM Server	BOOM_AGENT	Agent has been re-assigned to another server!	-	-	+		BOOM Ser...	2023-10-28 00:53:...

From: 2023-12-28 < 90 days > Back to: 2023-09-29

Rest: 6574 Min SrvTime: 2023-09-28 Loaded: 175524/175524 Selected: 1

Query Filters

Select Severity:
☑ unknown ☑ normal ☑ warning ☑ minor ☑ major ☑ critical

Host contains:
Group contains:
Object contains:
Agent contains:
Application contains:
Text contains:

Time frame and Query:

Specify a time frame to view archived indications.

Use the **From**:
Field to define a start date for the time filter. To select the start date use the calendar.

Use the **Back to**:
Field to define a prior finish date for the time filter. To select the prior finish date use the calendar.
e.g. from: 2012-01-15 Back to: 2011-10-26 (prior to the start date in the from: field)

Use the **Query filters** section for a more detailed search.

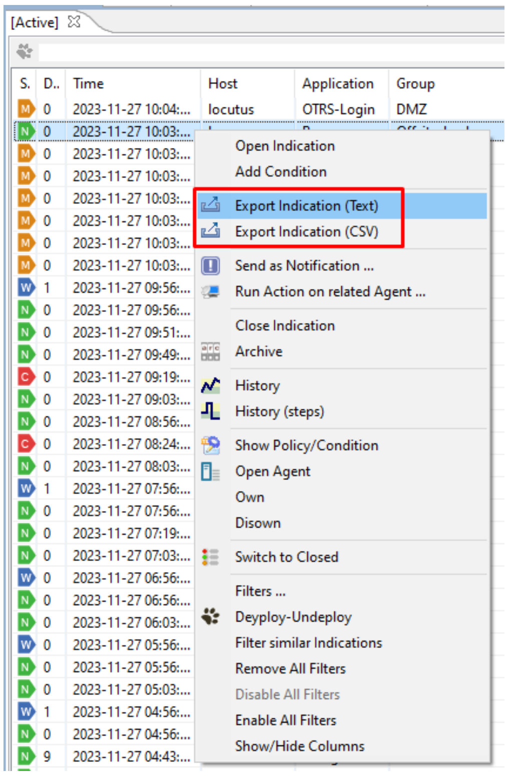
Please be aware that the search can't be interrupted. It is recommended to use an additional filter criteria. * is not supported as wildcard search. Just use the characters which are part of the filter attribute. e.g. TEST for TESTNODE

There are two parameters in the *boom* server config file boom.props to handle closed and archived indications:

- **AUTO_ARCHIVE_DAYS:** closed indications are automatically archived after x days
- **AUTO_DELETE_DAYS:** archived indications are automatically deleted after x days

6.4.9. Indication Export

One or more active/closed indications can be exported to a simple TEXT file or a comma-separated CSV file.

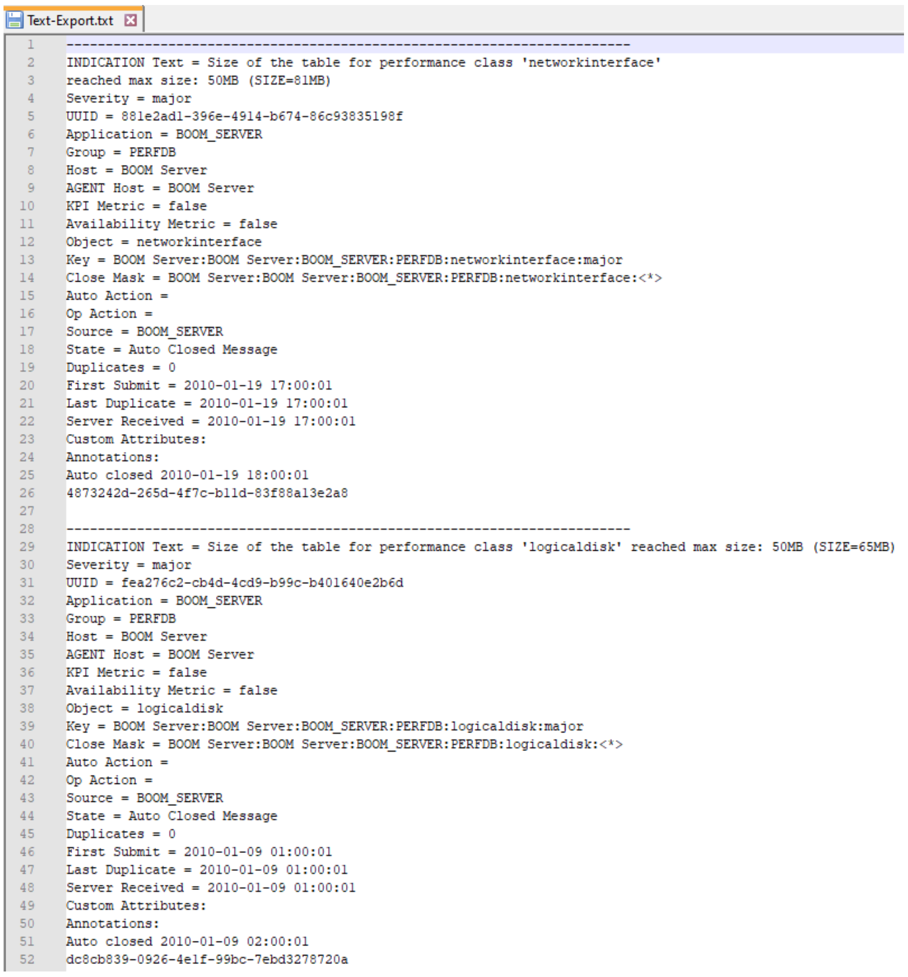


S.	D..	Time	Host	Application	Group
M	0	2023-11-27 10:04:...	locutus	OTRS-Login	DMZ
N	0	2023-11-27 10:03:...			
M	0	2023-11-27 10:03:...			
M	0	2023-11-27 10:03:...			
M	0	2023-11-27 10:03:...			
M	0	2023-11-27 10:03:...			
M	0	2023-11-27 10:03:...			
M	0	2023-11-27 10:03:...			
W	1	2023-11-27 09:56:...			
N	0	2023-11-27 09:56:...			
N	0	2023-11-27 09:51:...			
N	0	2023-11-27 09:49:...			
C	0	2023-11-27 09:19:...			
N	0	2023-11-27 09:03:...			
N	0	2023-11-27 08:56:...			
C	0	2023-11-27 08:24:...			
N	0	2023-11-27 08:03:...			
W	1	2023-11-27 07:56:...			
N	0	2023-11-27 07:56:...			
N	0	2023-11-27 07:19:...			
N	0	2023-11-27 07:03:...			
W	0	2023-11-27 06:56:...			
N	0	2023-11-27 06:56:...			
N	0	2023-11-27 06:03:...			
W	0	2023-11-27 05:56:...			
N	0	2023-11-27 05:56:...			
N	0	2023-11-27 05:03:...			
W	1	2023-11-27 04:56:...			
N	0	2023-11-27 04:56:...			
N	9	2023-11-27 04:43:...			

Right-click on any **open**, **closed** or **archived** indication you want to export. Select "Export Indication (Text)" respectively "Export Indication (CSV)".

6.4.9.1. Indication Export (Text)

Sample Output:

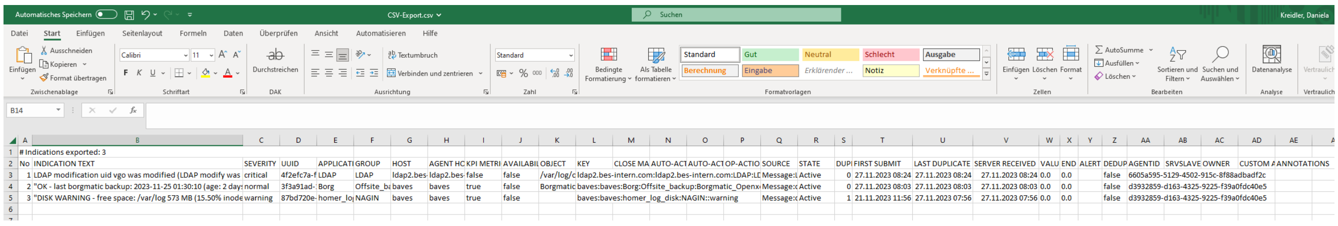


```
Text-Export.txt
------------------------------------------------------------------------
INDICATION Text = Size of the table for performance class 'networkinterface'
reached max size: 50MB (SIZE=81MB)
Severity = major
UUID = 881e2ad1-396e-4914-b674-86c93835198f
Application = BOOM_SERVER
Group = PERFDB
Host = BOOM Server
AGENT Host = BOOM Server
KPI Metric = false
Availability Metric = false
Object = networkinterface
Key = BOOM Server:BOOM Server:BOOM_SERVER:PERFDB:networkinterface:major
Close Mask = BOOM Server:BOOM Server:BOOM_SERVER:PERFDB:networkinterface:<*>
Auto Action =
Op Action =
Source = BOOM_SERVER
State = Auto Closed Message
Duplicates = 0
First Submit = 2010-01-19 17:00:01
Last Duplicate = 2010-01-19 17:00:01
Server Received = 2010-01-19 17:00:01
Custom Attributes:
Annotations:
Auto closed 2010-01-19 18:00:01
4873242d-265d-4f7c-b11d-83f88a13e2a8

------------------------------------------------------------------------
INDICATION Text = Size of the table for performance class 'logicaldisk' reached max size: 50MB (SIZE=65MB)
Severity = major
UUID = fea276c2-cb4d-4cd9-b99c-b401640e2b6d
Application = BOOM_SERVER
Group = PERFDB
Host = BOOM Server
AGENT Host = BOOM Server
KPI Metric = false
Availability Metric = false
Object = logicaldisk
Key = BOOM Server:BOOM Server:BOOM_SERVER:PERFDB:logicaldisk:major
Close Mask = BOOM Server:BOOM Server:BOOM_SERVER:PERFDB:logicaldisk:<*>
Auto Action =
Op Action =
Source = BOOM_SERVER
State = Auto Closed Message
Duplicates = 0
First Submit = 2010-01-09 01:00:01
Last Duplicate = 2010-01-09 01:00:01
Server Received = 2010-01-09 01:00:01
Custom Attributes:
Annotations:
Auto closed 2010-01-09 02:00:01
dc8cb839-0926-4e1f-99bc-7ebd3278720a
```

6.4.9.2. Indication Export (CSV)

"Export Indication (CSV)" utilizes ";" internally as separator suitable for opening CSV file by Windows "Excel".

Sample Output:



CSV-Export.csv

Indications exported: 3

No	INDICATION TEXT	SEVERITY	UUID	APPLICATI	GROUP	HOST	AGENT HC	KPI METRI	AVAILABIL	OBJECT	KEY	CLOSE MA	AUTO-ACT	AUTO-ACT	OP-ACTIO	SOURCE	STATE	DUPI	FIRST SUBMIT	LAST DUPLICATE	SERVER RECEIVED	VALU	END	ALERT	DEDUP	AGENTID	SRVSLAVE	OWNER	CUSTOM A	ANNOTATIONS
1	LDAP modification uid vgo was modified (LDAP modify was	critical	4f2efc7a-f	LDAP	LDAP	ldap2.bes-	ldap2.bes-	false	false	/var/log/c	ldap2.bes-intern.com:ldap2.bes-intern.com:LDAP:LD					Message:l	Active	0	27.11.2023 08:24	27.11.2023 08:24	27.11.2023 08:24	0.0	0.0		false	6605a595-5129-4502-915c-8f88adbadf2c				
2	"OK - last borgmatic backup: 2023-11-25 01:30:10 (age: 2 days	normal	3f3a91ad-1	Borg	Offsite_ba	baves	baves	true	false	Borgmatic	baves:baves:Borg:Offsite_backup:Borgmatic_Openx					Message:c	Active	0	27.11.2023 08:03	27.11.2023 08:03	27.11.2023 08:03	0.0	0.0		false	d3932859-d163-4325-9225-f39a0fdc40e5				
3	"DISK WARNING - free space: /var/log 573 MB (15.50% inode	warning	87bd720e-	homer_lo	NAGIN	baves	baves	true	false		baves:baves:homer_log_disk:NAGIN::warning					Message:c	Active	1	21.11.2023 11:56	27.11.2023 07:56	27.11.2023 07:56	0.0	0.0		false	d3932859-d163-4325-9225-f39a0fdc40e5				

```
# Indications exported: 1
No;INDICATION TEXT;SEVERITY;UUID;APPLICATION;GROUP;HOST;AGENT HOST;KPI METRIC;AVAILABILITY
METRIC;OBJECT;KEY;CLOSE MASK;AUTO-ACTION;AUTO-ACTION HOST;OP-ACTION;SOURCE;STATE;DUPLICATES;FIRST
SUBMIT;LAST DUPLICATE;SERVER RECEIVED;VALUE;END VALUE;ALERT FINISHED;DEDUPLICATION
KEYONLY;AGENTID;SRVSLAVE;OWNER;CUSTOM ATTRIBUTES;ANNOTATIONS
1;"""DISK WARNING - free space: /var/log 620 MB (16.77% inode=96%);|
/var/log=3078MB;3122;3512;0;3903""";warning;719aff37-9480-4032-a786-
11e5cc2b3fda;homer_log_disk;NAGIN;baves;baves;true;false;;baves:baves:homer_log_disk:NAGIN::warning;;;;;Me
ssage:check_nrpe_homer_log_disk:db4b1f36-50e0-4dda-b657-4ed1857f5423;Active;0;27.11.2023 05:56;27.11.2023
05:56;27.11.2023 05:56;0.0;0.0;;false;d3932859-d163-4325-9225-f39a0fdc40e5;;;;

# Indications exported: 2
No;INDICATION TEXT;SEVERITY;UUID;APPLICATION;GROUP;HOST;AGENT HOST;KPI METRIC;AVAILABILITY
METRIC;OBJECT;KEY;CLOSE MASK;AUTO-ACTION;AUTO-ACTION HOST;OP-ACTION;SOURCE;STATE;DUPLICATES;FIRST
SUBMIT;LAST DUPLICATE;SERVER RECEIVED;VALUE;END VALUE;ALERT FINISHED;DEDUPLICATION
KEYONLY;AGENTID;SRVSLAVE;OWNER;CUSTOM ATTRIBUTES;ANNOTATIONS
1;LDAP modification uid vgo was modified (LDAP modify was performed) # modify 1701069812 dc=bes-
intern,dc=com cn=admin,dc=bes-intern,dc=com IP=10.0.0.139:35662 conn=5457736 dn: uid=vgo,ou=user,dc=bes-
intern,dc=com changetype: modify replace: sambaNTPassword sambaNTPassword:
87186CBC3309488EBF2C55C64642F7FA - replace: sambaPwdLastSet sambaPwdLastSet: 1701069811 - replace:
userPassword userPassword:: e1NTSEF9WTlDczJ6SVRNNk9GVXFhQ2lKQURHcmxBeDdsM0c0cUI= - replace: pwdChangedTime
pwdChangedTime: 20231127072331Z - delete: pwdHistory pwdHistory:
20221018053754Z#1.3.6.1.4.1.1466.115.121.1.40#38#{SSHA}5gSWbJOB9  PLLwMJPNN0XYcG04ZNkCENF - add:
pwdHistory pwdHistory: 20231127072331Z#1.3.6.1.4.1.1466.115.121.1.40#38#{SSHA}hFumXycMS
L0x9P6RbYKnqkN10YBFOtmz - replace: entryCSN entryCSN: 20231127072331.997953Z#000000#000#000000 - replace:
modifiersName modifiersName: cn=admin,dc=bes-intern,dc=com - replace: modifyTimestamp modifyTimestamp:
20231127072331Z -;critical;4f2efc7a-ff37-4bd6-8ce9-e0e43076552c;LDAP;LDAP;ldap2.bes-intern.com;ldap2.bes
-intern.com;false;false;/var/log/openldap/auditlog.ldif;ldap2.bes-intern.com:ldap2.bes
-intern.com:LDAP:LDAP:/var/log/openldap/auditlog.ldif:critical;;;;;Message:LDAPauditreport:3ec90404-ce82
-4e12-b969-862cd76fd4c3;Active;0;2023-11-27 08:24:18;2023-11-27 08:24:18;2023-11-27
08:24:19;0.0;0.0;;false;6605a595-5129-4502-915c-8f88adbadf2c;;;;;
2;"""OK - last borgmatic backup: 2023-11-25 01:30:10 (age: 2 days, 6:33:23.562094) with name offsite-2023-
11-25T01:30:08.250932 | 'lastbackup_s'=196404""";normal;3f3a91ad-1e43-4987-9ef3-
1cea30e9d71e;Borg;Offsite_backup;baves;baves;true;false;Borgmatic_Openxen;baves:baves:Borg:Offsite_backup:
Borgmatic_Openxen:normal;;;;;Message:check_nrpe_borg_on_offsite_openxen_bck:9e288601-3d83-456e-bee9-
4108867ea1d5;Active;0;2023-11-27 08:03:33;2023-11-27 08:03:33;2023-11-27 08:03:43;0.0;0.0;;false;d3932859-
d163-4325-9225-f39a0fdc40e5;;;;;
```

6.5. Policy Management

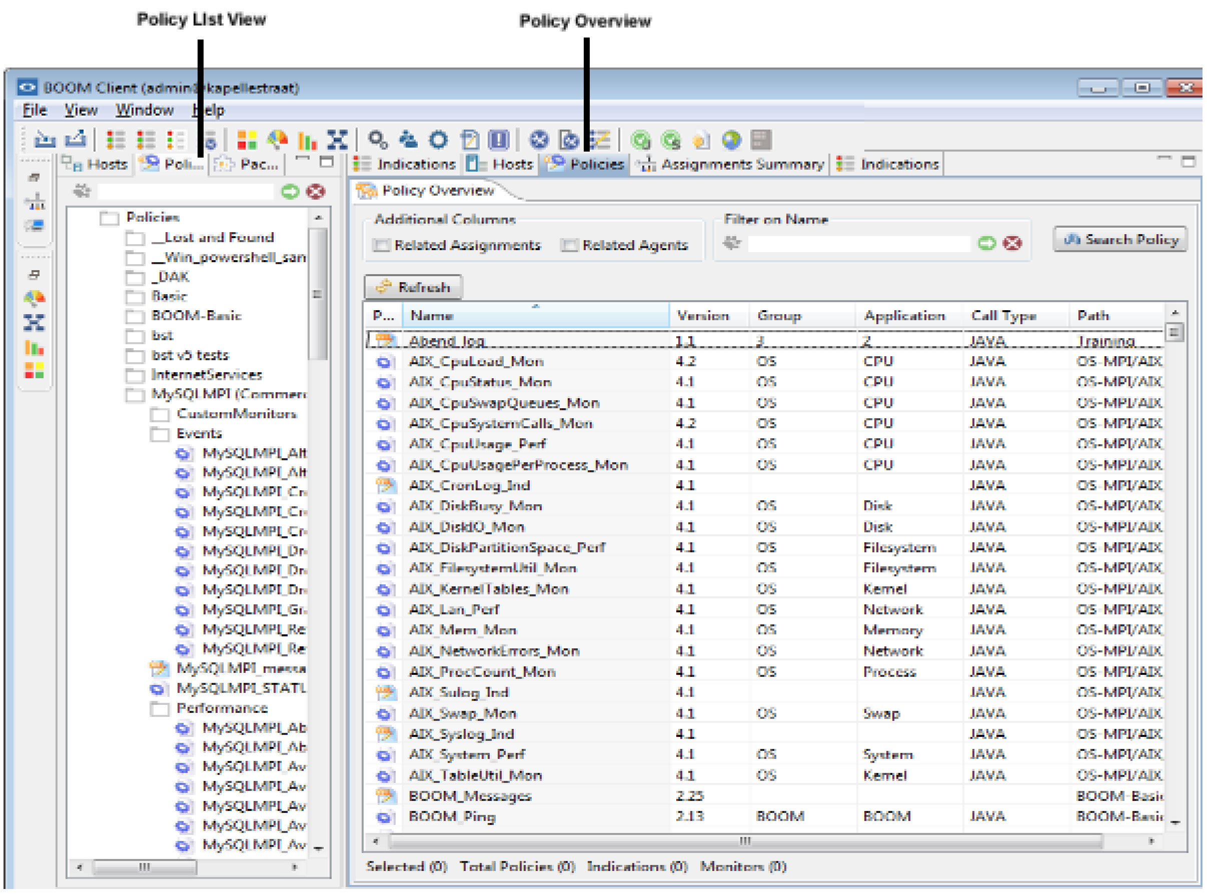
6.5.1. General Information

Two major concepts of the *boom* infrastructure are managed inside the policy management:

- Monitors
- Indication Policies

For more details about policy types see chapters Monitor Policy and Indication Policy.

The *boom* GUI workbench contains two views that are belonging to the policy management. The view on the left gives an overview of all existing policies and the view on the right provides all available policy details.



P...	Name	Version	Group	Application	Call Type	Path
	Abend_log	1.1	3	2	JAVA	Training
	AIX_CpuLoad_Mon	4.2	OS	CPU	JAVA	OS-MPI/AIX
	AIX_CpuStatus_Mon	4.1	OS	CPU	JAVA	OS-MPI/AIX
	AIX_CpuSwapQueues_Mon	4.1	OS	CPU	JAVA	OS-MPI/AIX
	AIX_CpuSystemCalls_Mon	4.2	OS	CPU	JAVA	OS-MPI/AIX
	AIX_CpuUsage_Perf	4.1	OS	CPU	JAVA	OS-MPI/AIX
	AIX_CpuUsagePerProcess_Mon	4.1	OS	CPU	JAVA	OS-MPI/AIX
	AIX_CronLog_Ind	4.1			JAVA	OS-MPI/AIX
	AIX_DiskBusy_Mon	4.1	OS	Disk	JAVA	OS-MPI/AIX
	AIX_DiskIO_Mon	4.1	OS	Disk	JAVA	OS-MPI/AIX
	AIX_DiskPartitionSpace_Perf	4.1	OS	Filesystem	JAVA	OS-MPI/AIX
	AIX_FilesystemUtil_Mon	4.1	OS	Filesystem	JAVA	OS-MPI/AIX
	AIX_KernelTables_Mon	4.1	OS	Kernel	JAVA	OS-MPI/AIX
	AIX_Lan_Perf	4.1	OS	Network	JAVA	OS-MPI/AIX
	AIX_Mem_Mon	4.1	OS	Memory	JAVA	OS-MPI/AIX
	AIX_NetworkErrors_Mon	4.1	OS	Network	JAVA	OS-MPI/AIX
	AIX_ProcCount_Mon	4.1	OS	Process	JAVA	OS-MPI/AIX
	AIX_Sulog_Ind	4.1			JAVA	OS-MPI/AIX
	AIX_Swap_Mon	4.1	OS	Swap	JAVA	OS-MPI/AIX
	AIX_Syslog_Ind	4.1			JAVA	OS-MPI/AIX
	AIX_System_Perf	4.1	OS	System	JAVA	OS-MPI/AIX
	AIX_TableUtil_Mon	4.1	OS	Kernel	JAVA	OS-MPI/AIX
	BOOM_Messages	2.25				BOOM-Basic
	BOOM_Ping	2.13	BOOM	BOOM	JAVA	BOOM-Basic

Selected (0) Total Policies (0) Indications (0) Monitors (0)

The **Policy Overview** view shows all policies and offers two filter alternatives:

Additional Columns

Related Assignment
Related Agents

Filter on Name

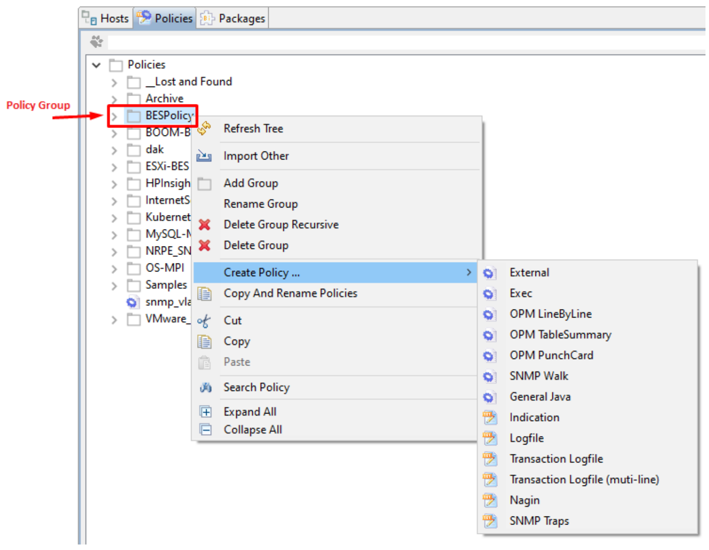
Specify the full policy name or a part of the name.

Policy tree folder "_Lost and Found"
The server job Find_Lost_Policies detects lost policies and adds them to the policy tree folder _Lost and Found. e.g. If the server job detects a policy which is not known in his policyGroups.xml file then the policy will be added to the _Lost and Found folder. An user can decide whether the policy should be deleted or copied to another folder.

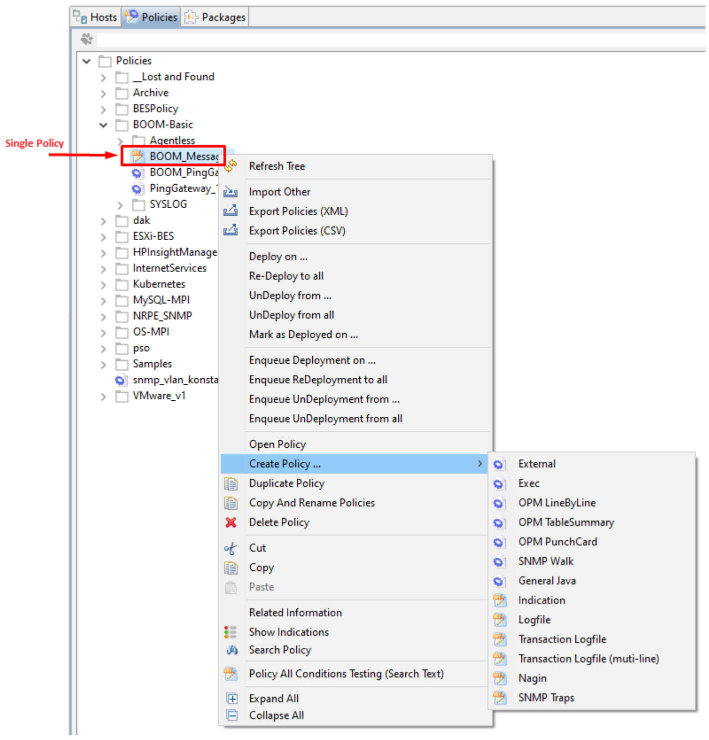
6.5.2. Policy Operations

All policies are displayed in form of a tree hierarchy. Double-click on a single policy to open the appropriate policy details. All available actions are located in the context menu.

Right-click on a **Policy Group** to open the appropriate context menu:



Right-click on a **single Policy** to open the appropriate context menu:



Import Other:	Import of a single policy from other vendors.
Export Policies:	Export the policy tree to a specified location. For the policy export XML respectively CSV format is selectable. Hint: CSV formatted output provides policy conditions only separated by ";" and any further policy attributes are not part of the CSV output.
Add Group:	Create a new policy group (tree folder).
Rename Group:	Rename a policy group.
Delete Group Recursive:	The selected policy group will be completely removed. That means all policies and all subfolders inside the policy group will be deleted.

Delete Group:	This will delete the selected policy group, but it will also move the complete content to the policy group (toplevel folder).
Deploy on:	Allows to deploy the selected policies to one or more *boom* agents. It also starts necessary monitor tasks. During the deployment process you have to select the agents where the selected policies should be deployed. For more information see the paragraph Deploy on/Undeploy below.
Redeploy to all:	Allows to trigger re-deployment of the selected policies to all *boom* agents where they were previously deployed.
Undeploy from:	Triggers the removing of the selected policies (and also stops monitor task) from the selected *boom* agents. During the undeployment process you have to select the agents from where the selected policies should be undeployed. For more information see the paragraph Deploy on/Undeploy below.

If the policy was deployed as part of an assignment group it must be undeployed together with the assignment or unlinked from the assignment group first.

Undeploy from all:	Triggers the removing of the selected policies from all *boom* agents where they were previously deployed.

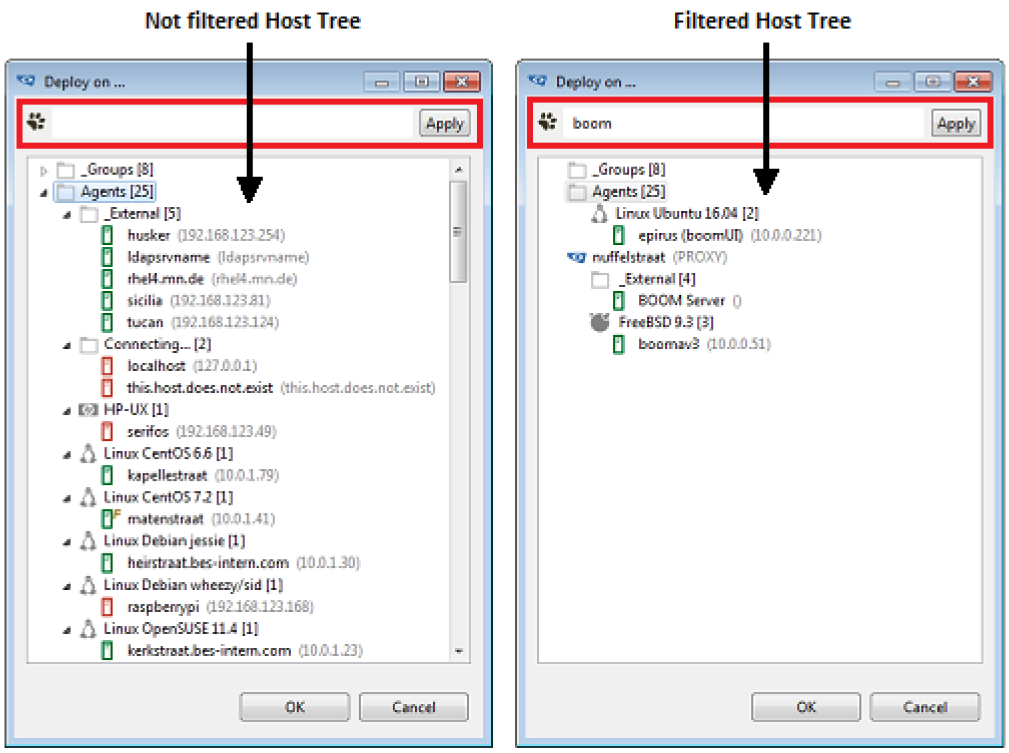
If the policy was deployed as part of an assignment group, it must be either undeployed together with the assignment or unlinked from the assignment group first.

Mark as Deployed on:	Enqueue Deployment on... Enqueue ReDeployment to all Enqueue UnDeployment from Enqueue UnDeployment from all

All "Enqueue Deployment" actions will be added to the Deployment Queue and must be triggered manually! This actions will NOT be started automatically! For more details about the Deployment Queue see the Chapter Assignments Summary.

Deploy on / Undeploy from:

During the deployment/undeployment process you need to select the *boom* agents where you want to deploy/undeploy the polices. A filter function inside the 'Deployment Dialog' helps to find the appropriate agents. Just enter some letters that matches with the *boom* agents you need for the deployment process and press the 'Apply' button. To remove an existing filter from the tree you have to clean the filter input field and press the 'Apply' button.



```
_Groups [8]
Agents [25]
  _External [5]
    husker (192.168.123.254)
    ldapsrvname (ldapsrvname)
    rhel4.mn.de (rhel4.mn.de)
    sicilia (192.168.123.81)
    tucan (192.168.123.124)
  Connecting... [2]
    localhost (127.0.0.1)
    this.host.does.not.exist (this.host.does.not.exist)
  HP-UX [1]
    serifos (192.168.123.49)
  Linux CentOS 6.6 [1]
    kapellestraat (10.0.1.79)
  Linux CentOS 7.2 [1]
    matenstraat (10.0.1.41)
  Linux Debian jessie [1]
    heirstraat.bes-intern.com (10.0.1.30)
  Linux Debian wheezy/sid [1]
    raspberrypi (192.168.123.168)
  Linux OpenSUSE 11.4 [1]
    kerkstraat.bes-intern.com (10.0.1.23)
```

```
boom
_Groups [8]
Agents [25]
  Linux Ubuntu 16.04 [2]
    epirus (boomUI) (10.0.0.221)
nuffelstraat (PROXY)
  _External [4]
    BOOM Server ()
  FreeBSD 9.3 [3]
    boomav3 (10.0.0.51)
```

Open Policy:

This will open the appropriate policy details in the 'Policy Details View'.

Add New Monitor Policy:

A new monitor policy tab will be opened. Please note: You have to save the policy before you can deploy it. After the policy has saved, the policy name cannot be changed any more. If you need to change a name, you have to use the 'Copy and Rename Policies'. Play Video

Add New Indication Policy:

A new indication policy tab will be opened. Please note: You have to save the policy before you can deploy it. After the policy has saved the first time, the policy name cannot be changed any more. If you need to change a policy name, you have to use the function 'Copy and Rename Policies'.

Duplicate Policy:

The duplicate policy function is used to duplicate a single policy within the same policy group. The duplication of multiple policies is described in 'Copy and Rename Policies' below. Please note: The policy name is unique and therefore it is not possible to have multiple policies with the same name. If you duplicate a policy you must provide a new unique name to the policy.

Cut/ Copy/ Paste:

Copy/Paste gives you the possibility to copy a single policy within the same policy group or to another policy group. Cut/Paste allows you to move a single or multiple policies to another policy group.

Copy And Rename Policies:

Copy or rename a single or multiple policies. For more details see the paragraph Copy/Rename Policies below

Delete Policy:

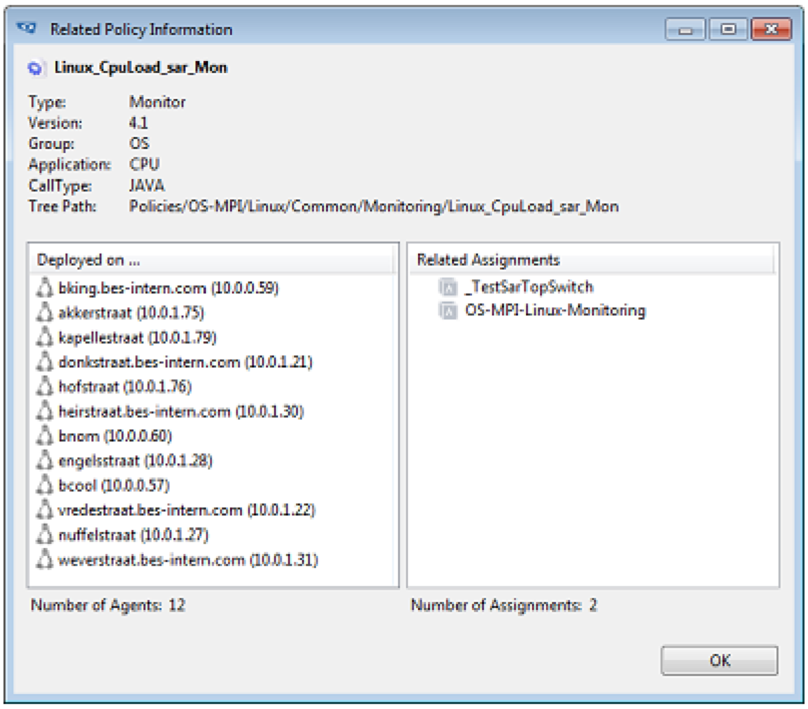
A single policy or multiple policies can be selected and deleted.

Show Indications:

This will open a new indication tab displaying all active indications for the selected policy.

Show Related Agents/Assignments:

This shows general policy information like type, version, group, etc. and a list of agents and assignments on which this policy is deployed on.



Related Policy Information

Linux_CpuLoad_sar_Mon

Type:	Monitor
Version:	4.1
Group:	OS
Application:	CPU
CallType:	JAVA
Tree Path:	Policies/OS-MPI/Linux/Common/Monitoring/Linux_CpuLoad_sar_Mon

Deployed on ...	Related Assignments
bking.bes-intern.com (10.0.0.59)	_TestSarTopSwitch
akkerstraat (10.0.1.75)	OS-MPI-Linux-Monitoring
kapellestraat (10.0.1.79)	
donkstraat.bes-intern.com (10.0.1.21)	
hofstraat (10.0.1.76)	
heirstraat.bes-intern.com (10.0.1.30)	
bnom (10.0.0.60)	
engelsstraat (10.0.1.28)	
bcool (10.0.0.57)	
vredestraat.bes-intern.com (10.0.1.22)	
nuffelstraat (10.0.1.27)	
weverstraat.bes-intern.com (10.0.1.31)	

Number of Agents: 12

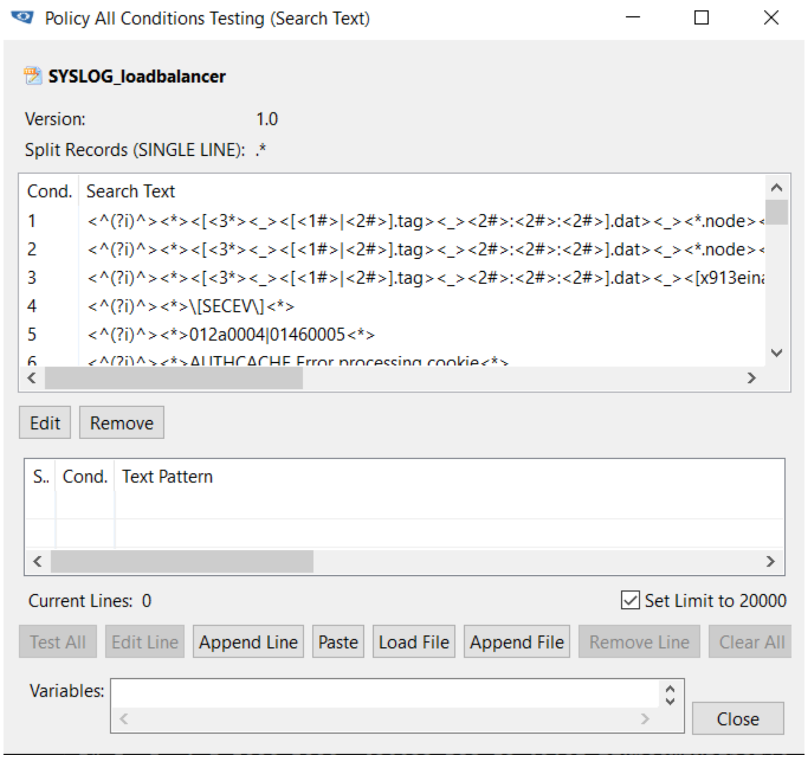
Number of Assignments: 2

Search Policy:

Opens a policy search view with a "Global Search" and an "Advanced Search" utility.

Policy All Conditions Testing (Search Text):

This will open all the policy conditions (indication policy only) in a separate window for testing and comparing attribute "Search Text" with a user specified pattern sequence.



Policy All Conditions Testing (Search Text)

SYSLOG_loadbalancer

Version: 1.0

Split Records (SINGLE LINE): .*

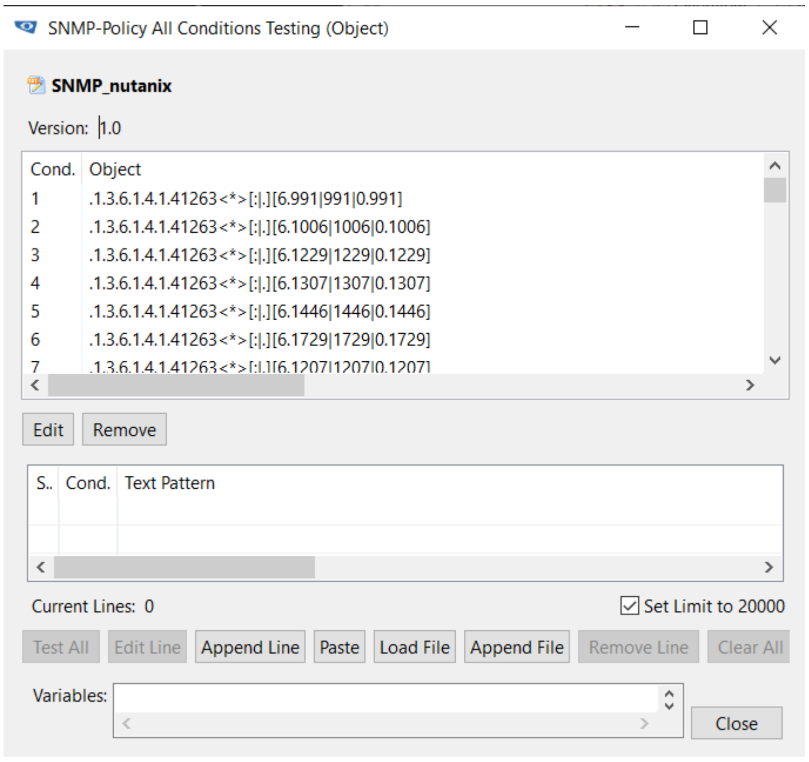
```
Cond.  Search Text
1      <^(?i)^><*><[<3*><_><[<1#>|<2#>].tag><_><2#>:<2#>:<2#>].dat><_><*.node><
2      <^(?i)^><*><[<3*><_><[<1#>|<2#>].tag><_><2#>:<2#>:<2#>].dat><_><*.node><
3      <^(?i)^><*><[<3*><_><[<1#>|<2#>].tag><_><2#>:<2#>:<2#>].dat><_><[x913eina
4      <^(?i)^><*>\[SECEV\]<*>
5      <^(?i)^><*>012a0004|01460005<*>
6      <^(?i)^><*>AUTHCACHE Error processing cookie<*>
```

Current Lines: 0

☑ Set Limit to 20000

SNMP-Policy All Conditions Testing (Object):

This will open all the SNMP-policy conditions (indication policy with "Application" attribute setting "SNMPTrapd" only) in a separate window for testing and comparing attribute "Object" with a user specified pattern sequence.



SNMP-Policy All Conditions Testing (Object)

SNMP_nutanix

Version: 1.0

Cond.	Object
1	.1.3.6.1.4.1.41263<*>[:|.][6.991|991|0.991]
2	.1.3.6.1.4.1.41263<*>[:|.][6.1006|1006|0.1006]
3	.1.3.6.1.4.1.41263<*>[:|.][6.1229|1229|0.1229]
4	.1.3.6.1.4.1.41263<*>[:|.][6.1307|1307|0.1307]
5	.1.3.6.1.4.1.41263<*>[:|.][6.1446|1446|0.1446]
6	.1.3.6.1.4.1.41263<*>[:|.][6.1729|1729|0.1729]
7	.1.3.6.1.4.1.41263<*>[:|.][6.1207|1207|0.1207]

S..	Cond.	Text Pattern

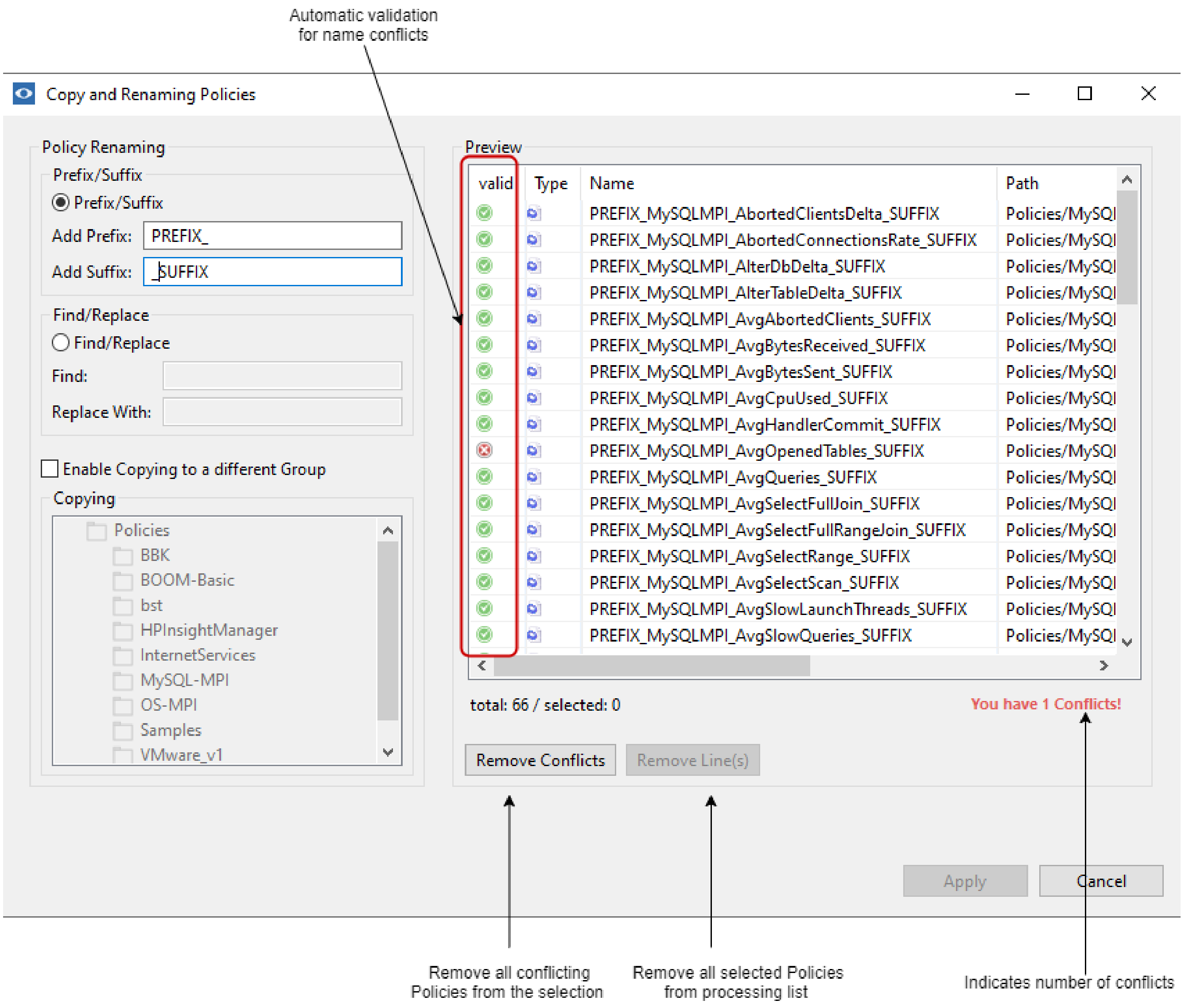
Current Lines: 0

☑ Set Limit to 20000

Variables:

6.5.3. Copy/Rename Policies

To rename one or multiple policies you have to select all the policies you want to copy/rename. If you select a policy group all policies inside this group and inside all subgroups will be processed. If you select a single policy only this policy will be processed. In the 'Copy and Renaming Dialog' you will see a list of all policies that you have selected:



[✓ Prefix/Suffix]

Add Prefix: PREFIX_

Add Suffix: _SUFFIX

Find/Replace

Enable Copying to a different Group [ ]

valid	Type	Name	Path
✓		PREFIX_MySQLMPI_AbortedClientsDelta_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AbortedConnectionsRate_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AlterDbDelta_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AlterTableDelta_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgAbortedClients_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgBytesReceived_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgBytesSent_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgCpuUsed_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgHandlerCommit_SUFFIX	Policies/MySQl
✗		PREFIX_MySQLMPI_AvgOpenedTables_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgQueries_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgSelectFullJoin_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgSelectFullRangeJoin_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgSelectRange_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgSelectScan_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgSlowLaunchThreads_SUFFIX	Policies/MySQl
✓		PREFIX_MySQLMPI_AvgSlowQueries_SUFFIX	Policies/MySQl

You have 1 Conflicts!

Renaming Options:

- adding Prefix/Suffix
- find/replace
- copy policies to a different group

Preview:
Any changes to the Prefix/Suffix or the Find/Replace field will reflect directly in the 'Preview Table'. To exclude one or more policies from the renaming process, select them in the preview table and remove them from the preview table by pressing the button *"Remove line(s)"*.

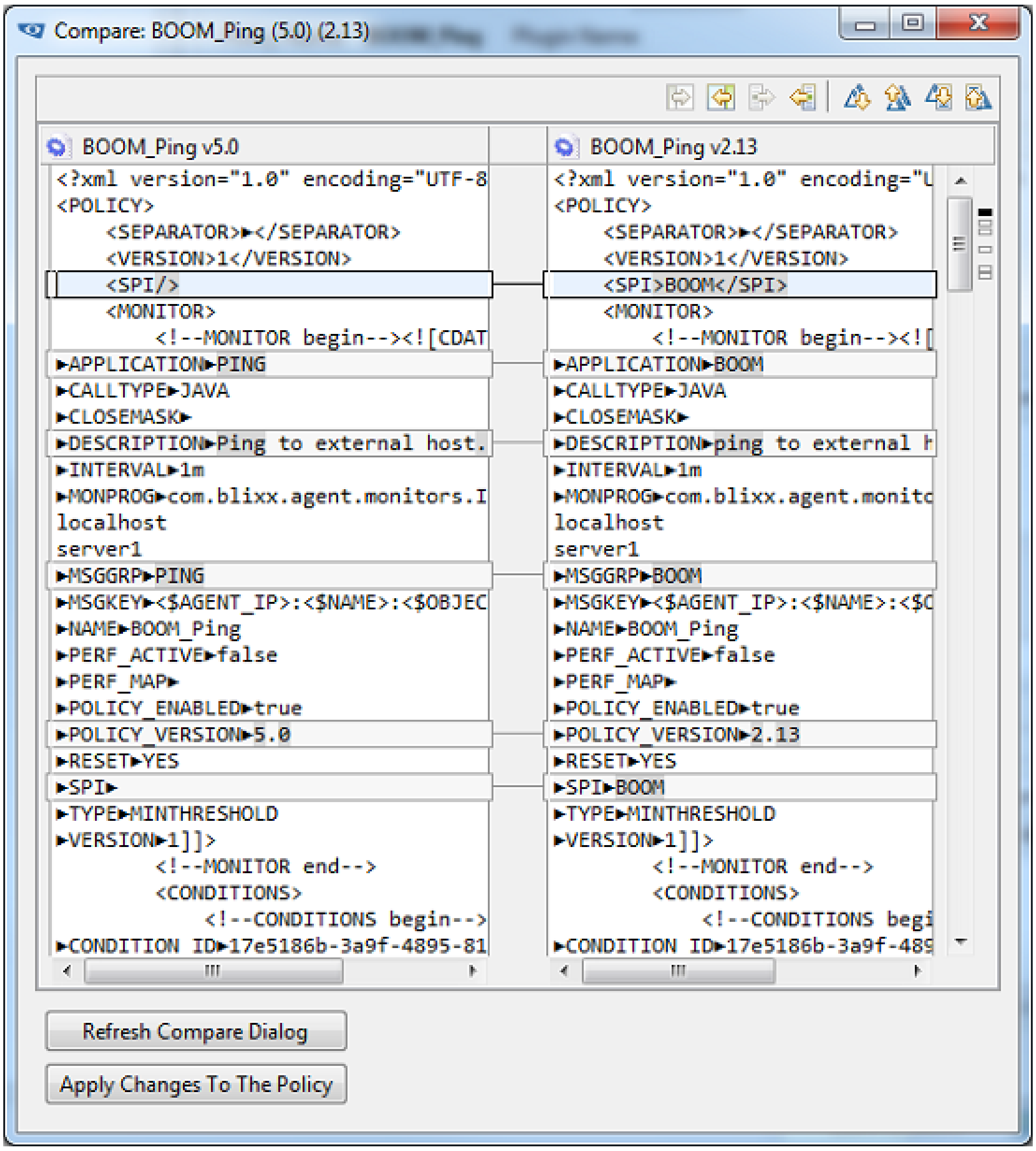
Naming Conflicts:
Changes to the Prefix/Suffix or the Find/Replace field also triggers automatic validation of the policy names. Naming conflicts are indicated in the preview table. To remove naming conflicts you can either change your renaming parameters or you can click on the *"Remove Conflicts"* button. If you click on the button, all policies with conflicts will be deleted from the preview table and excluded from the copying/renaming process.

Copy Policies and save them to a different Group:
If you want to save the renamed policies to a different location, you have to enable the section *"Enable Copying to a different Group"*. All original policies remain unchanged and the renamed policies will be saved to the selected group. If you need to add a new folder inside the policy tree, you can create the folder using the tree context menu.

The Policy name is a unique name across all Policies. If you rename a Policy and the new name already exists, you will get a naming conflict. The renaming process itself will be started by pressing the *"Rename"* button on the bottom right corner.

6.5.4. Compare Policies

Every change that is made to a policy will increase the policy version. To compare the current policy with an older version select the version you want to compare with and press the "compare" button. This opens the 'Policy Compare Dialog':



```
<?xml version="1.0" encoding="UTF-8
<POLICY>
    <SEPARATOR>►</SEPARATOR>
    <VERSION>1</VERSION>
    <SPI/>
    <MONITOR>
        <!--MONITOR begin--><![CDAT
►APPLICATION►PING
►CALLTYPE►JAVA
►CLOSEMASK►
►DESCRIPTION►Ping to external host.
►INTERVAL►1m
►MONPROG►com.blixx.agent.monitors.I
localhost
server1
►MSGGRP►PING
►MSGKEY►<$AGENT_IP>:<$NAME>:<$OBJEC
►NAME►BOOM_Ping
►PERF_ACTIVE►false
►PERF_MAP►
►POLICY_ENABLED►true
►POLICY_VERSION►5.0
►RESET►YES
►SPI►
►TYPE►MINTHRESHOLD
►VERSION►1]]>
        <!--MONITOR end-->
        <CONDITIONS>
            <!--CONDITIONS begin-->
►CONDITION ID►17e5186b-3a9f-4895-81
```

```
<?xml version="1.0" encoding="U
<POLICY>
    <SEPARATOR>►</SEPARATOR>
    <VERSION>1</VERSION>
    <SPI>BOOM</SPI>
    <MONITOR>
        <!--MONITOR begin--><![
►APPLICATION►BOOM
►CALLTYPE►JAVA
►CLOSEMASK►
►DESCRIPTION►ping to external h
►INTERVAL►1m
►MONPROG►com.blixx.agent.monito
localhost
server1
►MSGGRP►BOOM
►MSGKEY►<$AGENT_IP>:<$NAME>:<$O
►NAME►BOOM_Ping
►PERF_ACTIVE►false
►PERF_MAP►
►POLICY_ENABLED►true
►POLICY_VERSION►2.13
►RESET►YES
►SPI►BOOM
►TYPE►MINTHRESHOLD
►VERSION►1]]>
        <!--MONITOR end-->
        <CONDITIONS>
            <!--CONDITIONS begi
►CONDITION ID►17e5186b-3a9f-489
```

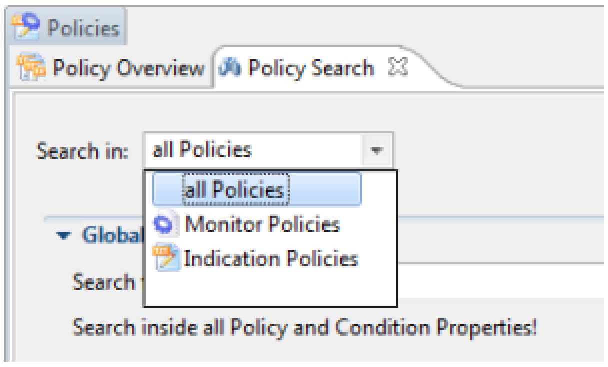
6.5.5. Searching Policies

There are two alternative ways to find the wanted **policy names**

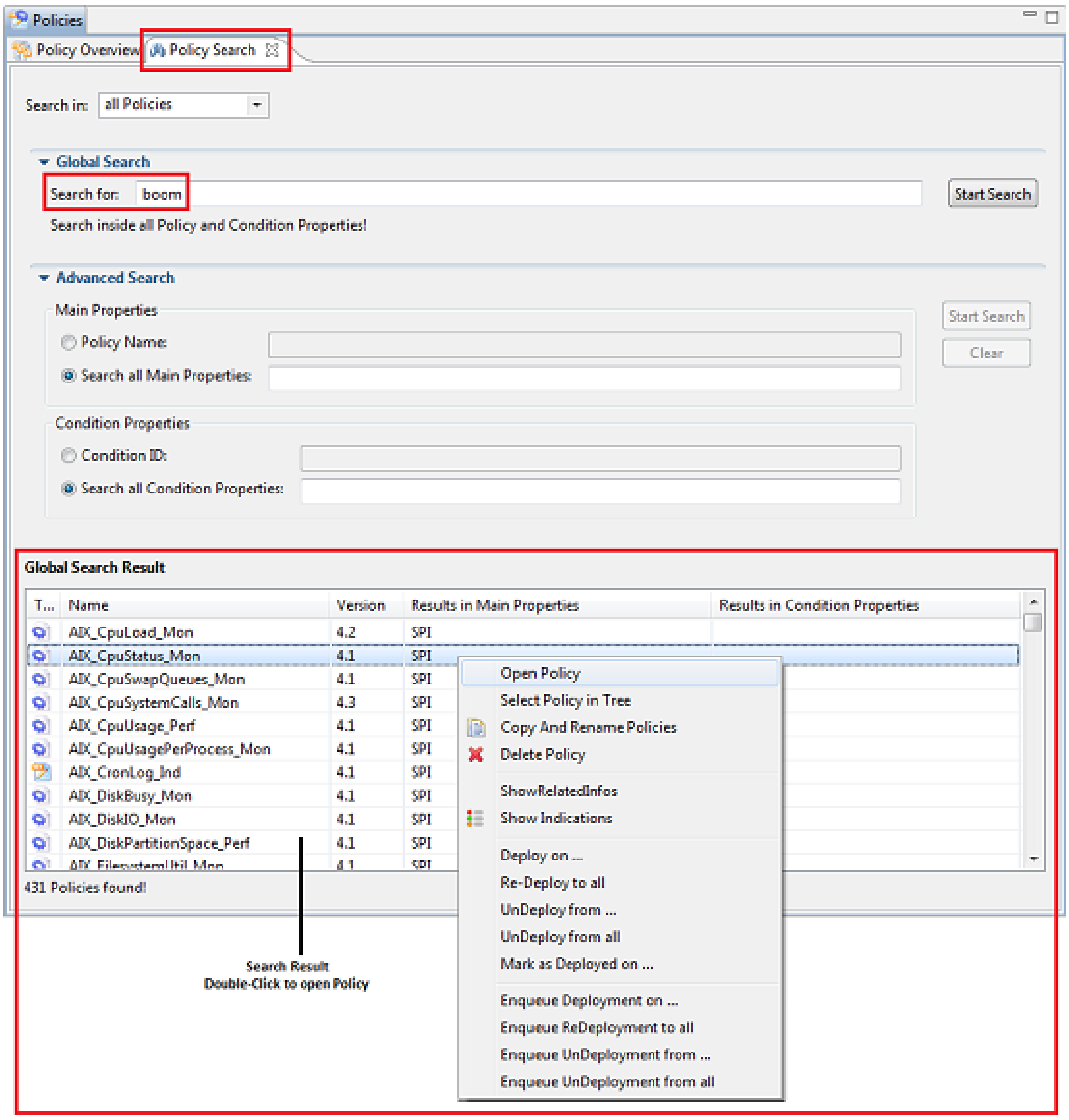
- Use the **Filter on name** field in the **Policy Overview** view. Specify the full policy name or a part of the name.
- Use **Policy Search**. Select **Policy Name** in the **Advanced Search**.

Policy Search provides a powerful algorithm for searching **policy attributes or policy names**.
Right-click in the policy overview view to open the context menu. Select **Search Policy** to open the **Policy Search** view.

First you have to select the search criteria. A search is possible over **all Policies**, **Monitor Policies** only or **Indication Policies** only.



Use **Global Search** to search through all **main attributes** and **condition attributes** of a policy. Specify the full attribute name or a part of the name.



Search in: all Policies

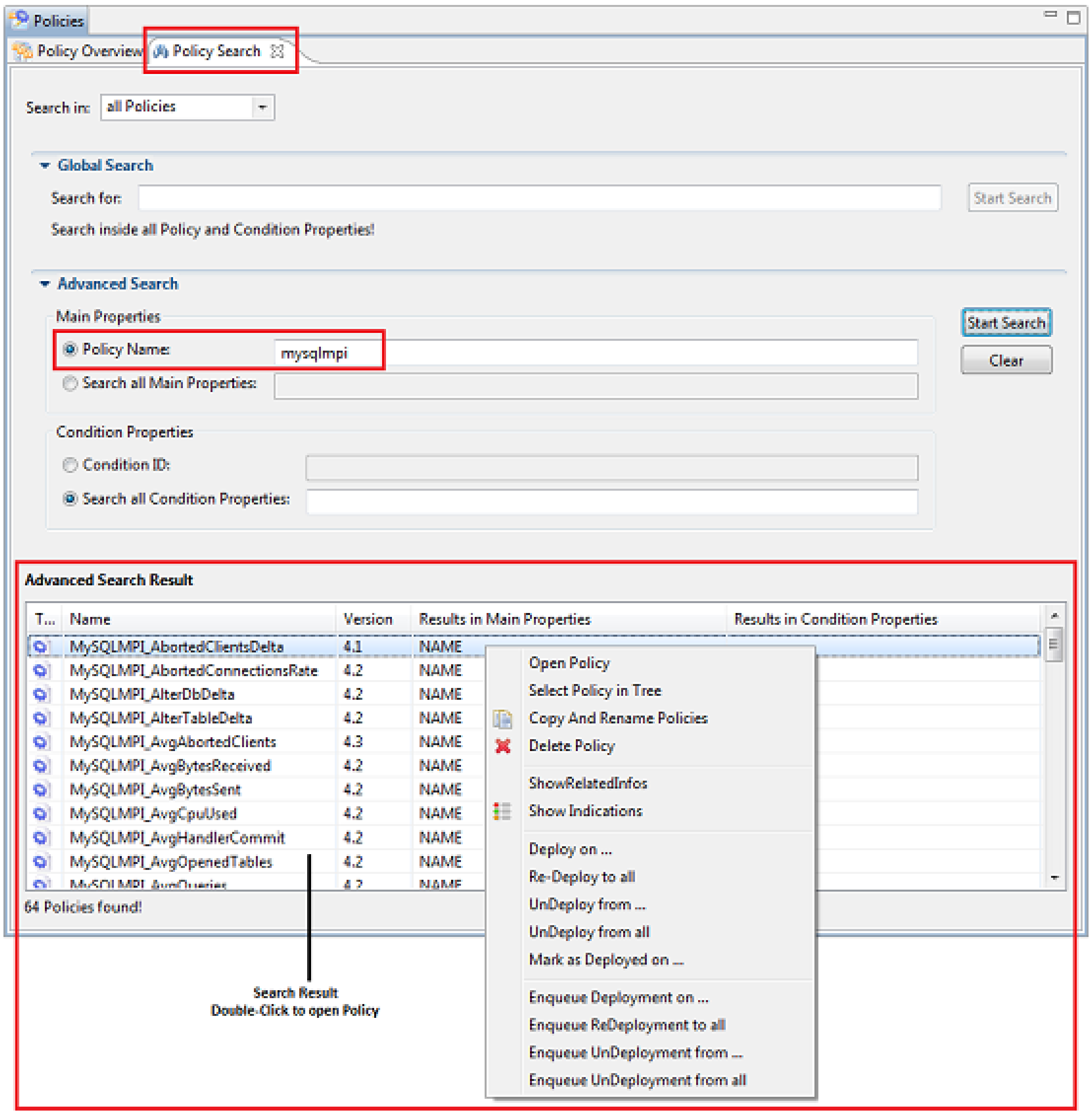
Search for: boom

Global Search Result

T...	Name	Version	Results in Main Properties	Results in Condition Properties
	AIX_CpuLoad_Mon	4.2	SPI	
	AIX_CpuStatus_Mon	4.1	SPI	
	AIX_CpuSwapQueues_Mon	4.1	SPI	
	AIX_CpuSystemCalls_Mon	4.3	SPI	
	AIX_CpuUsage_Perf	4.1	SPI	
	AIX_CpuUsagePerProcess_Mon	4.1	SPI	
	AIX_CronLog_Ind	4.1	SPI	
	AIX_DiskBusy_Mon	4.1	SPI	
	AIX_DiskIO_Mon	4.1	SPI	
	AIX_DiskPartitionSpace_Perf	4.1	SPI	
	AIX_FilesystemUtil_Mon	4.1	SPI	

431 Policies found!

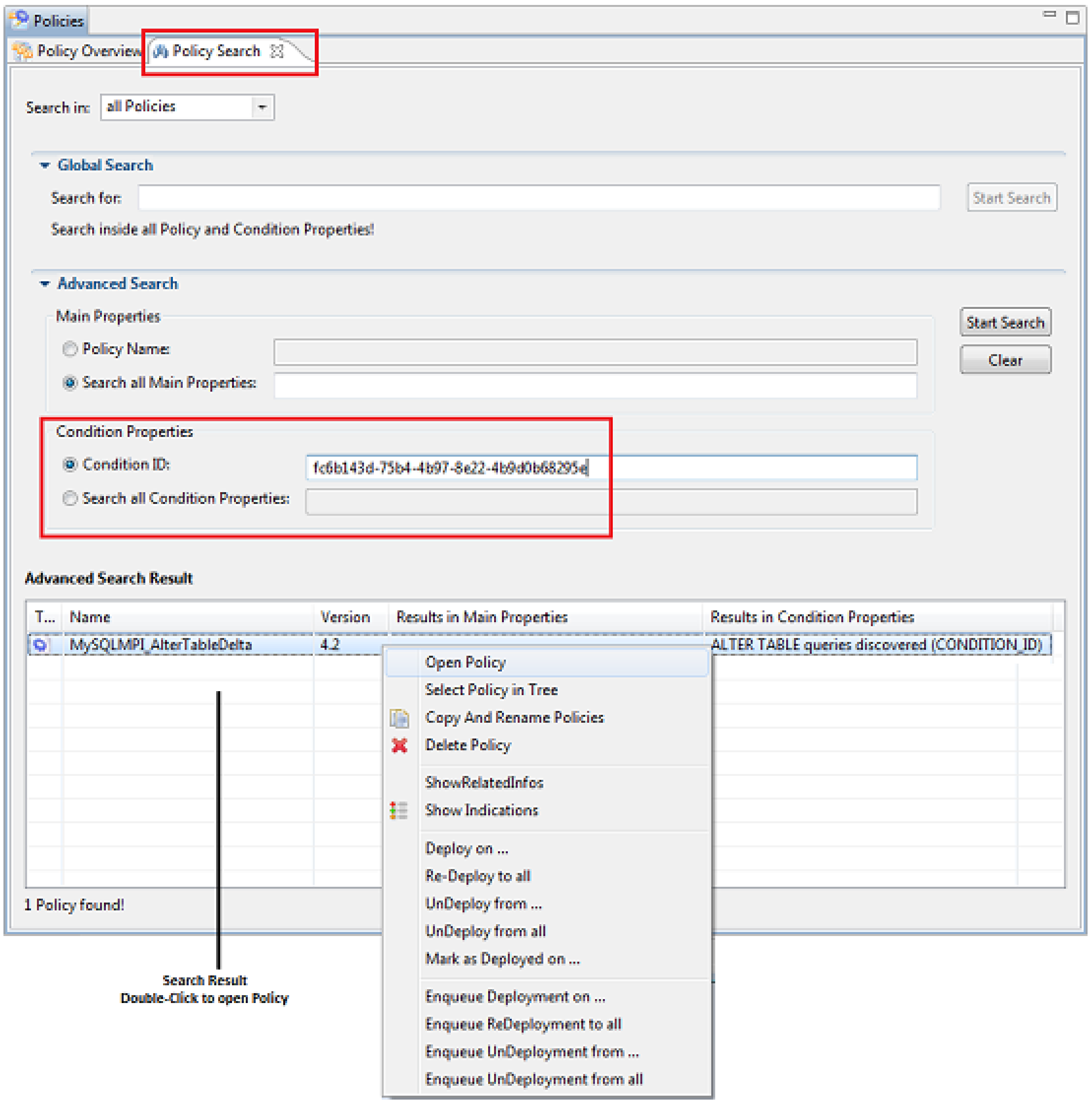
Use **Advanced Search** to either search through all **Main Properties** (select Policy Name or Main Attributes: all attributes without condition attributes)



T...	Name	Version	Results in Main Properties	Results in Condition Properties
	MySQLMPI_AbortedClientsDelta	4.1	NAME	
	MySQLMPI_AbortedConnectionsRate	4.2	NAME	
	MySQLMPI_AlterDbDelta	4.2	NAME	
	MySQLMPI_AlterTableDelta	4.2	NAME	
	MySQLMPI_AvgAbortedClients	4.3	NAME	
	MySQLMPI_AvgBytesReceived	4.2	NAME	
	MySQLMPI_AvgBytesSent	4.2	NAME	
	MySQLMPI_AvgCpuUsed	4.2	NAME	
	MySQLMPI_AvgHandlerCommit	4.2	NAME	
	MySQLMPI_AvgOpenedTables	4.2	NAME	
	MySQLMPI_AvgQueries	4.2	NAME	

64 Policies found!

or **Condition Properties** (select Condition ID or Condition Attributes: all condition attributes, no main attributes) of a policy.



Condition ID: fc6b143d-75b4-4b97-8e22-4b9d0b68295e

Advanced Search Result

T...	Name	Version	Results in Main Properties	Results in Condition Properties
	MySQLMPI_AlterTableDelta	4.2		ALTER TABLE queries discovered (CONDITION_ID)

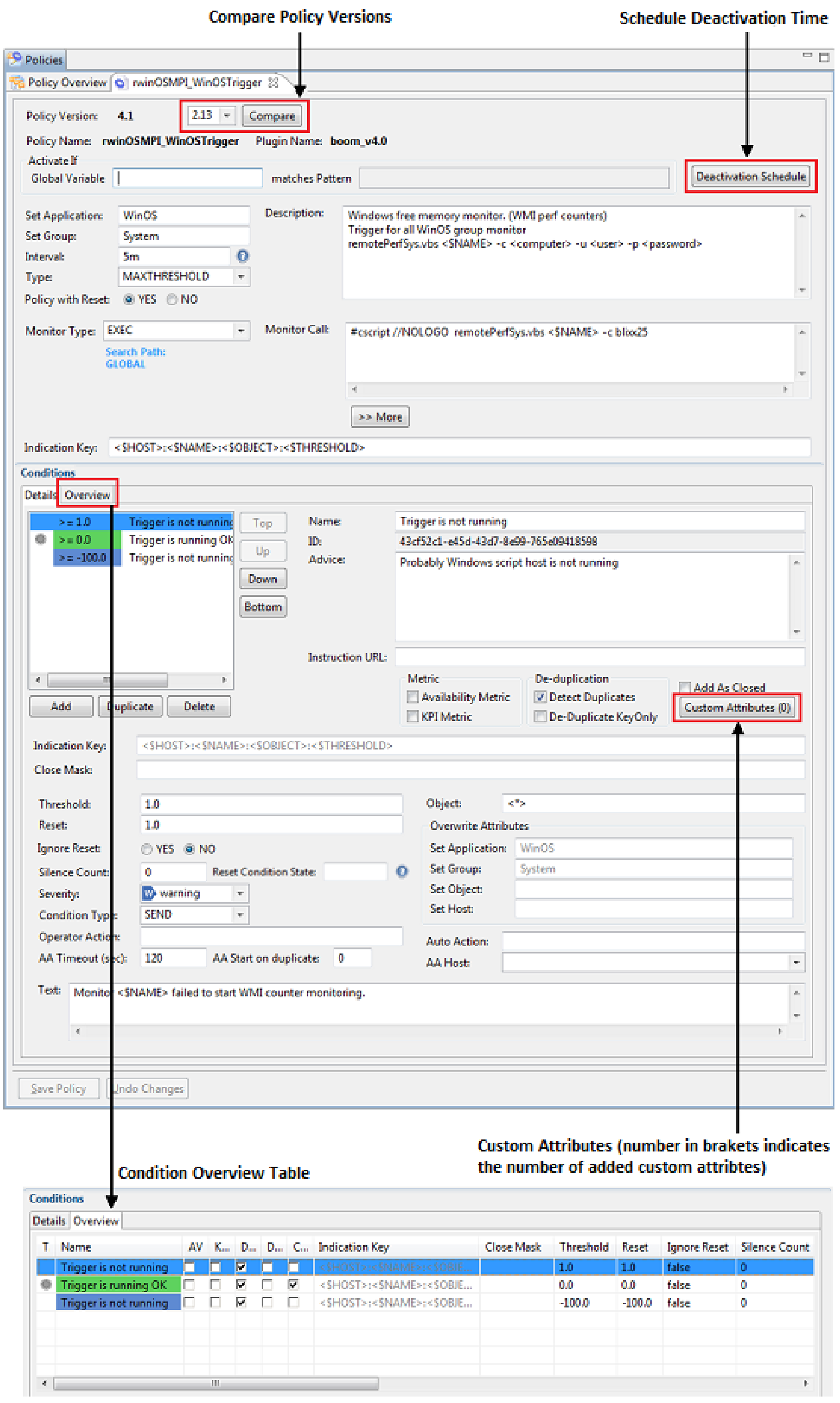
1 Policy found!

6.5.6. Monitor Policy

Monitor policies are designed to evaluate numeric values received from different sources. More about concept see Monitor Policy (Threshold monitoring)

6.5.6.1. Monitor Policy Details

Description of the monitor policy fields:



Policy Version: 4.1 2.13

Policy Name: rwinOSMPI_WinOSTrigger Plugin Name: boom_v4.0

Activate If Global Variable matches Pattern

Set Application: WinOS
Set Group: System
Interval: 5m
Type: MAXTHRESHOLD
Policy with Reset: [✓ YES] NO

Description:
Windows free memory monitor. (WMI perf counters)
Trigger for all WinOS group monitor
remotePerfSys.vbs <SNAME> -c <computer> -u <user> -p <password>

Monitor Type: EXEC
Search Path: GLOBAL

Monitor Call: #cscript //NOLOGO remotePerfSys.vbs <$NAME> -c blixx25

Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$THRESHOLD>

>= 1.0	Trigger is not running
>= 0.0	Trigger is running OK
>= -100.0	Trigger is not running

Name: Trigger is not running
ID: 43cf52c1-e45d-43d7-8e99-765e09418598
Advice: Probably Windows script host is not running
Instruction URL:

Metric: Availability Metric, KPI Metric
De-duplication: [✓ Detect Duplicates], De-Duplicate KeyOnly
Add As Closed

Indication Key: <$HOST>:<$NAME>:<$OBJECT>:<$THRESHOLD>
Close Mask:

Threshold: 1.0
Reset: 1.0
Ignore Reset: YES [✓ NO]
Silence Count: 0 Reset Condition State:
Severity: warning
Condition Type: SEND
Operator Action:
AA Timeout (sec): 120 AA Start on duplicate: 0

Object: <*>
Overwrite Attributes
Set Application: WinOS
Set Group: System
Set Object:
Set Host:
Auto Action:
AA Host:

Text: Monitor <$NAME> failed to start WMI counter monitoring.

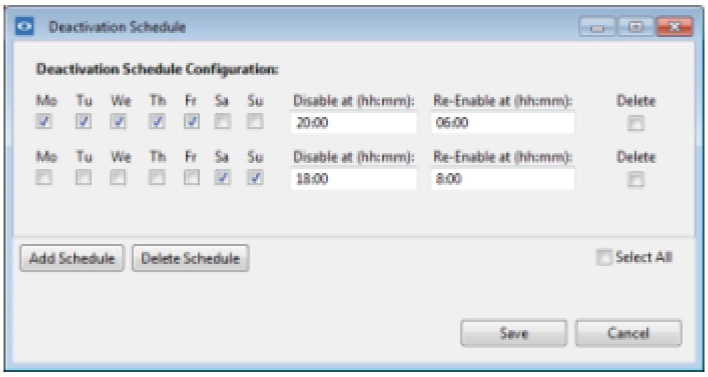
T	Name	AV	K...	D...	D...	C...	Indication Key	Close Mask	Threshold	Reset	Ignore Reset	Silence Count
	Trigger is not running	☐	☐	✓	☐	☐	<$HOST>:<$NAME>:<$OBJE...		1.0	1.0	false	0
●	Trigger is running OK	☐	☐	✓	☐	✓	<$HOST>:<$NAME>:<$OBJE...		0.0	0.0	false	0
	Trigger is not running	☐	☐	✓	☐	☐	<$HOST>:<$NAME>:<$OBJE...		-100.0	-100.0	false	0

Policy Version:	This shows the current version of the policy. The field will be automatically updated when changes are saved on the *boom* server. The drop-down menu at the right side displays all previous versions of this policy. The 'Compare' button allows you to review the history of changes that have been made to the the policy.
Compare:	With the compare function you can compare the current policy with an older version. More details about the compare function see chapter Policy Management.
Monitor Name:	This is the unique name of the policy (unique across all kind of policies).

Global Variable: The policy is only activated if the specified global variable matches the specified value. For more information on setting global variables see chapter Global Variables.

Plugin Name: The Plugin Name is a non editable field. The name comes from 3rd party plug-ins.

Deactivation Schedule:



Deactivation Schedule Configuration:

Mo	Tu	We	Th	Fr	Sa	Su	Disable at (hh:mm):	Re-Enable at (hh:mm):	Delete
✓	✓	✓	✓	✓	☐	☐	20:00	06:00	☐
☐	☐	☐	☐	☐	✓	✓	18:00	8:00	☐

Interval: The Interval specifies a polling interval for the monitor. Format of interval field:

Regular: <n>T

n - number of units,
T - type of units(m – minutes, h-hours, d-days).

5m - 5 minutes
10h - 10 hours
7d - 7 days

Cron-like: T<n1:n2:n3:n4>

n1 - days from Monday,
n2 - hours,
n3 - minutes,
n4 - seconds,
T - type of units(h-hours, d-days, w-weeks).

h0:00:20:10 - every hour on 20 min 10 sec
d0:18:11:05 - every day at 18:11:05
w4:23:10:00 - every week Friday on 23:10
where w0=monday, w1=tuesday, w2=wednesday...

Description: A description of the policy.

Type:	Possible types are: MAXTHRESHOLD or MINTHRESHOLD. The type of the monitor defines the main meaning of expected values: MINTHRESHOLD: The monitor delivers an indication when the submitted value is equal or less one of the specified threshold. One example can be the "Free Disk Space Monitor". MAXTHRESHOLD: The monitor delivers an indication when the submitted value is bigger or equal one of the specified thresholds. One example can be the "System CPU Load". MINONLYCHANGES: The same as MINTHRESHOLD except, that specified conditions used for indication construction and policy sends an indication each time the value changes. MAXONLYCHANGES: The same as MAXTHRESHOLD except, that specified conditions used for indication construction and policy sends an indication each time the value changes. STDDEV: Standard deviation monitor. Similar to MAXTHRESHOLD where thresholds are defining absolute "Distance" between value and and the Mean.

Please see chapter: Monitor Policy (Threshold monitoring) for more information.

Set Application:

The default application attribute can be set in the monitor policy. It is assigned to all indications that are generated by the monitor policy but is overwritten by a value set in the policy conditions.

Set Group:

The default indication group attribute can be set in the monitor policy. It is assigned to all indications that are generated by the monitor policy but is overwritten by a value set in the policy conditions.

Policy with Reset:

YES|NO flag. If this flag is set to "NO" all delivered values from this monitor will be delivered to the server without suppression. In other words "NO" flag defines "continuous" monitor.

Monitor Type:

Possible types are JAVA | EXEC | EXTERNAL | OPM. They define the type of the Monitor trigger.

JAVA expects a Java monitor deployed to the *boom* agent.
EXEC will trigger binary or script available on the *boom* agent.
EXTERNAL just waits for the monitor value from any external process.
OPM: Output Parser Monitor
SNMP: SNMP Walk or Get Monitor

For more detailed information see the paragraph Monitor Types below or chapter Monitor Policy (Threshold monitoring).

Call:

In this field you specify the Java class name or an executable/script with parameters that must be executed in the specified interval.

1. JAVA monitor expects a fully qualified Java class name in the first line.
2. EXEC monitor expects a binary or a script name relative to the agent’s plug-in directory ($BOOM_ROOT/spi/).
3. IF EXEC binary is a **system tool** and available in global PATH use the **'#'** character at the first position: (otherwise the agent uses the path /opt/boom/agent/spi)

```
#<exec_name>...
i.e.
#CScript scriptname [params]
#df -kl
#sh -c "/path/scriptname"
```

Indication Key: (on Policy level)

A key template identifies the monitor value.

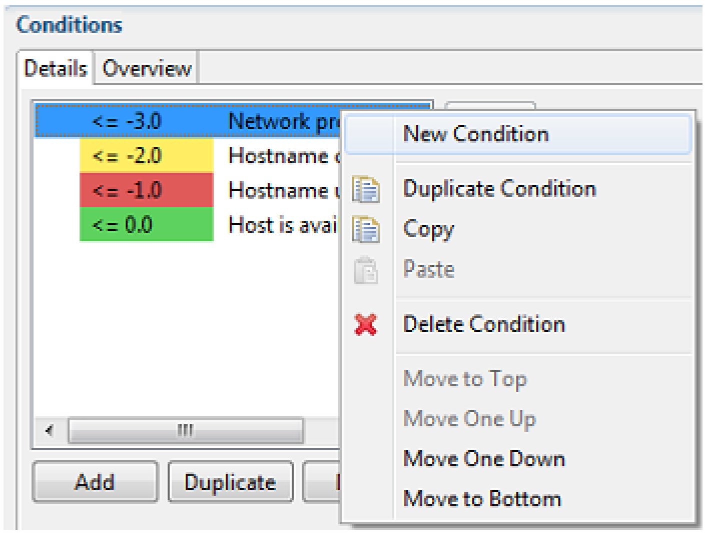
The indication key contains a list of supported variables that will be resolved during the processing of the incoming value. The result string is used for the correlation with the following indication for the auto acknowledgment and the duplicate suppression.

6.5.6.2. Monitor Policy Conditions

This section takes care of the monitor threshold lists. The order of the conditions depends on the type of the monitor. For MAXTHRESHOLD monitors the biggest threshold value must be on top of the list. The MINTHRESHOLD type expects the lowest value on top. The incoming values will be compared with the thresholds, resets and the object filters specified in the condition list in a order that is specified in the policy. The first condition that matches with the submitted value will be used as the source to generate a new indication. All following conditions will be skipped.

Field Description

Condition List:

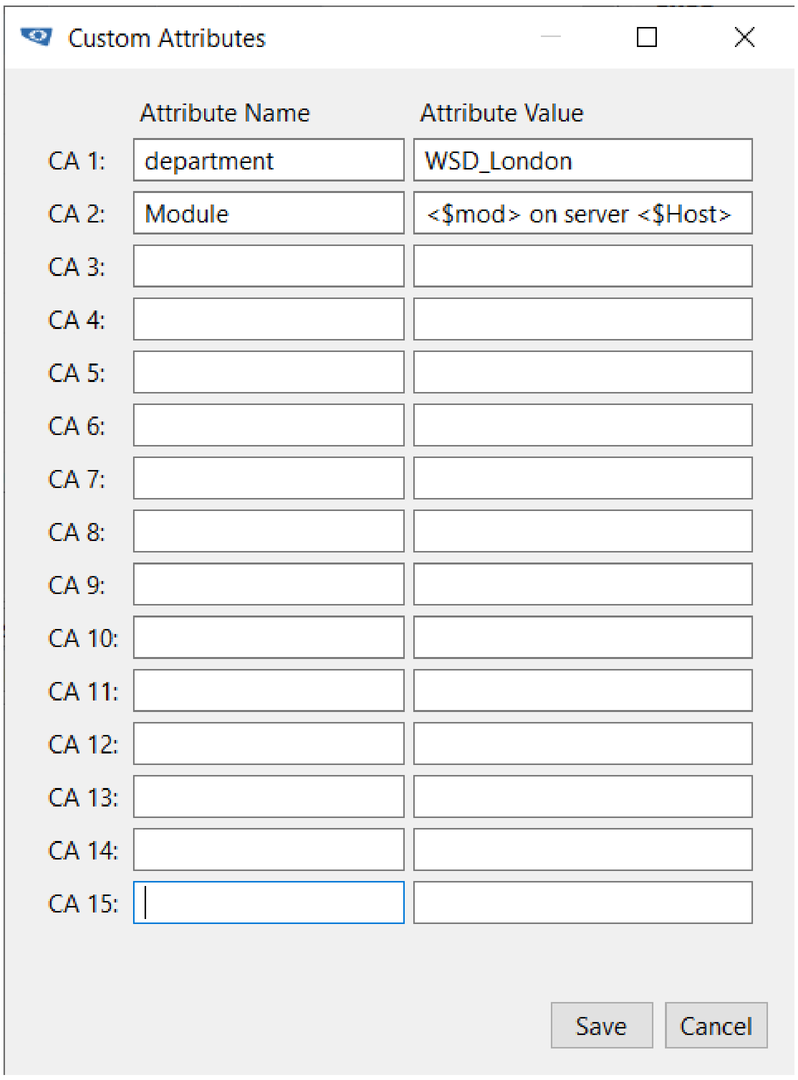


Name:	Name of the condition.
ID:	Auto generated ID of the condition.
Advice:	An advice message that will be shown in the indication browser
Instruction URL:	Define the URL of an external knowledge base e.g. http://myInstruction.server.net. Agent and indication variables can be used in the instruction URL.
Availability Metric:	Defines if the submitted value that matched with this condition has an impact on the availability of the monitored system.

KPI Metric:	This flag defines if the submitted value must be treated as a 'Key Performance Indicator'.
Add as Closed:	This will add the message already as a 'closed message' to the indication browser.
De-duplication:	If the close mask is set in the policy details one of the duplication flags (Detect Duplicates or De-Duplicate KeyOnly) has to be set. The flags cannot be unchecked in the case where the close mask is set.
Detect Duplicates:	This flag notifies the *boom* server to recognize duplicates for active indications and closed indications which were inserted directly as closed (policy flag "add as Closed" is set). By receiving duplicate indications the *boom* server increases the "duplicate count" and updates the "last duplicate time" of the first received indication. No new entry will be created in the database. The following message attributes are used for recognizing duplicates: AGENT_HOST,HOST,APPLICATION,GROUP,OBJECT,SEVERITY,INDICATION KEY, MESSAGE TEXT
De-Duplicate KeyOnly:	This flag notifies the *boom* server to recognize duplicates for active indications and closed indications which were inserted directly as closed (policy flag "add as Closed" is set) based on the defined indication key. By receiving duplicate indications the *boom* server increases the "duplicate count" and updates the "last duplicate time" of the first received indication. No new entry will be created in the database. The indication key attributes are used for recognizing duplicates hence an **Indication Key has to be defined**.

Custom Attributes:

User defined attributes which can be either just a text or a variable value. CA’s are forwarded to the *boom* server as specified , they aren’t subject to match on the agent. Up to 15 custom attributes can be defined in the 'Custom Attributes Dialog'.
Names of the custom attribute should not contain '=' sign.
Attribute values can use supported variables.



	Attribute Name	Attribute Value
CA 1:	department	WSD_London
CA 2:	Module	<$mod> on server <$Host>
CA 3:		
CA 4:		
CA 5:		
CA 6:		
CA 7:		
CA 8:		
CA 9:		
CA 10:		
CA 11:		
CA 12:		
CA 13:		
CA 14:		
CA 15:		

Indication Key: (on Condition level)

A key template identifies the monitor value.

The indication key contains a list of supported variables that will be resolved during the processing of the incoming value. The result string is used for the correlation with the following indication for the auto acknowledgment and the duplicate suppression.

Close Mask:

Pattern that is used to search and close previously submitted indications.

Threshold:

A double value defines the threshold level.

Reset:

A double value defines the reset value of the previously specified threshold. This value allows to keep the severity unchanged in case of a small deviation of following submitted values.

Ignore Reset:

YES | NO . 'YES' - instructs the *boom* agent that this condition must be processed as continuous. All incoming values will be forwarded as new indications to the server. This has no effect if the policy has "Policy with reset" flag set to "NO"

Silence Count:

A count of values since the first match of the threshold that must be suppressed and not delivered to the server. Used when it is necessary to ignore short and not important peaks.

Reset Condition State:

Interval of condition state reset. It starts when condition matches and sends an indication to the server. After specified interval, the Agent will reset condition and as result will initiate sending of new indication when next incoming value matches with the same condition.
Also it resets Silence Count. By using combination of Silence Count and Reset Condition State - it’s possible to create monitor for "match X values on Y time interval" scenario.
Note: Works on Agent v5.7 and higher. Older Agents will silently ignore this setting.

Severity:

Defines the severity of the indication that sends on current threshold level.

Condition Type:

STOP | SEND

STOP - means that an incoming message that matched with the current condition must be ignored and all following conditions must not be checked. This helps to increase the performance of the *boom* agent by using a STOP condition at the beginning of the policy. As result the correlation engine doesn’t need to check all following condition and drops all text messages that have no interest.

SEND - indicates that the matched text must be send as an indication. The rest of the conditions inside the current policy will be automatically ignored.

Object:

Object mask pattern. If the submitted object doesn’t match with the specified pattern, the condition will be skipped. It allows to create a combined threshold list (multiple overlapping threshold conditions) for multiple objects in one policy.

Overwrite Attributes:

After filtering is done and a match for a condition is successful, the attributes in this section can be set to a desired value to overwrite the incoming value or to define a value if the attribute has not been set. In addition, the "Set Application" and "Set Group" attributes overwrite the **default attributes** for "Set Application" and "Set Group" defined in the general part of the policy.

Set Host:

Overwrites the hostname of the submitted monitor value. Default value is an agent hostname. All supported and optional variables can be used in this field as well. In case the monitor submits the hostname as optional variable you can use this variable in the "Set Host" field.

Set Application:

Overwrites the application name of the submitted monitor value or defines a value if the attribute has not been set. In addition, overwrites the **default attribute** of "Set Application" if defined.

Set Object:

Overwrites the object of the submitted monitor value or defines a value if the attribute has not been set.

Set Group:

Overwrites the group of the submitted monitor value or defines a value if the attribute has not been set. In addition, overwrites the **default attribute** of "Set Group" if defined.

Auto Action:

The Auto Action will be automatically triggered by the ***boom*** Server if the condition has matched. Indication variables can be used in the call of the executable i.e. as parameters and will be resolved during the execution. The returned output of the action will be visible as an annotation in the details window of the according indication.

Default working directory for actions is $BOOM_ROOT/spi (instrumentation directory of agent)

For example a policy which monitors the status of an application process detects that the process isn't running anymore. The matching condition includes a call of a restart script as an automatic action. The execution is triggered automatically and restarts the process. The output of the script if available will be visible for the operator in the annotation section of the details of the indication.

Example:

```
mail.sh teamA <$OBJECT> <$ORIG_TEXT>
```

This automatic action will execute a mail script which will send an email to the mail account "teamA" with the resolved indication object as the mail subject and the resolved indication text as the message body.

AA Host:

The target agent hostname for the remote automatic action. If nothing is specified, the action will be triggered on the agent where the indication was created. All indication variables are supported and will be resolved.

AA Timeout:

Timeout in seconds for the remote automatic action.

AA Start on duplicate:

Set the flag to a value N. The AutoAction will be activated after N duplicates.

Operator Action:

This action works in the same way as the Automatic Action but it's execution will be triggered manually by an operator through a button in the indications detail window. This action is also defined in the policy and its output will be displayed in the annotations section in the indications details window as well.

Default working directory for actions is $BOOM_ROOT/spi (instrumentation directory of agent)

Text:	Sets the Indication text. All supported and optional variables will be replaced with the submitted values.

Use CTRL+Space in the policy fields to pop-up the variable list.

6.5.6.3. Monitor Types

The 'Monitor Type' defines the type of the monitor trigger.
For more information see chapter Monitor Policy (Threshold monitoring).

The following monitor types are possible:

EXEC: - will trigger binary or script available on the *boom* agent.
JAVA: - expects a Java monitor deployed to the *boom* agent.
EXTERNAL: - waits for the value from any external process.
OPM: - there are three types of OPM (output parser monitor)

- **LineByLine** - process line by line and submits matched values
- **TableSummary** - process output as table columns and makes summarization
- **PunchCard** - extract variables from the output and calculates result value

SNMP - A monitor policy of type SNMP is used for fetching values with SNMP GET or WALK.

6.5.6.4. Supported variables and Pre-Process functions

For more information see chapter Supported Variables and Functions

6.5.6.5. Building a Monitor Policy

Introduction

This example will show you step by step how to build your own monitor policy.

Please note that the goal of this example is an easy to follow guide to create your own policy and therefore is kept on a basic level, without exploring all the features and options available to you.

For a list and explanation of all the features and options please refer to chapter Monitor Policy.

Considerations

Before you begin to build your own policy please consider the following points:

- Min- or Maxthreshold
- Command to execute

Min- or Maxthreshold:	should be easy, if a high value is good use the Minthreshold, if a low value is good then use Maxthreshold. So if you are writing a policy to monitor used diskspace then use Maxthreshold and if you are going to monitor free diskspace use Minthreshold.
Command to execute:	The command to obtain the data which will be processed by the *boom* agent. This command has to / will be / should be a standard command in the OS for which you are writing the policy. Common commands are "df" for diskspace utilization, "ps" for processes or "top" for CPU and memory utilization.

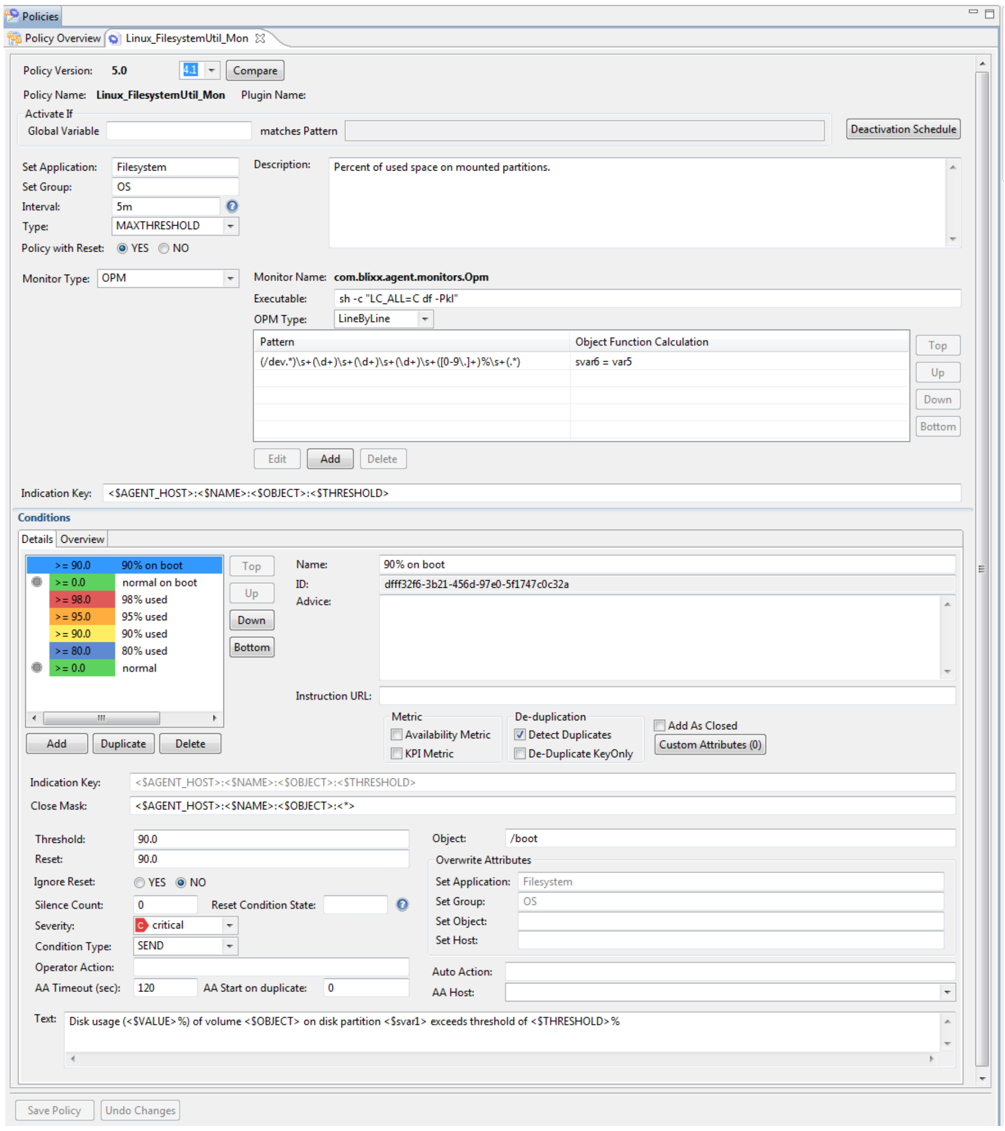
Building the Policy

1. **Monitor Type:** set to OPM, this is the most common monitor type.
2. **Executable:** Put your command here, with parameters if necessary.

   > If you use pipes, semicolons or redirection use this form: `sh -c "your_command_goes_here"`. This will correctly quote your command

3. **OPM Type:** Set to **LineByLine** which is the most common usage. You use LineByLine when all the information you need is in one line. On the other hand you use **TableSummary** when the Information needed is in a column. If the values must be collected from different places select the **PunchCard** type. For more details regarding the OPM types, refer to the chapter Output Parser Monitor (OPM).
4. **Pattern:** This is the most complicated step, but *boom* offers the Pattern Validation which will be a great help. Please define which data/values you want to extract. *boom* uses patterns for this purpose.(*you may know it as regular expressions*).
   a. First open a shell/terminal to the machine where the policy will be running.
   b. Execute the command you put in **Executable**
   c. Open the 'Pattern Validation Dialog (Java)'
   d. Now copy the results/output of the command in the shell and paste it into the 'Pattern Validation Dialog'
   e. In the 'Pattern Validation Dialog' right click on one line for which you want to generate a pattern (*so it will be the line or one of the lines which contents you want*) and click on **Pattern Example.**
   f. Now a pattern is generated. Copy this Pattern and paste it into the 'Pattern Input Field'
   g. Having done this you can further customize the pattern to suit your needs, but for this policy the pattern example should be enough.
   h. Now click on **Validate all Lines** and look on the results area. You will find a number of extracted values and their internal reference name (var1, var2, var3, and so on)
   i. Copy the pattern and change back to the policy. There click on **add** and paste the pattern example in the first line of the appearing dialog window (called **Pattern**).
   j. Now under **Object** define a name which identifies your value (for the free disk space you could use *FREEDISKSPACE*) or use a variable identified under 4.8. if you identify more than one line with your pattern.
   k. Then select "=" and under **Calculation** put the variable you want **OR** put a mathematical operation between the identified variables
   l. Click Save
5. **Indication Key:** Set the indication Key to <**$AGENT_HOST**>:<**$NAME**>:<**$OBJECT**>:<**$THRESHOLD**>

The Conditions



Policies

Policy Overview | Linux_FilesystemUtil_Mon

Policy Version: 5.0 | 4.1 | Compare

Policy Name: Linux_FilesystemUtil_Mon | Plugin Name:

Activate If

Global Variable: | matches Pattern: | Deactivation Schedule

Set Application:	Filesystem
Set Group:	OS
Interval:	5m
Type:	MAXTHRESHOLD
Policy with Reset:	[✓ YES] NO
Monitor Type:	OPM

Description: Percent of used space on mounted partitions.

Monitor Name: com.blixx.agent.monitors.Opm

Executable: sh -c "LC_ALL=C df -Pkl"

OPM Type: LineByLine

Pattern	Object Function Calculation
(/dev.*)\s+(\d+)\s+(\d+)\s+(\d+)\s+([0-9\.]+)%\s+(.*)	svar6 = var5

Top | Up | Down | Bottom | Edit | Add | Delete

Indication Key: <$AGENT_HOST>:<$NAME>:<$OBJECT>:<$THRESHOLD>

Conditions

Details | Overview

	>= 90.0	90% on boot
●	>= 0.0	normal on boot
	>= 98.0	98% used
	>= 95.0	95% used
	>= 90.0	90% used
	>= 80.0	80% used
●	>= 0.0	normal

Top | Up | Down | Bottom | Add | Duplicate | Delete

Name: 90% on boot

ID: dfff32f6-3b21-456d-97e0-5f1747c0c32a

Advice:

Instruction URL:

Metric: ☐ Availability Metric ☐ KPI Metric

De-duplication: ☑ Detect Duplicates ☐ De-Duplicate KeyOnly

☐ Add As Closed

Custom Attributes (0)

Indication Key: <$AGENT_HOST>:<$NAME>:<$OBJECT>:<$THRESHOLD>

Close Mask: <$AGENT_HOST>:<$NAME>:<$OBJECT>:<*>

Threshold:	90.0
Reset:	90.0
Ignore Reset:	YES [✓ NO]
Silence Count:	0
Reset Condition State:	
Severity:	critical
Condition Type:	SEND
Operator Action:	
AA Timeout (sec):	120
AA Start on duplicate:	0

Object: /boot

Overwrite Attributes

Set Application:	Filesystem
Set Group:	OS
Set Object:	
Set Host:	
Auto Action:	
AA Host:	

Text: Disk usage (<$VALUE>%) of volume <$OBJECT> on disk partition <$svar1> exceeds threshold of <$THRESHOLD>%

Save Policy | Undo Changes

Now you have an object and a value which you want to test. Continue to create one or more conditions in which you compare your value against a certain threshold and add a suitable message text you want to appear when the condition is met.
Please consider the following point if you use more than one condition: The first matched condition will be used, after that *boom* stops comparing the value against all following conditions. So be sure to put your conditions in the right order. For the maxthreshold conditions you begin with the highest threshold condition and work your way down. With minthreshold conditions it is the other way around.

Conditions Details:

1. **ID:** Cannot be changed
2. **Advice:** a brief description of this condition
3. **Instruction URL:** keep it empty
4. **Metric:** keep the boxes unchecked
5. **De-duplication:** Uncheck **Detect Duplicates** and check **De-Duplicate KeyOnly**
6. **Add As Closed:** keep it unchecked
7. **Indication key:** Add the the Indication key:
   <$AGENT_HOST>:<$NAME>:<$OBJECT>:<$THRESHOLD>
8. **Close Mask:** Set it to: **<$AGENT_HOST>:<$NAME>:<$OBJECT>:<*>**
9. **Threshold:** Here you can set when the condition matches. If it is a maxthreshold policy all values equal or greater will match. If it is a minthreshold policy all values equal or lower will match. E.g. you are monitoring the

free disk space and you get the value 30%. Using a maxthreshold policy a threshold of 25 would match and one of 35 will not

10. **Reset:** set this value to the same value as threshold.
11. **Ignore Reset:** set it to **no**
12. **Silence Count:** keep it at **0**
13. **Severity:** This depends on your individual requirements, there are 5 levels of severity, which in order are *normal, warning, minor, major, critical* - and a sixth, called unknown. You can, but do not have to set one condition per severity, but it is good practice to do it. So if we continue with our example of a disk free space policy (as maxthreshold policy) you could set one condition with severity critical when a disk is 95% full, one with severity major when only 85% are full, minor with 80%, warning with 75% and normal for everything below 75%.
14. **Condition type:** Set it to **SEND**
15. **AA Timeout (sec):** set it to 120
16. **Object:** There you have three options, depending on your defined object in (see Submitting Monitor Values / Indications):
    - If you set a fixed object name set it to that name (like FREEDISKSPACE)
    - If you set a variable object name:
        - Set "<*>" if you want to treat all object equally
        - Set it to the variable value you set as object name if you want to define specific conditions for certain objects. You can use different objects in different conditions, but be aware that if one condition meets the other are not evaluated, so if you want to use specific object conditions put them before the general ones.
17. **Operator Action, Set Application, Set Group, Set Object, Set Host, Auto Action and AA Host:** leave them blank
18. **Text:** Here you set the text which will appear in the indication when it matches. You can use variables like "<$VALUE>", "<$THRESHOLD>" and others (refer to Monitor Policy to see a complete list).

Now create as many conditions as you need to cover all the different cases which could appear. Take into account that the normal severity condition is special in the way that it should match all cases where the other conditions did not match. So in our example of a free disk space policy set the **threshold** to **"0"** and the reset to **"0"** too. So when the other conditions do not match because the free disk space is enough you will get an indication with severity normal to be sure that everything is ok.

6.5.7. Indication Policy

6.5.7.1. Indication and Hybrid Indication Policy

The indication policy is used to define **two major types** of policies:

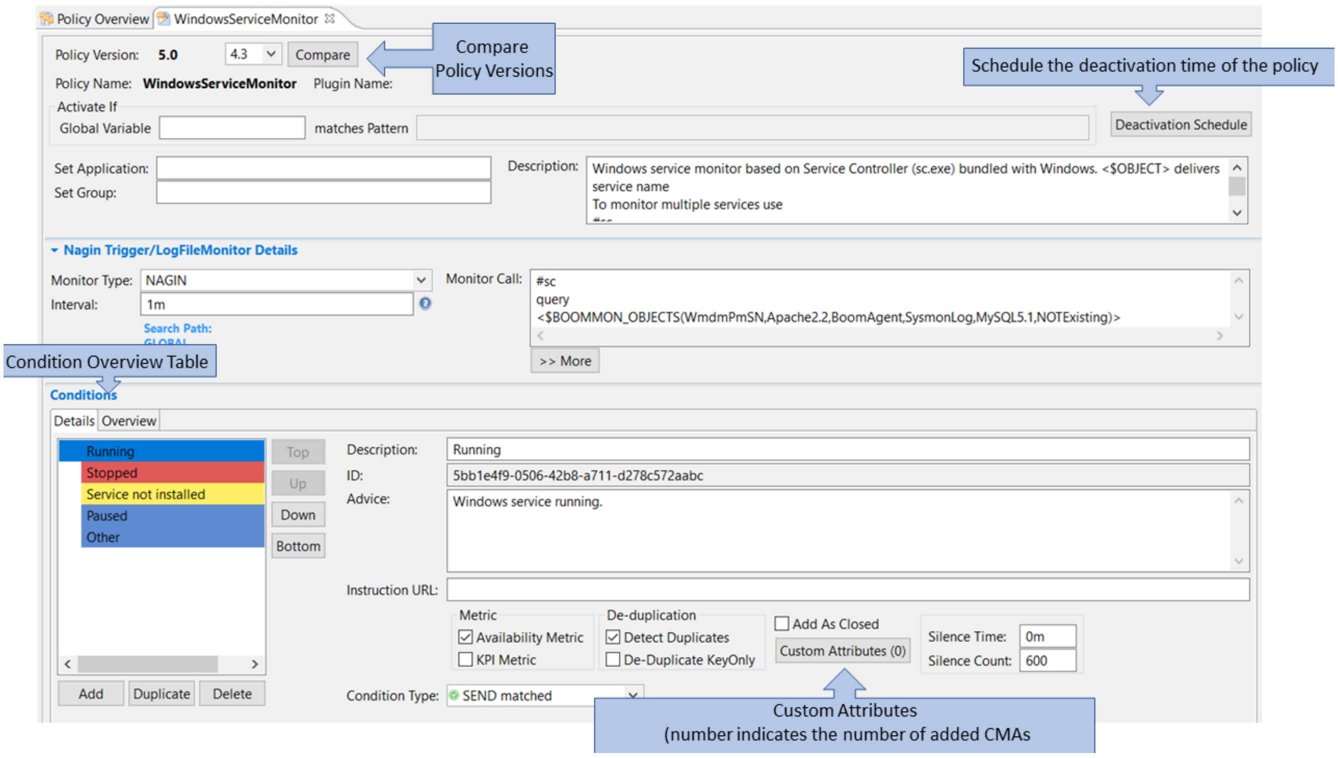
Indication Policy:	An indication policy is used for processing incoming messages. This policy doesn't use the "Nagin Trigger/LogFileMonitor Details" field. Examples: SNMP trap policy, boomindi
Hybrid Policy:	A hybrid policy is used for processing text output from executables or scripts. A hybrid policy can trigger Java based tasks or Nagios® like plugins (NAGINs) for every specified interval. Actually any executable or script that produces an output can be used as trigger. The output will be processed by conditions specified in the policy.

One main difference between these two types is that hybrid policies are **excluded from the global message filtering** and are only used for processing the output of the defined trigger.

An indication policy becomes a hybrid policy when the "Type" field in the "Nagin trigger/LogFileMonitor Details" section is not empty.

For more information about hybrid policies see chapter Hybrid Policy.

6.5.7.2. Indication Policy Details



Policy Version: 5.0 4.3

Policy Name: WindowsServiceMonitor Plugin Name:

Activate If Global Variable matches Pattern

Set Application:

Set Group:

Description: Windows service monitor based on Service Controller (sc.exe) bundled with Windows. <$OBJECT> delivers service name
To monitor multiple services use

Monitor Type: NAGIN

Interval: 1m

Search Path: GLOBAL

Monitor Call: #sc
query
<$BOOMMON_OBJECTS(WmdmPmSN,Apache2.2,BoomAgent,SysmonLog,MySQL5.1,NOTExisting)>

Running
Stopped
Service not installed
Paused
Other

Description: Running

ID: 5bb1e4f9-0506-42b8-a711-d278c572aabc

Advice: Windows service running.

Instruction URL:

Metric: [✓ Availability Metric] KPI Metric

De-duplication: [✓ Detect Duplicates] De-Duplicate KeyOnly

Add As Closed

Silence Time: 0m

Silence Count: 600

Condition Type: SEND matched

Field Description

ℹ	Use CTRL+Space in the policy fields to pop-up the variable list.

Policy Version: This shows the current version of the policy. The field will be automatically updated when changes are saved on the *boom* server. The drop-down menu at the right side displays all previous versions of this policy. The 'Compare' button allows you to review the history of changes that have been made to the the policy.

Compare: With the compare function you can compare the current policy with an older version. More details about the compare function see the chapter Policy Management.

Plugin Name: The Plugin Name is a non-editable field. The name comes from 3d party plug-ins.

Deactivation Schedule: image::images/deactivationschedule.png[]

Indication Name: This is a unique name of the policy (unique across all kind of policies).

Set Application: The default application attribute can be set in the indication policy. It is assigned to all indications that are generated by the indication policy but is overwritten by a value set in the policy conditions.

Set Group: The default indication group attribute can be set in the indication policy. It is assigned to all indications that are generated by the indication policy but is overwritten by a value set in the policy conditions.

Description: A description of the policy

Global Variable: The policy is only activated if the specified global variable matches the specified value. For more information on setting global variables see chapter Supported Variables and Functions.

Nagin Trigger: Mandatory section for the hybrid policies, where the following parameters have to be specified:

trigger type, interval and trigger parameters.

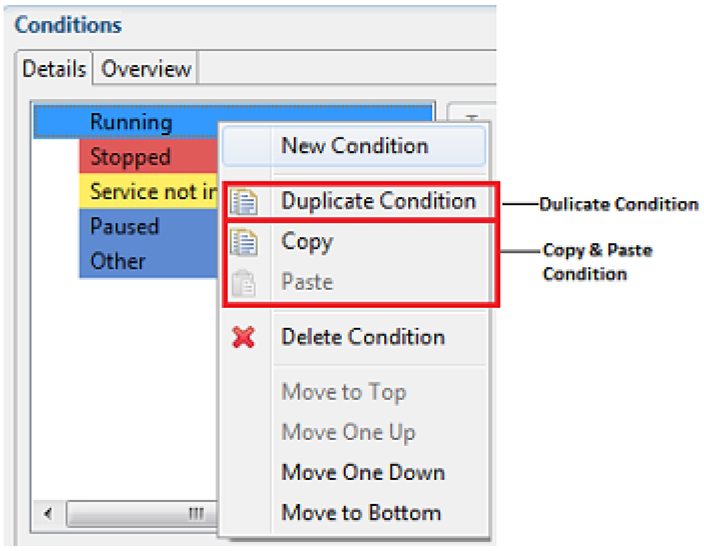
For more information see Chapter Logfile Monitors.

6.5.7.3. Indication Policy Conditions

The Condition section contains a list of filtering rules for processing the incoming text messages.

Field Description

Condition List:



Description: A condition description.

ID: An auto generated ID of the condition.

Advice: An advice message that will be shown in the 'Indication Browser' :: Instruction URL: Define the URL of an external knowledge base e.g. http://myInstruction.server.net.
Agent and indication variables can be used in the instruction URL.

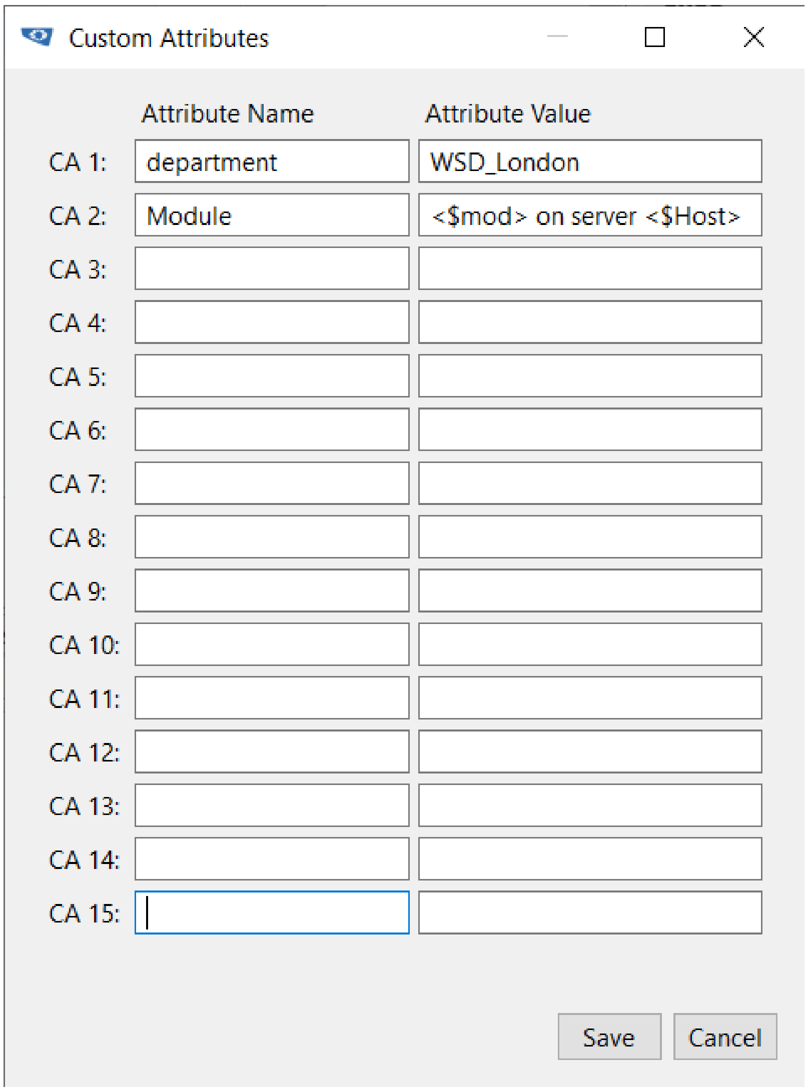
Availability Metric: Defines if the indication that matched with this condition has an impact on the availability of the monitored system.

KPI Metric: This flag defines if the indication must be treated as a 'Key Performance Indicator'.

De-duplication: If the close mask is set in the policy details one of the duplication flags (Detect Duplicates or De-Duplicate KeyOnly) has to be set. The flags cannot be unchecked in the case where the close mask is set.

Detect Duplicates:	This flag notifies the *boom* server to recognize duplicates for active indications and closed indications which were inserted directly as closed (policy flag "add as Closed" is set). By receiving duplicate indications the *boom* server increases the "duplicate count" and updates the "last duplicate time" of the first received indication. No new entry will be created in the database. The following message attributes are used for recognizing duplicates: AGENT_HOST,HOST,APPLICATION,GROUP,OBJECT,SEVERITY,INDICATION KEY, MESSAGE TEXT
De-Duplicate KeyOnly:	This flag notifies the *boom* server to recognize duplicates for active indications and closed indications which were inserted directly as closed (policy flag "add as Closed" is set) based on the defined indication key. By receiving duplicate indications the *boom* server increases the "duplicate count" and updates the "last duplicate time" of the first received indication. No new entry will be created in the database. The indication key attributes are used for recognizing duplicates hence an **Indication Key has to be defined**.
Add As closed:	This will add the message already as a 'closed message' to the indication browser.

Custom Attributes: User defined attributes which can be either just a text or a variable value. CA's are forwarded to the *boom* server as specified, they aren't subject to match on the agent. Up to 15 'Custom Attributes' can be defined in the 'Custom Attributes Dialog'. Names of the custom attribute should not contain '=' sign.
Values can use supported variables.



	Attribute Name	Attribute Value
CA 1:	department	WSD_London
CA 2:	Module	<$mod> on server <$Host>
CA 3:		
CA 4:		
CA 5:		
CA 6:		
CA 7:		
CA 8:		
CA 9:		
CA 10:		
CA 11:		
CA 12:		
CA 13:		
CA 14:		
CA 15:		

Indication Key: A key template that is used for correlation. The indication key contains a list of supported variables which will be resolved during the processing of an incoming value. The result string is used for correlation with following indication for auto acknowledgment and duplicate suppression.

Close Mask: Pattern that is used to search and close previously submitted indications.

Filter Section: Defines filters for application, indication group, object, host and severity.

Important: Pre-Filtering for SNMP Traps:

- Application has to be set to **SNMPTrapd**
- Indication group has to be set to **SNMP**

These fields support only limited subset of the pattern notation.

```
<*>                 matches all
abc                 matches "abc" string
abc | dce | aaa     matches one of the {"abc","dce", "aaa"}
```

Overwrite Attributes: After filtering is done and a match for a condition is successful, the attributes in this section can be set to a desired value to overwrite the incoming value or to define a value if the attribute has not been set. In addition, the "Set Application" and "Set Group" attributes overwrite the **default attributes** for "Set Application" and "Set Group" defined in the general part of the policy.

Set Host: Overwrites the hostname of the submitted monitor value. Default value is an agent hostname. All supported and optional variables can be used in this field as well. In case the monitor submits the hostname as optional variable you can use this variable in the "Set Host" field.

Set Application: Overwrites the application name of the submitted monitor value or defines a value if the attribute has not been set. In addition, overwrites the **default attribute** of "Set Application" if defined.

Set Object: Overwrites the object of the submitted monitor value or defines a value if the attribute has not been set.

Set Group: Overwrites the group of the submitted monitor value or defines a value if the attribute has not been set. In addition, overwrites the default attribute of "Set Group" if defined.

Set Severity: Overwrites the severity of the submitted monitor. Set the severity to critical, major, minor, warning, normal or unknown.

Search Text: Main text filter and parser template.

Match Variables: Variable: <$n> incoming variable (n = number of variable)

Simplified Pattern: *boom* simplified pattern definition for matching the incoming variable (see also chapter Patterns)

These variables are subject to match on the agent, they aren't forwarded to the server. They can be used for checking SNMP trap variables or any other variables e.g. boomindi optional variables.

Set Text: Define the message text that should appear in the browser.

Silence Time:	Suppression time interval since the first match. The default setting for this parameter is 0.
Silence Count:	Suppress number of messages since the first indication has been sent. The default setting for this parameter is 0.
Negation:	A result of matching will be inverted at the end. This flag gives a possibility to create exclude conditions. NEGATION=FALSE (default) the matching result won’t be inverted. NEGATION=TRUE the matching result will be inverted.
Condition Type:	STOP | SEND STOP - means that a incoming message that matched with the current condition must be ignored and all following conditions must not be checked. This helps to increase the performance of the *boom* agent by using a STOP condition at the beginning of the policy. As result the correlation engine doesn’t need to check all following condition and drops all text messages that have no interest. SEND - indicates that the matched text must be send as an indication. The rest of the conditions inside the current policy will be automatically ignored. STOP/SEND aren’t valid for a global MATCHALL condition (all filters = <*>). At least one filter criteria has to be set to work as a normal condition.
Auto Action:	The 'Auto Action' will be automatically triggered by the *boom* server if the condition has matched. Indication variables can be used in the call of the executable i.e. as parameters and will be resolved during the execution. The returned output of the action will be visible as an annotation in the details window of the according indication. Default working directory for actions is $BOOM_ROOT/spi (instrumentation directory of agent) For example a policy which monitors the status of an application process detects that the process isn’t running anymore. The matching condition includes a call of a restart script as an automatic action. The execution is triggered automatically and restarts the process. The output of the script if available will be visible for the operator in the annotation section of the details of the indication. Example: `mail.sh teamA <$OBJECT> <$ORIG_TEXT>` This automatic action will execute a mail script which will send an email to the mail account "teamA" with the resolved indication object as the mail subject and the resolved indication text as the message body.
AA Host:	The target agent hostname for the remote automatic action. If nothing is specified, the action will be triggered on the agent where the indication was created. All indication variables are supported and will be resolved.
AA Timeout:	Timeout in seconds for the remote automatic action. AA Start on duplicate Set the flag to a value N. The AutoAction will be activated after N duplicates.

Operator Action:	This action works in the same way as the 'Automatic Action' but it’s execution will be triggered manually by an operator through a button in the indications detail window. This action is also defined in the policy and its output will be displayed in the annotations section in the indications details window as well. Default working directory for actions is $BOOM_ROOT/spi (instrumentation directory of agent)

6.5.7.4. Nagin Trigger and LogfileMonitor Types

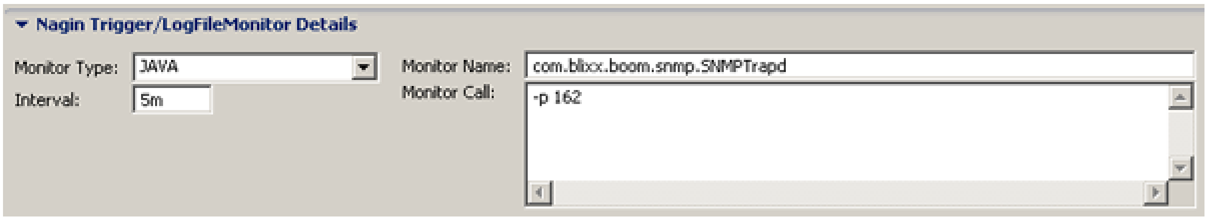
The "NaginTrigger/LogFileMonitor Details" section is mandatory for hybrid policies. You have to specify an interval, some trigger parameters and one of the trigger type:

JAVA, NAGIN, Logfile monitor, Logfile transaction monitor, Logfile transaction monitor(multiline), (MP) Logfile monitor, (MP) Logfile transaction monitor, (MP) Logfile transaction monitor(multiline)

This section is **NOT used** for **SNMP Trap** policies.

For more information see chapter Logfile Monitors.

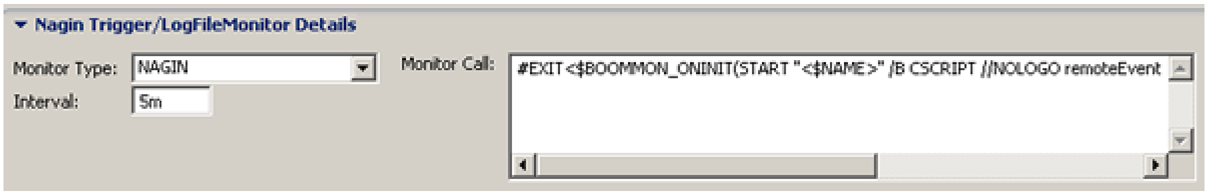
JAVA:



▾ Nagin Trigger/LogFileMonitor Details

Monitor Type: JAVA
Interval: 5m
Monitor Name: com.blixx.boom.snmp.SNMPTrapd
Monitor Call: -p 162

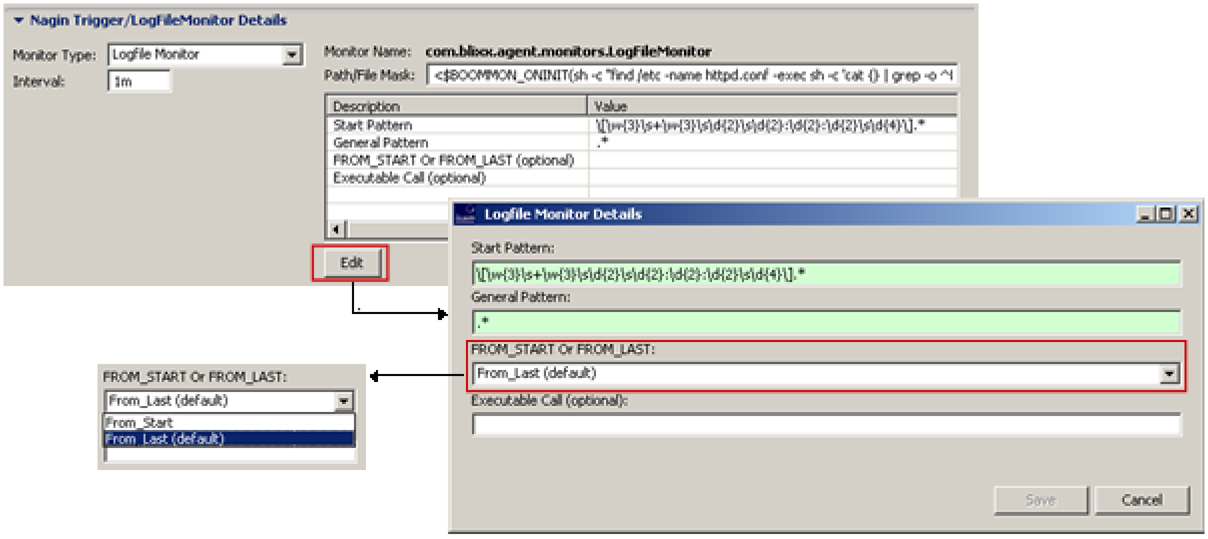
NAGIN:



▾ Nagin Trigger/LogFileMonitor Details

Monitor Type: NAGIN
Interval: 5m
Monitor Call: #EXIT<$BOOMMON_ONINIT(START "<$NAME>" /B CSCRIPT //NOLOGO remoteEvent

Logfile Monitor:



▾ Nagin Trigger/LogFileMonitor Details

Monitor Type: Logfile Monitor
Interval: 1m
Monitor Name: com.blixx.agent.monitors.LogFileMonitor
Path/File Mask: <$BOOMMON_ONINIT(sh -c "find /etc -name httpd.conf -exec sh -c 'cat {} | grep -o ^\

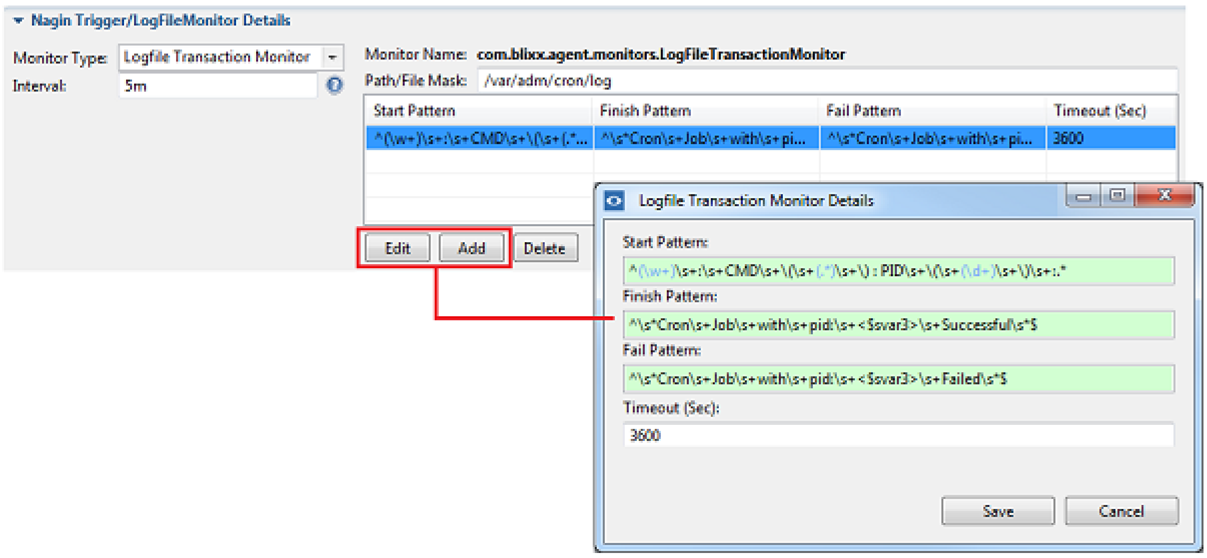
Description	Value
Start Pattern	\[\w{3}\s+\w{3}\s\d{2}\s\d{2}:\d{2}:\d{2}\s\d{4}\].*
General Pattern	.*
FROM_START Or FROM_LAST (optional)	
Executable Call (optional)	

Logfile Monitor Details

Start Pattern: \[\w{3}\s+\w{3}\s\d{2}\s\d{2}:\d{2}:\d{2}\s\d{4}\].*
General Pattern: .*
FROM_START Or FROM_LAST: From_Last (default)
Executable Call (optional):

FROM_START Or FROM_LAST: From_Last (default)
From_Start
From_Last (default)

Logfile Transaction Monitor:


Nagin Trigger/LogFileMonitor Details
Edit
Add
Delete
Logfile Transaction Monitor Details
Save
Cancel

Monitor Type: Logfile Transaction Monitor
Interval: 5m
Monitor Name: com.blixx.agent.monitors.LogFileTransactionMonitor
Path/File Mask: /var/adm/cron/log

Start Pattern	Finish Pattern	Fail Pattern	Timeout (Sec)
^(\w+)\s+:\s+CMD\s+\(\s+(.*...	^\s*Cron\s+Job\s+with\s+pi...	^\s*Cron\s+Job\s+with\s+pi...	3600

Start Pattern:
```
^(\w+)\s+:\s+CMD\s+\(\s+(.*)\s+\) : PID\s+\(\s+(\d+)\s+\)\s+:.*
```
Finish Pattern:
```
^\s*Cron\s+Job\s+with\s+pid:\s+<$var3>\s+Successful\s*$
```
Fail Pattern:
```
^\s*Cron\s+Job\s+with\s+pid:\s+<$var3>\s+Failed\s*$
```
Timeout (Sec): 3600

6.5.7.5. Indication Policy BOOM_Messages

General:

The policy "BOOM_Messages" is of the type indiciation policy and is delivered with the software release. Indication policies are for messages from the following sources:

- SNMP trap
- boomindi/opcmsg
- LogfileNAGIN trigger

Traps and boomindi/opcmsg calls are by default handled by all deployed policies on an agent, whereas logfile and NAGIN messages are only reported to the triggering policy. This means if several policies have conditions, which match one trap or boomindi message, several indication will be generated. Internal server messages are not send over an agent, so no policy is needed.

Conditions:

SUPPRESS all SNMP Traps

This condition suppresses (Condition Type "STOP") all SNMP traps (Indiciation Group "SNMP")

All other agent messages

This condition forwards all agent messages (filter: Application=AGENT; Condition Type "SEND") to the server, which are not matched in any other policy/condition.

Global MATCH_ALL condition

Conditions from the type "all other/ forward unmatched" works **globally** over all policies. This means, that only one indication is generated, if several policies contain such conditions.
The global MATCH_ALL condition has **NO** filter criteria, hence all messages with no own policy/condition are forwarded to the server.

To get rid of unwanted messages, the "all other/ forward unmatched" condition has to be removed. The Condition Type **"STOP"** isn't valid for a MATCH_ALL condition.

For performance reasons, it is recommended, that all SNMP trap policies have a first condition, which suppresses all unwanted traps ("Suppress Unmatched").

6.5.7.6. Indication Policy for SNMP traps

For setting up an indication policy for SNMP traps you have to consider the following points:

The assignment group "SNMP-TrapReceiver" or combination of the "SNMP" package and the "SNMPTrapd_Trigger" policy must be deployed to the specified agent to enable trap processing.

Do not use a SNMPTrapd_Trigger policy for setting up SNMP trap conditions. This policy is ignored during traps processing. By default, the "SNMP-TrapReceiver" assignment group includes "SNMPTraps" policy which contains list of conditions matching most common traps and at last "ALL matched" condition accepting all incoming traps.

Each SNMP traps policy has to use following **required filters** in every condition:

Filter	Value
Application	SNMPTrapd
Indication Group	SNMP

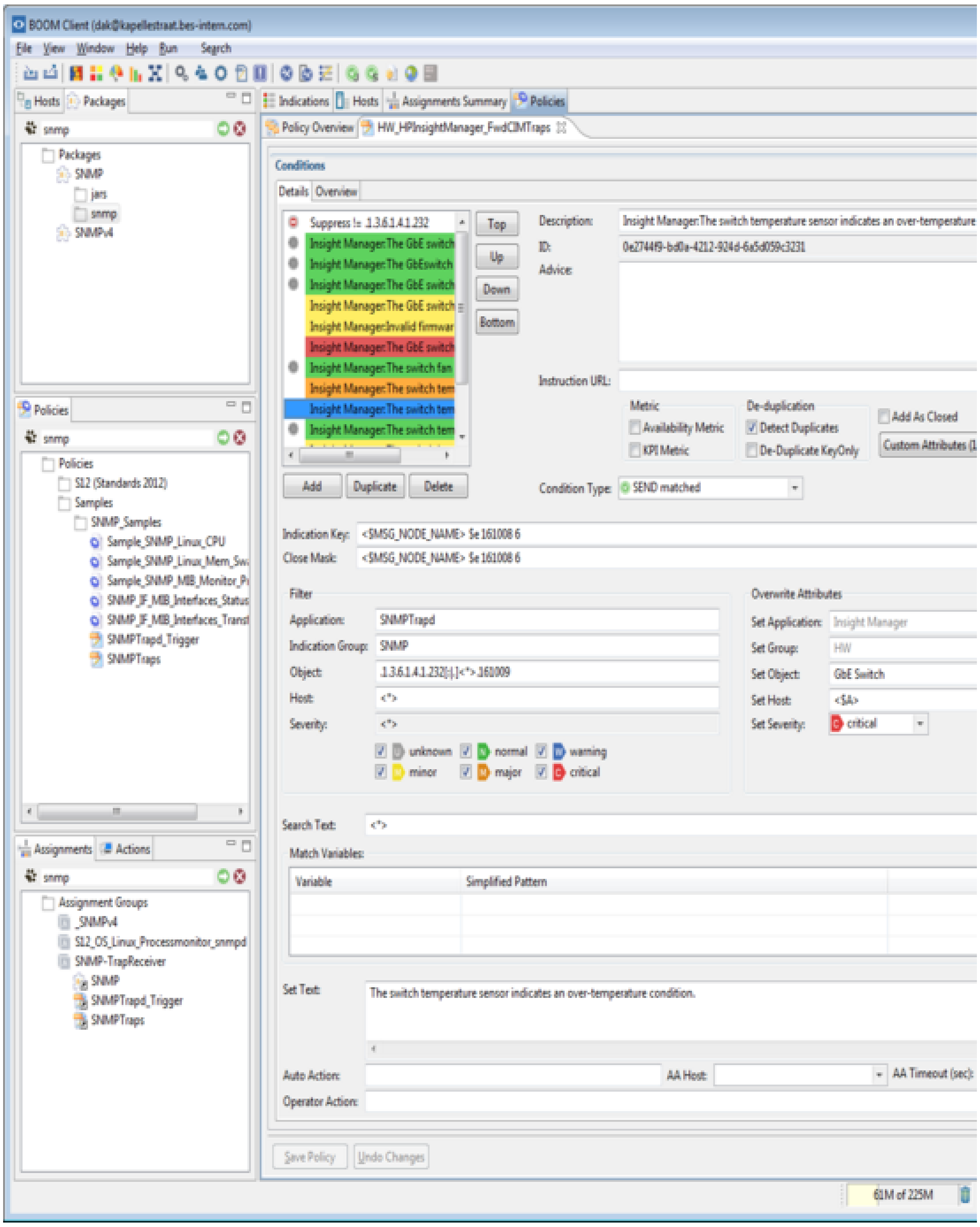
The Object filter used to match the trap OID: **OBJECT**=<**Trap OID**>

<**$1**>**...**<**$n**> or <**$OID1**>**...**<**$OIDn**> are the trap variables.

The **"Search Text"** field can be used for pattern matches on concatenated TrapOID and variables.

Any trap variable can be matched with help of **"Match Variables"** filters.

Use a "stop condition" for non matching traps (e.g. different OID) as first condition. This stops further processing of listed in the policy conditions if incoming trap doesn’t match this condition. In this case the NEGATION flag has to be set to TRUE and Condition Type to STOP.


BOOM Client (dak@kapellestraat.bes-intern.com)
File View Window Help Run Search
Hosts Packages
Indications Hosts Assignments Summary Policies
Policy Overview HW_HPInsightManager_FwdCIMTraps
Conditions
Details Overview
Top Up Down Bottom
Add Duplicate Delete
Policies
Assignments Actions
Save Policy Undo Changes
61M of 225M

Packages: SNMP, jars, snmp, SNMPv4

Policies: S12 (Standards 2012), Samples, SNMP_Samples, Sample_SNMP_Linux_CPU, Sample_SNMP_Linux_Mem_Sw, Sample_SNMP_MIB_Monitor_Pr, SNMP_IF_MIB_Interfaces_Status, SNMP_IF_MIB_Interfaces_Transf, SNMPTrapd_Trigger, SNMPTraps

Assignment Groups: _SNMPv4, S12_OS_Linux_Processmonitor_snmpd, SNMP-TrapReceiver, SNMP, SNMPTrapd_Trigger, SNMPTraps

Conditions:
- Suppress != .1.3.6.1.4.1.232
- Insight Manager: The GbE switch
- Insight Manager: The GbEswitch
- Insight Manager: The GbE switch
- Insight Manager: The GbE switch
- Insight Manager: Invalid firmwar
- Insight Manager: The GbE switch
- Insight Manager: The switch fan
- Insight Manager: The switch tem
- Insight Manager: The switch tem
- Insight Manager: The switch tem

Description: Insight Manager: The switch temperature sensor indicates an over-temperature
ID: 0e2744f9-bd0a-4212-924d-6a5d059c3231
Advice:
Instruction URL:
Metric: Availability Metric, KPI Metric
De-duplication: ✓ Detect Duplicates, De-Duplicate KeyOnly
Add As Closed
Custom Attributes
Condition Type: SEND matched
Indication Key: <$MSG_NODE_NAME> $e 161008 6
Close Mask: <$MSG_NODE_NAME> $e 161008 6

Filter
Application: SNMPTrapd
Indication Group: SNMP
Object: .1.3.6.1.4.1.232[.|.]<*>.161009
Host: <*>
Severity: <*>
✓ unknown ✓ normal ✓ warning ✓ minor ✓ major ✓ critical

Overwrite Attributes
Set Application: Insight Manager
Set Group: HW
Set Object: GbE Switch
Set Host: <$A>
Set Severity: critical

Search Text: <*>
Match Variables:

Variable	Simplified Pattern

Set Text: The switch temperature sensor indicates an over-temperature condition.
Auto Action: AA Host: AA Timeout (sec):
Operator Action:

SNMP V1/V2c/V3 differences

Major difference between SNMP V1 and V2c/V3 is format of traps and addressing, which reflects on Object value of created *boom* indication. SNMP V3 traps are not different from V2c except encryption introduced in V3.

Object

The Object atrribute of incoming *boom* indications for v1 traps will be in the form: <enterpriseOID>:<Generic>.<Specific>. For v2c/v3 traps and informs it will be trap OID.

Text

The Text of incoming *boom* indications indications for v2 traps will include Uptime and .1.3.6.1.6.3.1.1.4.1.0=<trapOID> as first two line in addition to trap variables

i.e. sending V1 vs V2c traps :

```
snmptrap -v 1 -c public localhost '' localhost 6 1 '' SNMPv2-MIB::sysLocation.0 s "Location 001"
snmptrap -v 2c -c public localhost '' .1.3.6.1.4.1.3.1.1.6.1 SNMPv2-MIB::sysLocation.0 s "Location 001"
```

will result in two *boom* indications with following attributes differences:

Attribute	Value from V1 trap	Value from V2c trap
Application	SNMPTrapd	SNMPTrapd
Indication Group	SNMP	SNMP
Object	.1.3.6.1.4.1.3.1.1 **:6.1**	.1.3.6.1.4.1.3.1.1.6.1
Text	.1.3.6.1.2.1.1.6.0=Location 001	.1.3.6.1.2.1.1.3=47 days, 23:44:35.41 .1.3.6.1.6.3.1.1.4.1.0=.1.3.6.1.4.1.3.1.1.6.1 .1.3.6.1.2.1.1.6.0=Location 001

Additional variables defined by *boom* SNMP Trap receiver that can be used:

Variable	Value from V1 trap	Value from V2c trap	Description
<$C>	public	public	SNMP community
<$A>	127.0.0.1	127.0.0.1	snmp agent address
<$o>	.1.3.6.1.4.1.3.1.1.6.1	.1.3.6.1.4.1.3.1.1.6.1	trap OID
<$e>	.1.3.6.1.4.1.3.1.1		enterprise OID for V1
<$T>	414618703 (TimeTicks)	47 days, 23:44:35.41	system uptime
<$G>	6		generic id for V1
<$S>	1		specific id for V1
<$#>	1	1	count of available trap variables
<$1>	Location 001	Location 001	value of 1st variable
<$.1.3.6.1.2.1.1.6.0>	Location 001	Location 001	value by trap variable OID

6.5.7.7. Indication Policy for SNMP gets

An indication policy with trigger is used for fetching strings with SNMP get.

The monitor type is "JAVA", the monitor name is "com.blixx.boom.snmp.SNMPWalkPrinter".

SNMPWalkMonitor supports following parameters:

```
-h <hostIP>
-o <oid of target field>
-c <community>
-v <SNMP version[1,2]>
-p <port>
-t <timeout in seconds>
-r <retries>
```

Object=OID

Text=SNMP value as string

6.5.7.8. Indication Policy for Logfile Monitors

The logfile monitor has flexible possibilities to specify the name of the monitored log file. It supports file masks for finding rotating log files.

For setting up an indication policy for logfile monitoring (monitor type=Logfile Monitor) you have to consider the following points:

Monitor Type:	Set the **"Monitor Type"** in the "Nagin Trigger/LogFile Monitor details" section to **"Logfile Monitor"**.
Interval:	The interval specifies how often logfile is checked.

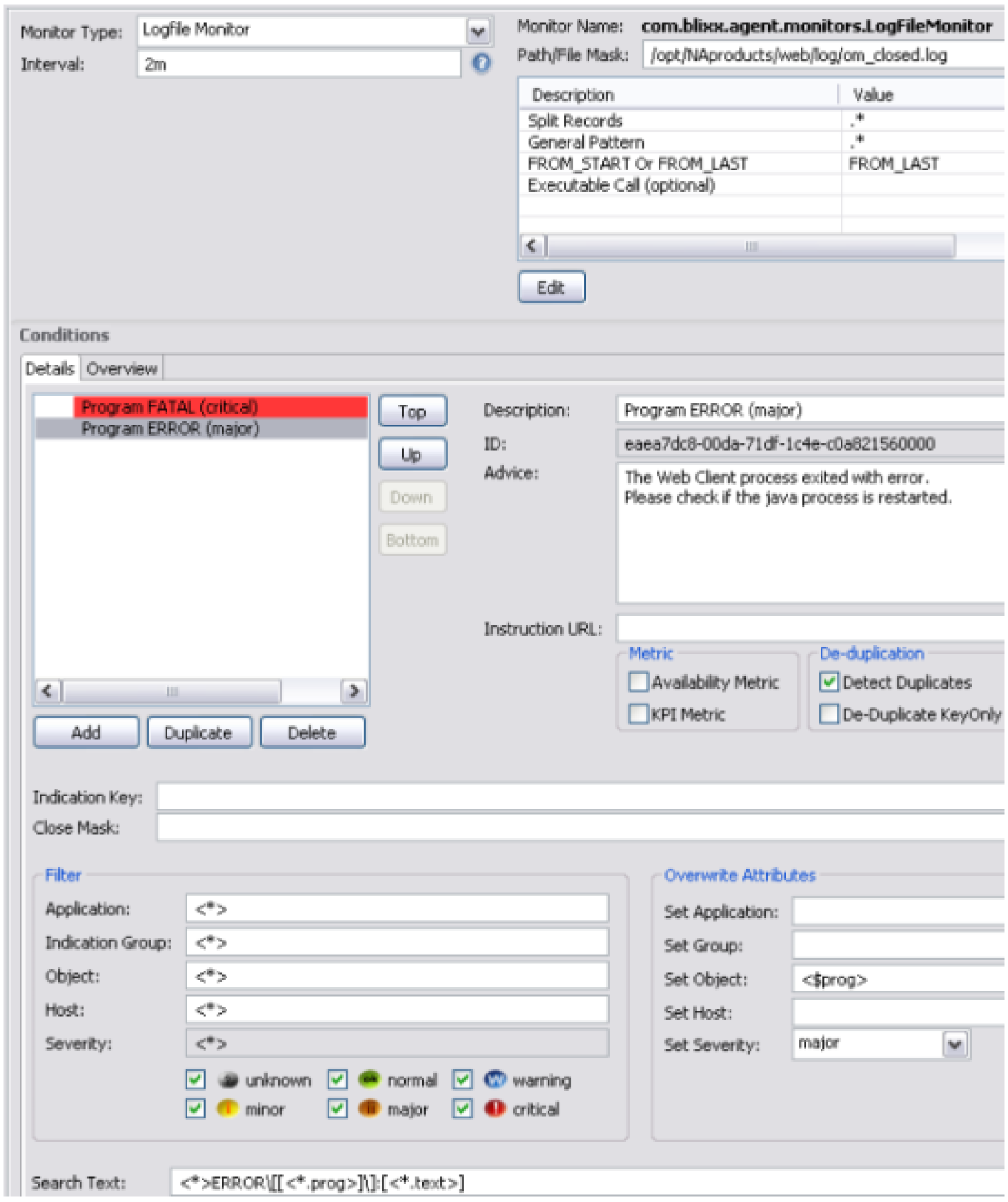
Logfile Path:	The logfile path contains the path of the monitored file or it can contain **a mask for the filename**. The latest file matching the mask will be monitored.
	`e.g. /var/webapps/app1/error_*.log`
	`'*' (star) symbol matches any char sequence within the name. Among all matching files, the file with the latest modification time will be taken.`

Split Records:	Split records allows multi-line processing, because many of the log files contain multi-line entries. It is necessary to have here a Java pattern that is matching the start of the message. Usually it can be a fixed prefix containing the date, severity, log level, etc. If a logfile has no multi-line entries a simple ".*" (match all) pattern can be used.
General Pattern:	A Java pattern as a pre-process filter. This filter allows to reduce the number of messages that must be processed by an agent. pattern ".*" matches all lines - that means all entries will be processed.
FROM_START:	Indicates that the logfile has to be monitored from the begin on every scheduled interval
FROM_LAST:	The logfile will be scanned from the last processed point.
Executable Call:	Allows you to specify a trigger that needs to be executed before parsing the logfile. This can be used together with FROM_START parameter to process logfiles re-generated by trigger on every polling interval. This parameter is optional.

The following optional parameters are coming with every indication:

<***$LOGPATH***>	current log file full path
<***$LOGFILE***>	current log file name

The example shows an indication policy for a simple logfile with single-line entries.



Monitor Type: Logfile Monitor

Interval: 2m

Monitor Name: com.blixx.agent.monitors.LogFileMonitor

Path/File Mask: /opt/NAproducts/web/log/om_closed.log

Description	Value
Split Records	.*
General Pattern	.*
FROM_START Or FROM_LAST	FROM_LAST
Executable Call (optional)	

Program FATAL (critical)
Program ERROR (major)

Description: Program ERROR (major)

ID: eaea7dc8-00da-71df-1c4e-c0a821560000

Advice: The Web Client process exited with error.
Please check if the java process is restarted.

Instruction URL:

Metric: ☐ Availability Metric ☐ KPI Metric

De-duplication: ✓ Detect Duplicates ☐ De-Duplicate KeyOnly

Indication Key:

Close Mask:

Filter

Application: <*>

Indication Group: <*>

Object: <*>

Host: <*>

Severity: <*>

✓ unknown ✓ normal ✓ warning
✓ minor ✓ major ✓ critical

Overwrite Attributes

Set Application:

Set Group:

Set Object: <$prog>

Set Host:

Set Severity: major

Search Text: <*>ERROR\[[<*.prog>]\]:[<*.text>]

6.5.7.9. Indication Policy for Multipath Logfile Monitors

The 'Multipath Logfile Monitors' are extensions of the logfile monitors. The main difference is that the file/path mask is extended to support not only file masks but also path masks. It monitors log file contents for entries of interest. It allows flexible configuration of file rotation policy, specifications of the logfile format and supports pre-processing filter.

The monitor can be configured to monitor a single particular file or a list of logfiles of similar format, or even load a batch list of files to process, provided by an external command.

For setting up an indication policy for multipath logfile monitoring (monitor type=(MP) Logfile Monitor) you have to consider the following points:

Monitor Type:	Set the **"Monitor Type"** in the "Nagin Trigger/LogFile Monitor details" section to **"(MP) Logfile Monitor"**.
Path/FileMask:	The logfile path contains the path of the monitored file.

The path mask syntax is specialized for finding rotating log files. It supports **file and path masks** and consists of the following rules:

- '*' (star) symbol matches any char sequence within path element name. Among all matching files, the file with latest modification time should be taken. This is useful to monitor rotating logfiles, so if logging has switched to another file, the monitor will also pick the newest file (after finishing the previous one).
- Several '*' (star) symbols in different path elements (i.e. on different file tree levels) still comprise the same single group for selecting the newest file.
- '**' (double star) symbol matches any char sequence within path element name. All matching files should be taken, regardless of modification time.
- The path element consisting of exactly double star (like '/**/') specifies recursive scan and matches arbitrary path depth, including zero.
- Double star overrides single star in the same path element, that is all matching files will be taken.
- Path starting from wildcard is considered relative to current dir, unless followed by colon on MS Windows. This is reserved as a special notation for selecting all disk drives on MS Windows: *: (is the same as **:).
- All slashes and backslashes are treated as file (path element) separators, interim double slashes are reduced to single slash. On MS Windows, network paths starting with double (back-)slash are also recognized.
- Several independent patterns can be specified, separated by the '|' (pipe) symbol.

Please see the above section "Logfile Monitors" for an explanation of all other parameters.

The following optional **parameters** are coming with every indication:

<***$PATTERN***>	active pattern
<***$LOGPATH***>	current log file full path
<***$LOGFILE***>	current log file name

Examples:

1. We have the following directory Structure:

   ```
   /web/logs/error1.log
   /web/logs/error2.log
   /web/logs/error3.log
   /web/logs/error4.log
   ```

   Path/File Mask:

   ```
   /web/logs/error*.log
   ```

   Results: Assuming that the most recent file is “/web/logs/error4.log” this file will be monitored.

   Explanation: picks a newest (most lately modified) file whose name starts with 'error' and ends with '.log' in the directory `/web/logs`. This is a typical configuration to monitor a rotated log file, e.g. Apache’s logs.

2. We have the following directory Structure:

   ```
   /app/services/errors.log
   /app/services/warnings.log
   /app/services/information.log
   /app/services/debug.log
   /app/services/exportedfile.txt
   ```

Path/File Mask:

```
/app/services/**.log
```

Results:

```
/app/services/errors.log
/app/services/warnings.log
/app/services/information.log
/app/services/debug.log
```

Explanation: The Path/File mask selects all files with extension `.log` in directory `/app/services`. Unlike the previous example, this pattern assumes no log rotation, and simply selects a bunch of existing files for batch processing.

3. We have the following directory Structure:

```
/var/log/apache/instance1/access_log1
/var/log/apache/instance1/access_log2
/var/log/apache/instance1/access_log3
/var/log/apache/instance1/access_log4
/var/log/apache/instance2/access_log1
/var/log/apache/instance2/access_log2
/var/log/apache/instance2/access_log3
/var/log/apache/instance2/access_log4
/var/log/apache/instance3/access_log1
/var/log/apache/instance3/access_log2
/var/log/apache/instance3/access_log3
/var/log/apache/instance3/access_log4
```

Path/File Mask:

```
/var/log/apache/instance**/access_log*
```

Results: Assuming that the last logfile of every directory is the newest one the following files would be selected: /var/log/apache/instance1/access_log4 /var/log/apache/instance2/access_log4 /var/log/apache/instance3/access_log4

Explanation:
The Path/File mask separates every directory in a virtual group, so 3 groups will be created:

Group1:

```
/var/log/apache/instance1/access_log1
/var/log/apache/instance1/access_log2
/var/log/apache/instance1/access_log3
/var/log/apache/instance1/access_log4
```

Group2:

```
/var/log/apache/instance2/access_log1
/var/log/apache/instance2/access_log2
/var/log/apache/instance2/access_log3
/var/log/apache/instance2/access_log4
```

Group3:

```
/var/log/apache/instance3/access_log1
/var/log/apache/instance3/access_log2
/var/log/apache/instance3/access_log3
/var/log/apache/instance3/access_log4
```

Now from every group the most recent file will be selected leaving us with the files specified in "results".

So the mask `/var/log/apache/instance*/`**`access_log`** in our directory structure would be the same as:

```
/var/log/apache/instance1/access_log*
/var/log/apache/instance2/access_log*
/var/log/apache/instance3/access_log*
```

But the advantage of using "**" is that it is flexible in the number of instances (directories) since they don't have to be entered manually.

4. We have the following directory Structure:

```
/archive/2011-09-15/Maintenance1.log
/archive/2011-09-15/Maintenance2.log
/archive/2011-09-15/Maintenance3.log
/archive/2011-09-15/error.log
/archive/2011-09-16/Maintenance1.log
/archive/2011-09-16/Maintenance2.log
/archive/2011-09-16/Maintenance3.log
/archive/2011-09-16/error.log
/archive/2011-09-17/Maintenance1.log
/archive/2011-09-17/Maintenance2.log
/archive/2011-09-17/Maintenance3.log
/archive/2011-09-17/error.log
```

Path/File Mask:

```
/archive/*/Maintenance*.log
```

Results:
Assuming that the last logfile of every directory is the newest one the following files would be selected:

```
/archive/2011-09-17/Maintenance3.log
```

Explanation: The Path/File mask selects a newest file named like `MaintenanceXXX.log` among all files in direct subdirectories of directory `/archive`. This is a more sophisticated example of log rotation policy, e.g. when logs are grouped per sub-directories by date: `/archive/--/Maintenance-.log`

5. We have the following directory Structure:

```
/var/log/archive/2011-09-15/Maintenance1.log
/var/log/archive/2011-09-15/Maintenance2.log
/var/log/archive/2011-09-15/error.log
/var/log/archive/2011-09-16/Maintenance1.log
/var/log/archive/2011-09-16/Maintenance2.log
/var/log/archive/2011-09-16/error.log
/var/log/archive/2011-09-17/Maintenance1.log
/var/log/archive/2011-09-17/Maintenance2.log
/var/log/archive/2011-09-17/error.log
/var/log/apache/instance1/access_log1
/var/log/apache/instance1/access_log2
/var/log/apache/instance2/access_log1
/var/log/apache/instance2/access_log2
/var/log/apache/instance3/access_log1
/var/log/apache/instance3/access_log2
/var/log/app/services/errors.log
/var/log/app/services/warnings.log
/var/log/app/services/information.log
/var/log/app/services/debug.log
/var/log/app/services/exportedfile.txt
/var/log/web/error1.log
/var/log/web/error2.log
/var/log/web/error3.log
/var/log/web/error4.log
/var/log/test.log
/var/log/error.log
/var/log/messages
```

Path/File Mask:

```
/var/log/**/**.log
```

Results:

```
/var/log/archive/2011-09-15/Maintenance1.log
/var/log/archive/2011-09-15/Maintenance2.log
/var/log/archive/2011-09-15/error.log
/var/log/archive/2011-09-16/Maintenance1.log
/var/log/archive/2011-09-16/Maintenance2.log
/var/log/archive/2011-09-16/error.log
/var/log/archive/2011-09-17/Maintenance1.log
/var/log/archive/2011-09-17/Maintenance2.log
/var/log/archive/2011-09-17/error.log
/var/log/app/services/errors.log
/var/log/app/services/warnings.log
/var/log/app/services/information.log
/var/log/app/services/debug.log
/var/log/web/error1.log
/var/log/web/error2.log
/var/log/web/error3.log
/var/log/web/error4.log
/var/log/test.log
/var/log/error.log
```

Explanation:
The double star "**" will recursively scan every number of directories and subdirectories, including zero (so files directly in "`/var/log`" will be matched two; in this example `/var/log/test.log` and `/var/log/error.log`). The

double star "**" of the files will select every file which ends in "`.log`".

6. We have the following directory Structure:

```
/var/log/archive/2011-09-15/Maintenance1.log
/var/log/archive/2011-09-15/Maintenance2.log
/var/log/archive/2011-09-15/error.log
/var/log/archive/2011-09-16/Maintenance1.log
/var/log/archive/2011-09-16/Maintenance2.log
/var/log/archive/2011-09-16/error.log
/var/log/archive/2011-09-17/Maintenance1.log
/var/log/archive/2011-09-17/Maintenance2.log
/var/log/archive/2011-09-17/error.log
/var/log/apache/instance1/access_log1
/var/log/apache/instance1/access_log2
/var/log/apache/instance2/access_log1
/var/log/apache/instance2/access_log2
/var/log/apache/instance3/access_log1
/var/log/apache/instance3/access_log2
/var/log/app/services/errors.log
/var/log/app/services/warnings.log
/var/log/app/services/information.log
/var/log/app/services/debug.log
/var/log/app/services/exportedfile.txt
/var/log/web/error1.log
/var/log/web/error2.log
/var/log/web/error3.log
/var/log/web/error4.log
/var/log/test.log
/var/log/error.log
/var/log/messages
```

   Path/File Mask:

```
/var/log/**
```

   Results:
   Every file in `/var/log` and subdirectories will be matched

   Explanation: The double star "**" will recursively scan every number of directories and subdirectories and select every file.

6.5.7.10. Policy Specific Logfiles

The policy specific logfile shows information about incoming indications and results of matching/non-matching conditions.

ℹ	Monitor policies will not be taken into account, since the correlation is straight forward.

Enable/disable policy specific logging

The indication policy specific logging will be activated on an agent for a specified set of policies. The server and the policies on the server are not affected by these settings. This means the policy logging is no global setting for the policies, but an **agent** specific setting.

The logging is switched off and on via a **BOOM_AGENTS Action**:

```
    SET_PLOG <ON|OFF>
            <LOGCOUNT>
            <LOGSIZE>
            <polName1> [<polName2> ...]

    LOGCOUNT:   maximum number of logfiles
    LOGSIZE:    maximum size of each logfile in MB
    polName:    policy names
```

The activation flags will be stored in the policy xml file only on the agent side as:

```
        <PLOG_ENABLED>true</PLOG_ENABLED>
        or
        <PLOG_ENABLED>false</PLOG_ENABLED>

        Number of Logfiles allowed:
        <PLOG_LOGCOUNT>3</PLOG_LOGCOUNT>
        Size of each logfile in MB:
        <PLOG_LOGSIZE>1</PLOG_LOGSIZE>
        Default: 3x1MB = 3MB
```

Logfiles and content The logfiles will be written to the following output directory:

```
<boom_agent>/plogs/
```

Logfile names:

```
<policyName>_<date>_<count>.log
```

Logfile content (separated by space):

```
        <date> <result> <conditionUUID> <indi_UUID> <DETAILS>

        Where <result> is 3 characters code and can be one of the following:
                NMA - not matched with any condition
                MSE - matched and sent
                SUT - suppressed by time
                SUC - suppressed by count
                MST - matched and dropped because the condition's type is STOP

        If result is NMA, the condition UUID will be "null"
```

The Logfiles will be limited by size and count of Logfiles.

DETAILS: fields that are coming with indication (incoming values, not transformed by policy)

host@@group@@appl@@object@@sev@@text_as_one_line_all_NEWLINE_chars_converted_to_TAB

In case of match (MSE): DETAILS will be changed to result values! So application, group, object, severity and text can be changed. Indication UUID remains unchanged.

6.5.7.11. Supported variables and Pre-process functions

For more information see chapter Supported Variables and Functions

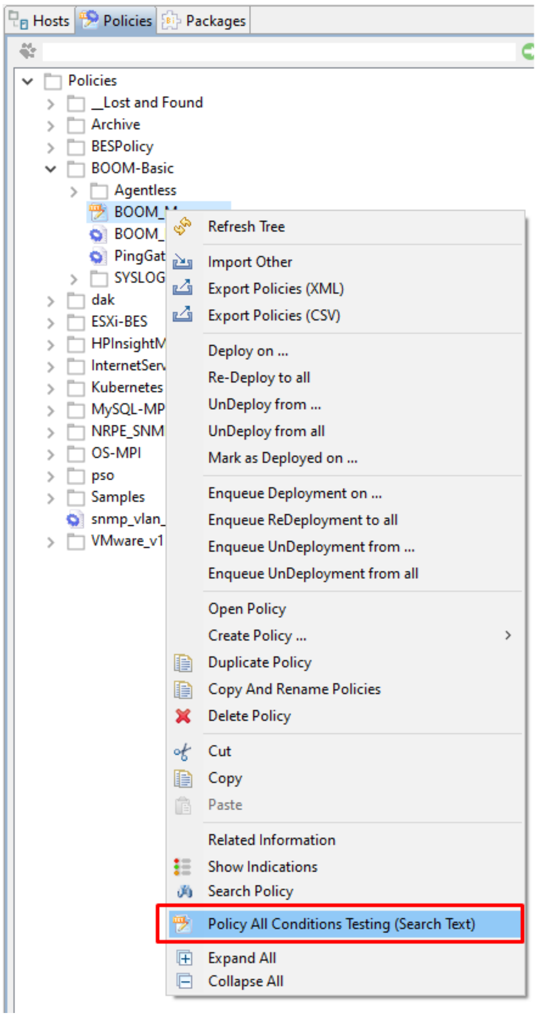
6.5.7.12. Policy All Conditions Testing

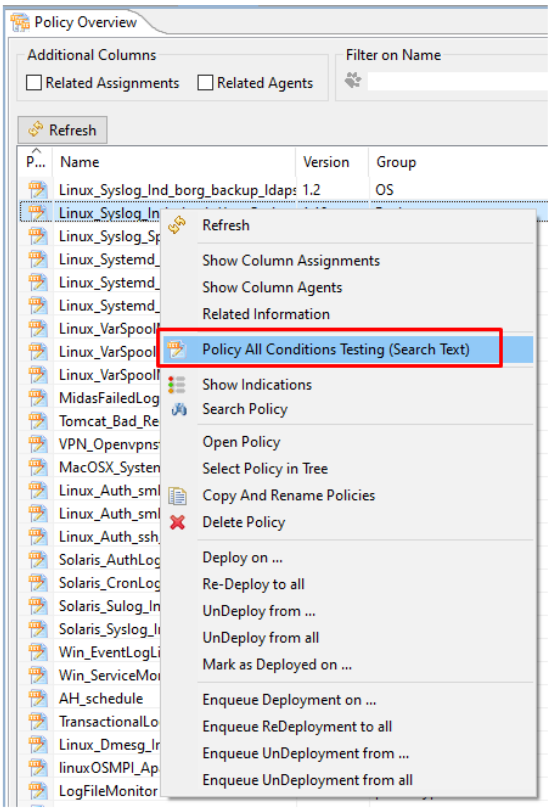
This functionality provides the possibility to test/compare the "Search Text" attribute and the "Object" attribute of all conditions with a defined pattern sequence.

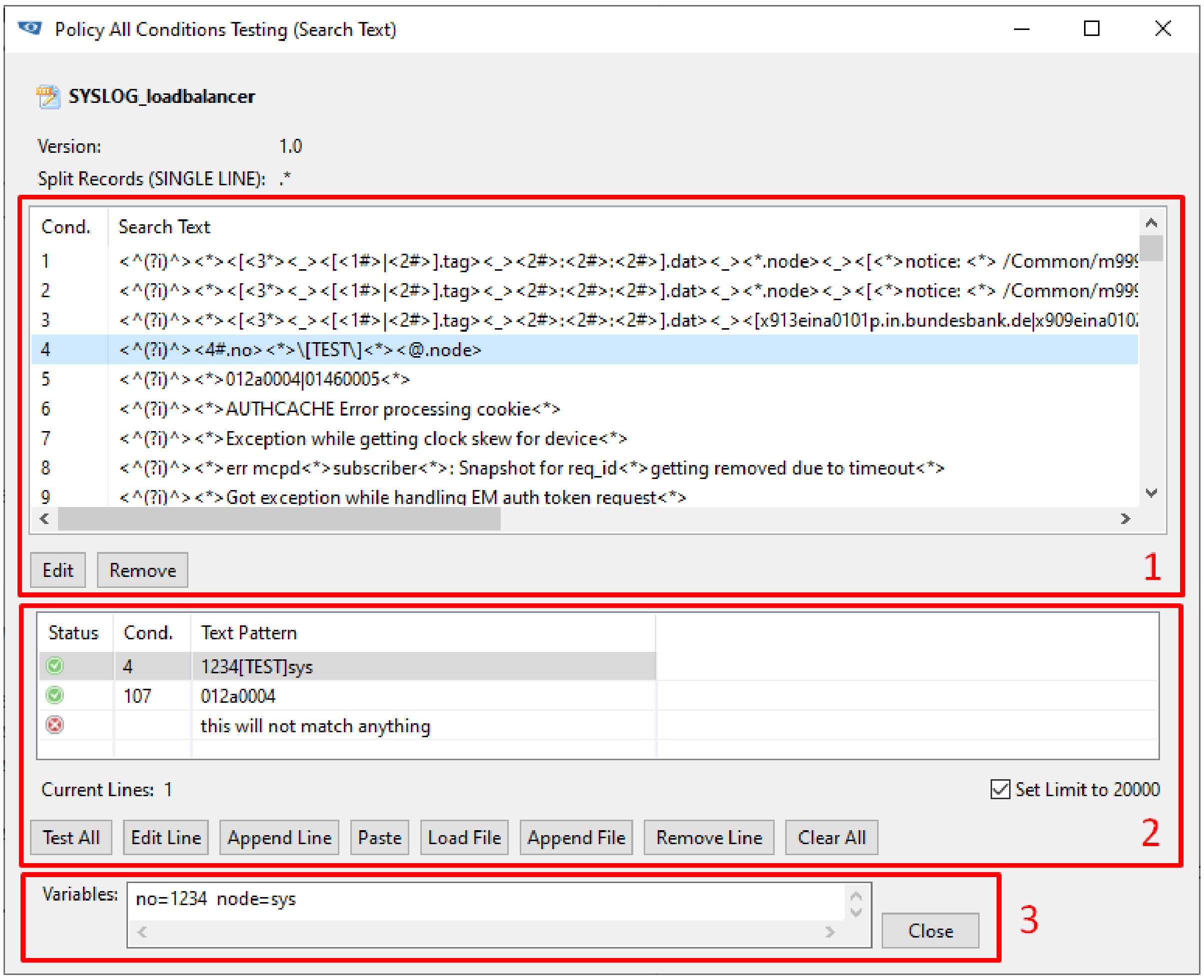
!	- This can be done for Indication Policies only. - Policies with defined "Split Records" (Transaction Logfile "multi-line") can be also tested. - SNMP Policies must have set Application=SNMPTrapd in order to be tested.

Policy All Conditions Testing Dialog

You can open the Dialog from the Policy List View or the Policy Overview Tab:
Right-click on an Indication Policy and select "Policy All Conditions Testing (Search Text)".







Policy All Conditions Testing (Search Text)

SYSLOG_loadbalancer

Version: 1.0

Split Records (SINGLE LINE): .*

Cond.	Search Text
1	<^(?i)^><*><[<3*><_><[<1#>|<2#>].tag><_><2#>:<2#>:<2#>].dat><_><*.node><_><[<*>notice: <*> /Common/m99
2	<^(?i)^><*><[<3*><_><[<1#>|<2#>].tag><_><2#>:<2#>:<2#>].dat><_><*.node><_><[<*>notice: <*> /Common/m99
3	<^(?i)^><*><[<3*><_><[<1#>|<2#>].tag><_><2#>:<2#>:<2#>].dat><_><[x913eina0101p.in.bundesbank.de|x909eina010
4	<^(?i)^><4#.no><*>\[TEST\]<*><@.node>
5	<^(?i)^><*>012a0004|01460005<*>
6	<^(?i)^><*>AUTHCACHE Error processing cookie<*>
7	<^(?i)^><*>Exception while getting clock skew for device<*>
8	<^(?i)^><*>err mcpd<*>subscriber<*>: Snapshot for req_id<*>getting removed due to timeout<*>
9	<^(?i)^><*>Got exception while handling EM auth token request<*>

Status	Cond.	Text Pattern
✓	4	1234[TEST]sys
✓	107	012a0004
✗		this will not match anything

Current Lines: 1

Set Limit to 20000

Variables: no=1234 node=sys

1 Condition List

All conditions are numbered consecutively.

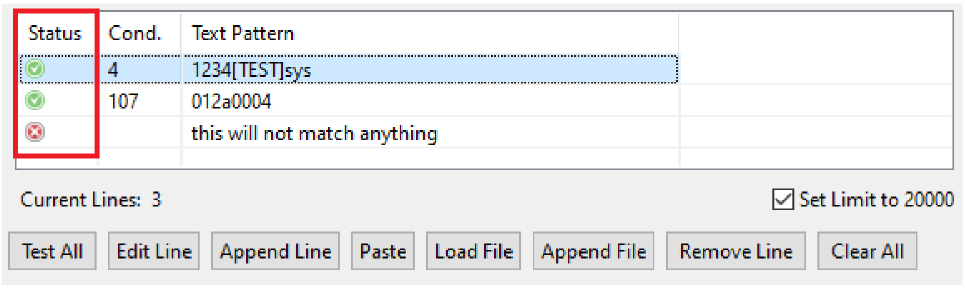
Edit You can edit only one condition at a time. Instead of pressing the Edit button you can also double-click on the condition you want to edit.

Remove You can remove only one condition at a time.

2 Text Pattern Table

Here you have to define the appropriate text pattern you want to run against all conditions in the condition list above.

Column 'Status' This column indicates if the text pattern matches successfully to one or more conditions.



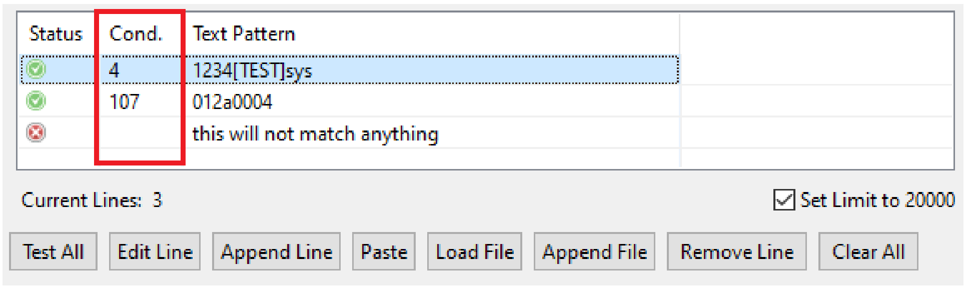
Status	Cond.	Text Pattern
✓	4	1234[TEST]sys
✓	107	012a0004
✗		this will not match anything

Current Lines: 3

Set Limit to 20000

Column 'Cond.'

This Column displays the condition numbers of all matching conditions. Multiple matches are possible and will be displayed as comma separated.
In this example the text pattern *"1234[TEST]sys"* matches exactly to one condition which is condition number 4.



Status	Cond.	Text Pattern
✓	4	1234[TEST]sys
✓	107	012a0004
✗		this will not match anything

Current Lines: 3

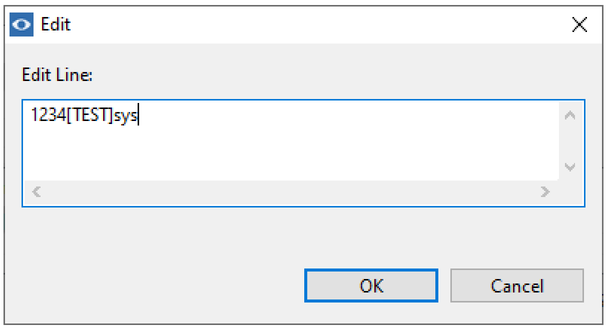
[✓] Set Limit to 20000

Test All

Test all text patterns against the condition list.
The 1. column indicates, if a text pattern matches at least to one of the conditions.

Edit Line

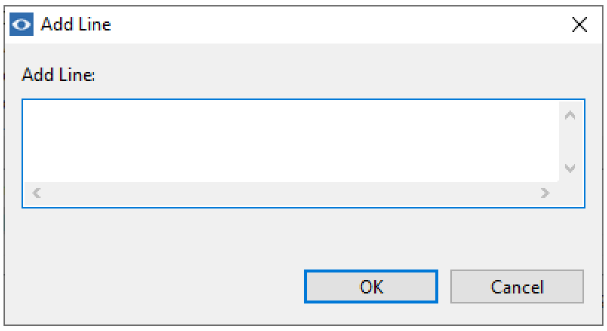
You can edit only one text pattern at a time. The 'all condition testing' process starts immediately after pressing the "OK" button.



1234[TEST]sys

Append Line

Here you can append a singe text pattern. You have the same behaviour as for 'Edit Line': The 'all condition testing' process starts immediately after pressing the "OK" button.



Paste

Paste (append) the data from the clipboard into the text pattern table.

Load from file

This will load the content of a file to the text pattern table.
Please note: This will overwrite all existing lines!

Append from file

Loads and appends the content of a file to the table.
Please note: The existing lines will not be removed! The new data will be appended to the table.

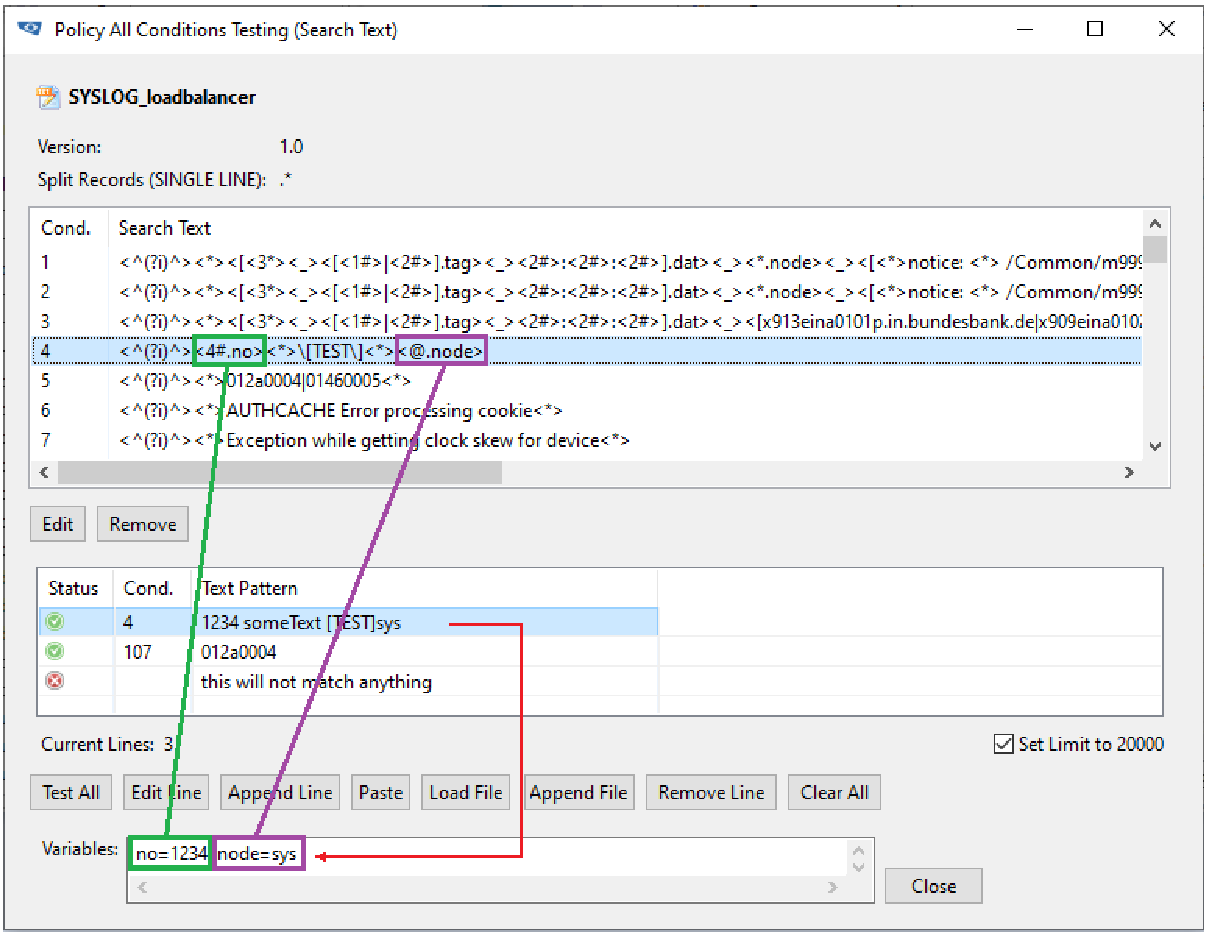
Remove line

Remove selected line(s) from the text pattern table.

Clear All Lines	Remove all text patterns from the table.
Limit number of lines	Due to performance reason you have the possibility to limit the number of lines to 20.000.

3 Capturing Groups

Displays the capturing groups that are defined by the conditions.
The example condition in the screenshot defines 2 capturing groups: <4#.no> and <@.node>


Policy All Conditions Testing (Search Text)

SYSLOG_loadbalancer

Version: 1.0
Split Records (SINGLE LINE): .*

Cond.	Search Text
1	<^(?i)^><*><[<3*><_><[<1#>|<2#>].tag><_><2#>:<2#>:<2#>].dat><_><*.node><_><[<*>notice: <*> /Common/m99
2	<^(?i)^><*><[<3*><_><[<1#>|<2#>].tag><_><2#>:<2#>:<2#>].dat><_><*.node><_><[<*>notice: <*> /Common/m99
3	<^(?i)^><*><[<3*><_><[<1#>|<2#>].tag><_><2#>:<2#>:<2#>].dat><_><[x913eina0101p.in.bundesbank.de|x909eina010
4	<^(?i)^><4#.no><*>\[TEST\]<*><@.node>
5	<^(?i)^><*>012a0004|01460005<*>
6	<^(?i)^><*>AUTHCACHE Error processing cookie<*>
7	<^(?i)^><*>Exception while getting clock skew for device<*>

Status	Cond.	Text Pattern
✓	4	1234 someText [TEST]sys
✓	107	012a0004
✗		this will not match anything

Current Lines: 3
[✓] Set Limit to 20000

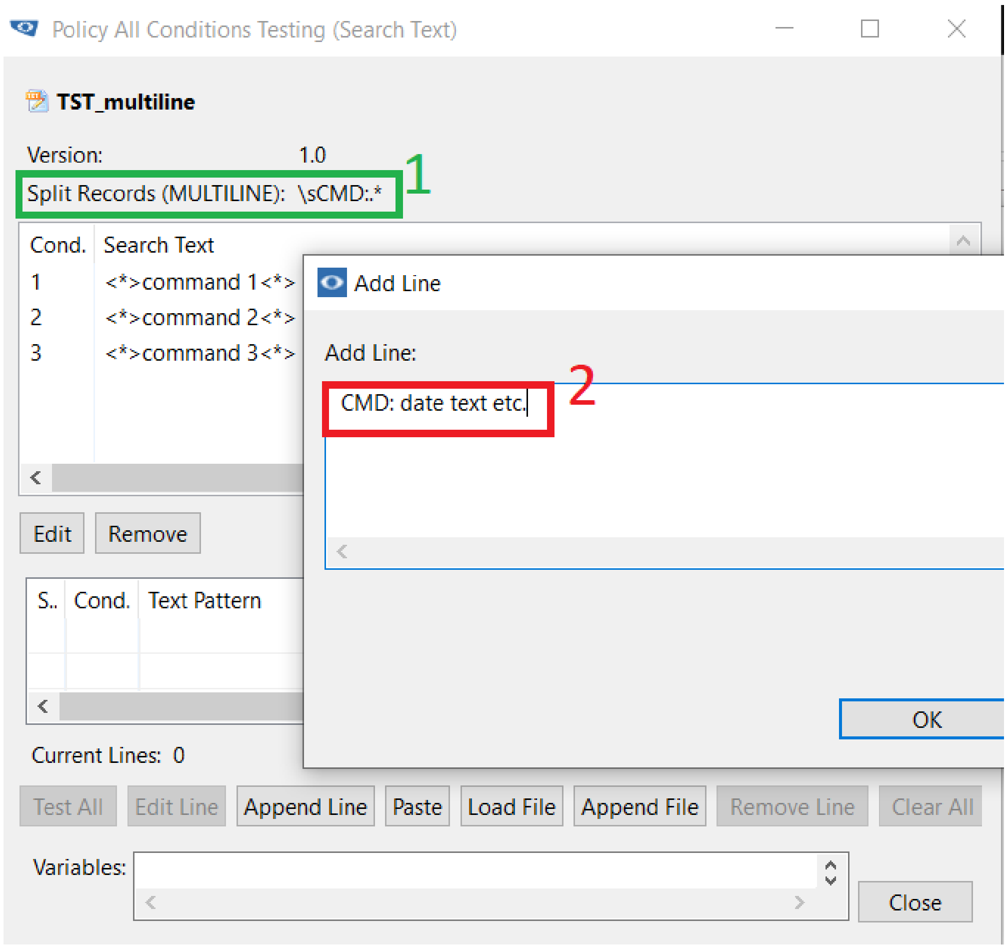
Variables: no=1234 node=sys

Split Records (Transaction Logfile "multi-line")

"Split Records" means that a Java pattern is matching the start of a message or log entry. More information abut split records can be found here Logfile Transaction Monitor (multiline).
For more information about Multiline Logfile and Split-Records see chapter Logfile Transaction Monitor (multiline).

Example

- The Indication Policy has the **"Split Records (MULTILINE)"** definition (1) set to: **\sCMD:.***.
  Syntax configuration "<space>CMD:<any further text>"


Policy All Conditions Testing (Search Text)

TST_multiline

Version: 1.0
Split Records (MULTILINE): \sCMD:.*

Cond.	Search Text
1	<*>command 1<*>
2	<*>command 2<*>
3	<*>command 3<*>

Add Line

Add Line:

CMD: date text etc

Current Lines: 0

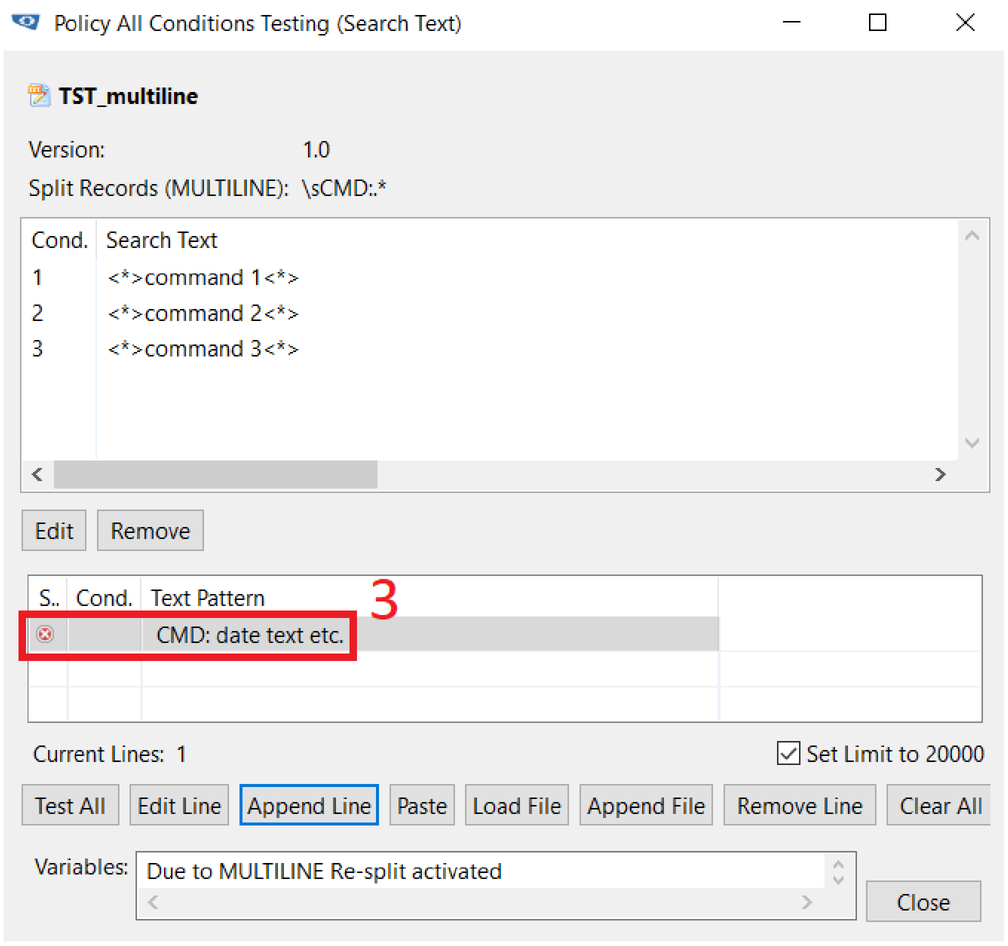
Variables:

- **Append Text Pattern**

  Adding the following text pattern (2): **" CMD: date text etc"**

- **Test Result**

  the result fails (3), because the text pattern (2) does not match any of the conditions.



TST_multiline

Version: 1.0
Split Records (MULTILINE): \sCMD:.*

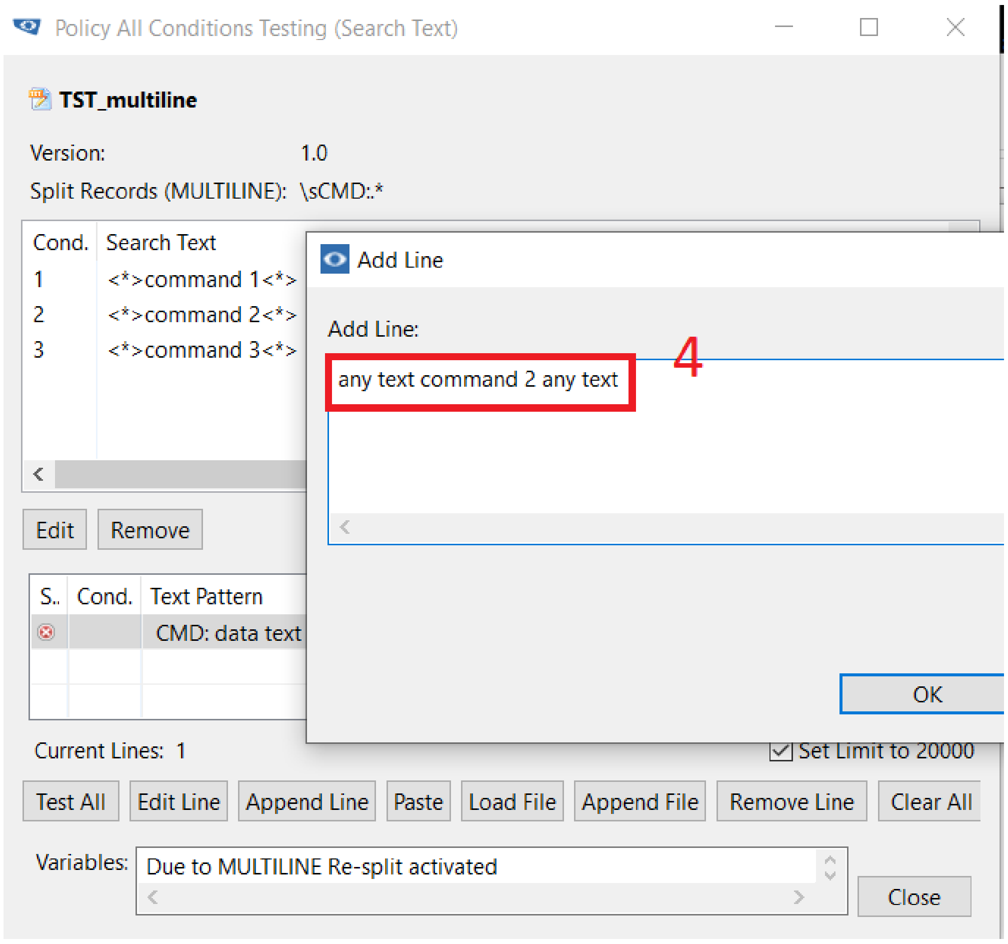
Cond.	Search Text
1	<*>command 1<*>
2	<*>command 2<*>
3	<*>command 3<*>

S..	Cond.	Text Pattern
⊗		CMD: date text etc.

Current Lines: 1
☑ Set Limit to 20000

Variables: Due to MULTILINE Re-split activated

- **Append one more Text Pattern**
  Append another text pattern (4): **"any text command 2 any text"**


Policy All Conditions Testing (Search Text)

TST_multiline

Version: 1.0

Split Records (MULTILINE): \sCMD:.*

Cond.	Search Text
1	<*>command 1<*>
2	<*>command 2<*>
3	<*>command 3<*>

S..	Cond.	Text Pattern
⊗		CMD: data text

Current Lines: 1

[✓] Set Limit to 20000

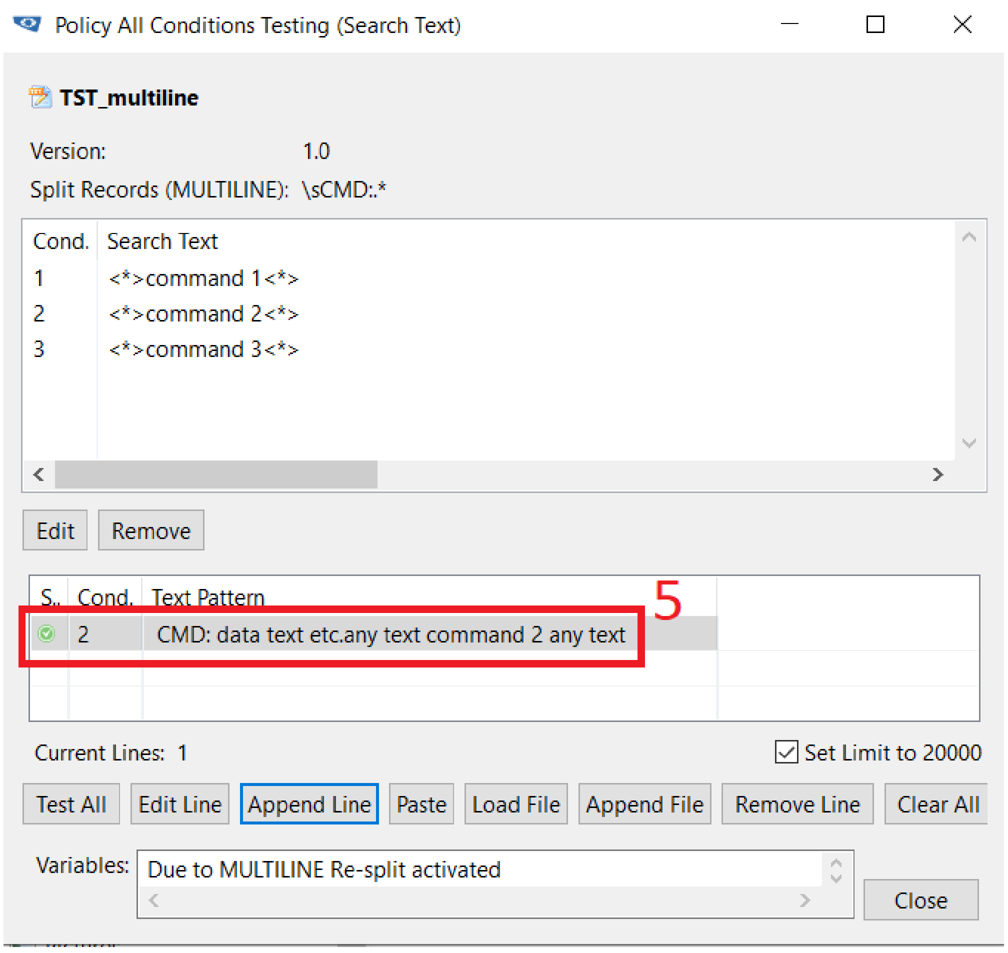
Variables: Due to MULTILINE Re-split activated

Add Line

Add Line:

any text command 2 any text

- **Test Result**
  Because the just appended text pattern has the format "<*>command 2<*>", it will match the 2. condition. See the green success symbol (1. column) and the condition number 2 (2. column) in the test result table (5).

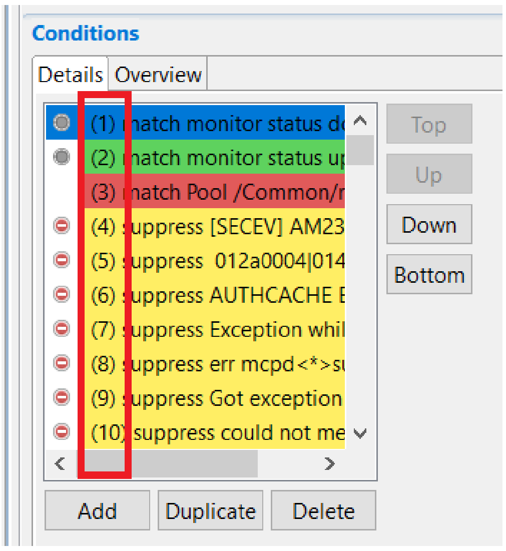


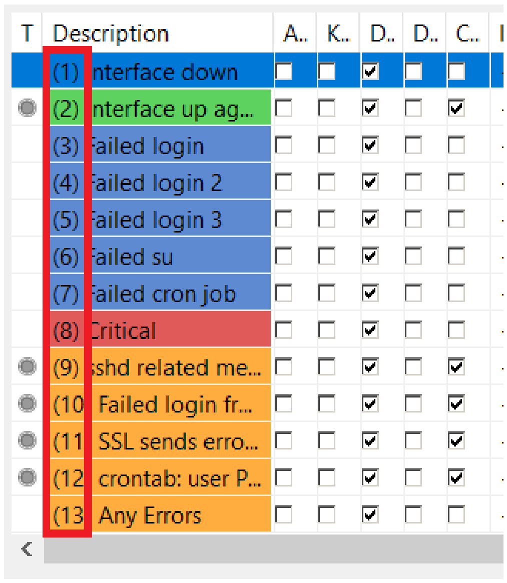
Cond.	Search Text
1	<*>command 1<*>
2	<*>command 2<*>
3	<*>command 3<*>

S..	Cond.	Text Pattern
✓	2	CMD: data text etc.any text command 2 any text

Impact to Indication Policy Condition listing

Within the Indication Policy Details View, both condition lists inside the 'Details' and 'Overview' Tab contains a consecutive numbering of conditions.
Please Note: The numbering is for display only and will NOT be saved to the database. The description itself remains unchanged.


Conditions
Details
Overview
(1) match monitor status d
(2) match monitor status u
(3) match Pool /Common/r
(4) suppress [SECEV] AM23
(5) suppress 012a0004|014
(6) suppress AUTHCACHE E
(7) suppress Exception whil
(8) suppress err mcpd<*>s
(9) suppress Got exception
(10) suppress could not me
Top
Up
Down
Bottom
Add
Duplicate
Delete



T	Description	A..	K..	D..	D..	C..	I
	(1) nterface down	☐	☐	☑	☐	☐	
●	(2) nterface up ag...	☐	☐	☑	☐	☑	
	(3) ailed login	☐	☐	☑	☐	☐	
	(4) ailed login 2	☐	☐	☑	☐	☐	
	(5) ailed login 3	☐	☐	☑	☐	☐	
	(6) ailed su	☐	☐	☑	☐	☐	
	(7) ailed cron job	☐	☐	☑	☐	☐	
	(8) Critical	☐	☐	☑	☐	☐	
●	(9) sshd related me...	☐	☐	☑	☐	☑	
●	(10 Failed login fr...	☐	☐	☑	☐	☑	
●	(11 SSL sends erro...	☐	☐	☑	☐	☑	
●	(12 crontab: user P...	☐	☐	☑	☐	☑	
	(13 Any Errors	☐	☐	☑	☐	☐	

6.5.8. Indication Attribute Sizes

The *boom* server and agent truncates the following indication attributes to a maximum size of characters.

AttributeName	MaxSize (characters)	Server truncates	Agent truncates
Agent Host	255	x	-
Application	1024	x	x
AutoAction	1024	x	x
AA Host	255	x	x
Indication Key	1024	x	x
Group	1024	x	x
Close Mask	1024	x	x
Text	2048	x	x
Host	255	x	x
Object	1024	x	x
Operator Action	1024	x	x
Source	1024	x	-
Customized Attributes	1024	x	-

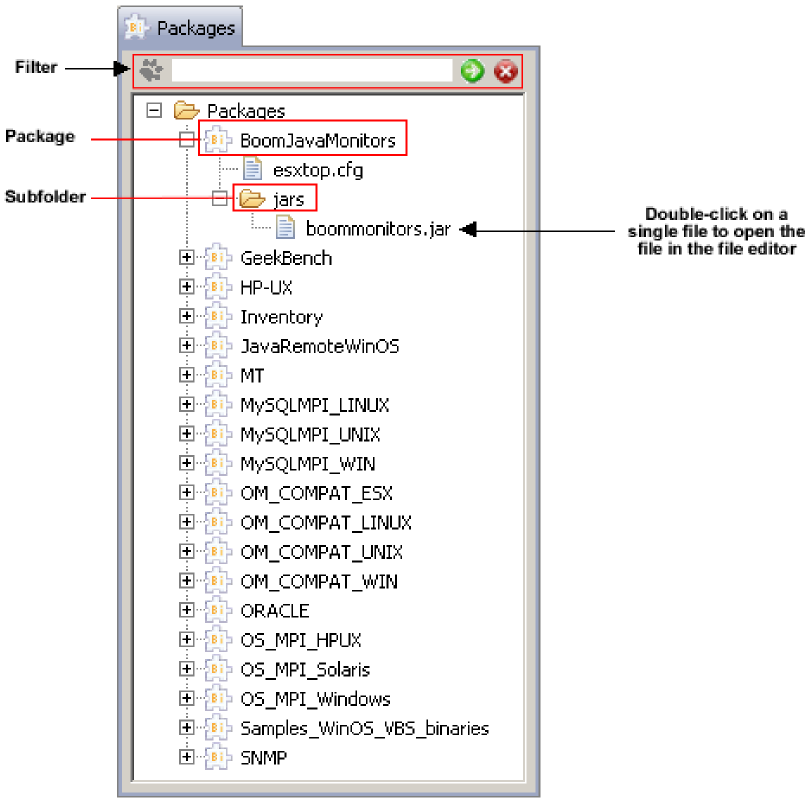
All numeric attributes are not truncated.

Please note, that character size is NOT EQUAL size in bytes! For UTF-8 it can be up to 4 bytes per character.

6.6. Packages

6.6.1. General Information

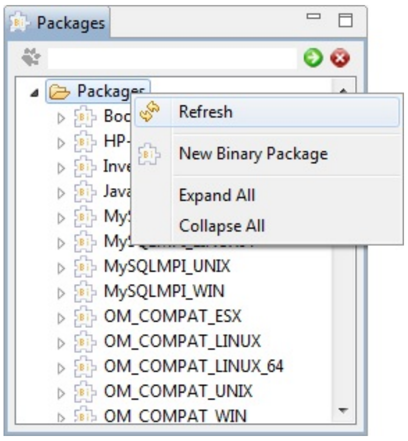
The *boom* workbench allows to maintain the binary packages without direct access to the boomserver. The binary tree reflects the physical file structure on the *boom* server (srv/packages):



6.6.2. Package Operations

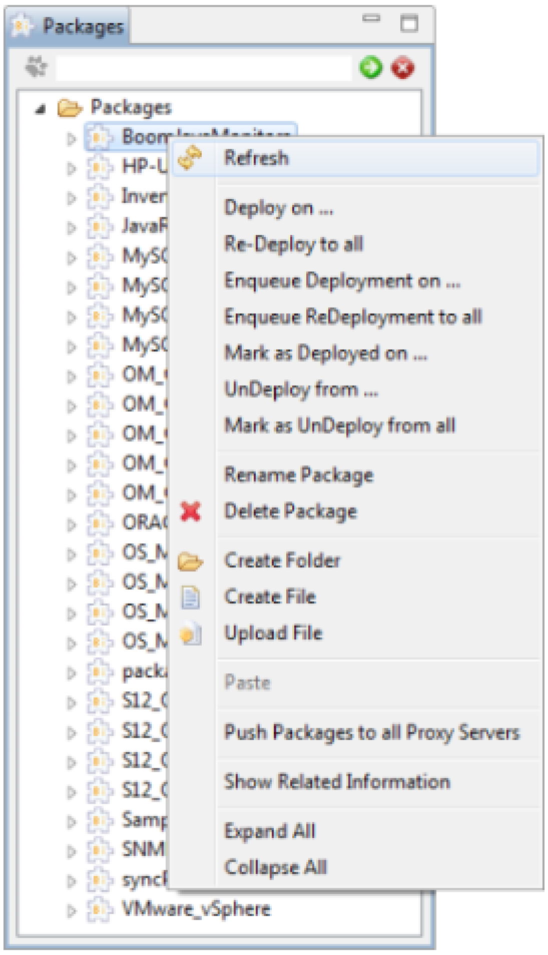
Right-click on the Package List opens the following context menu:

Context menu on top folder level



New Binary Package:	Create a new directory on the *boom* Server under `srv/packages/`. and put all necessary files into the new directory. Changes on the server file system automatically reflect on the *boom* Workbench.

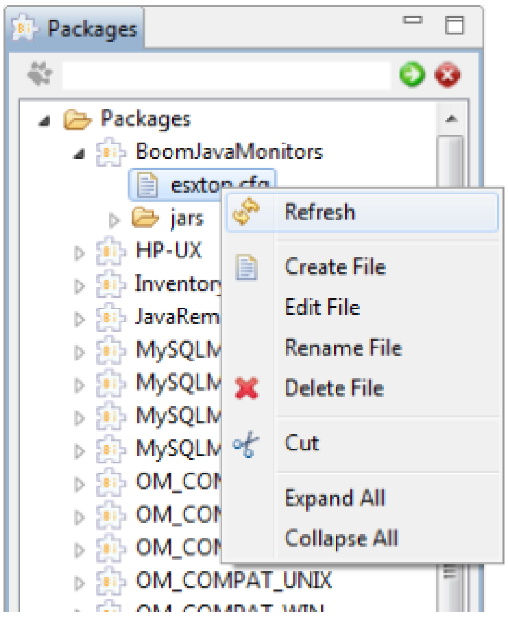
Context menu on Package level



Deploy on:	Allows to deploy a package to one or more *boom* agents. Deployment of a single file is not supported. A binary package can be deployed as single package or as part of an assignment group. For more details see also chapter Assignments.
Re-Deploy to all:	Allows to trigger re-deployment of the selected package to all boomagents where they were previously deployed.
Enqueue Deployment on...:	This is the same action like 'Deploy on' except that the deployment will be NOT started but added to the 'Deployment Queue'. For more details see chapter Assignments Summary.
Enqueue ReDeployment to all:	This is the same action like 'Redeploy to all' except that the deployment will be NOT started but added to the 'Deployment Queue'. For more details see chapter Assignments Summary.
Mark as Deployed on ...	
UnDeploy from ...	Un-deploys the selected Assignments from the appropriate agent.
Mark as Undeploy from all	
Rename Package:	Renaming a package will also update the 'Assignment List'. Please Note: If a package is renamed directly on the file system, it will cause problems on the assignment tree and the stored assignments.
Delete Package:	Deletes a package.

Create Folder:	Creates a new folder inside the package.
Create File:	Opens a new file in the file editor.
Upload File:	Allows you to add an existing file stored on the file system to a package. The selected file will be uploaded to the `<boom_install_dir>/srv/packages/BoomJavaMonitors/`
Paste:	Push package to all Proxy Servers: Copies the physical package files to all Proxy servers.
Show Related Information:	Shows a list of agents on which this package is deployed on and a list of assignments to which this package is linked to.

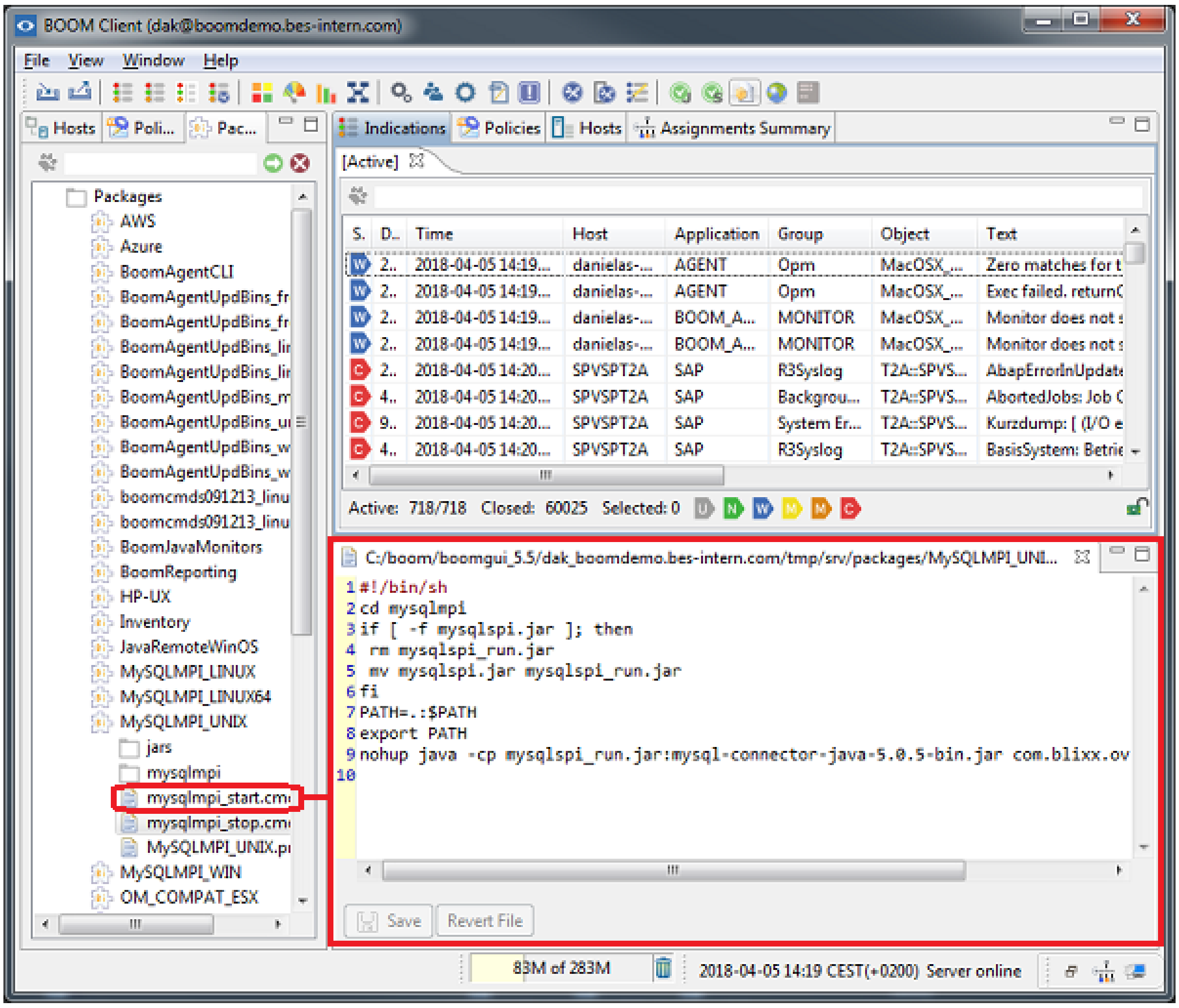
Context menu on file level:



Create File:	Opens a new file in the file editor.
Edit File:	Opens the selected file in the file editor.
Rename File:	Renames the physical file on the server system.
Delete File:	This will delete the physical file from the server system.
Cut/Paste:	Allows you to move a single or multiple packages to another package group.

6.6.3. File Editor

The integrated file editor is a simple text editor that allows you to edit server side files. To edit an existing file just double-click on the file to open it in the file editor. If you select "Create File" from the context menu, a new file will be created and opened in the file editor:



S.	D..	Time	Host	Application	Group	Object	Text
W	2..	2018-04-05 14:19...	danielas-...	AGENT	Opm	MacOSX_...	Zero matches for t
W	2..	2018-04-05 14:19...	danielas-...	AGENT	Opm	MacOSX_...	Exec failed. return(
W	2..	2018-04-05 14:19...	danielas-...	BOOM_A...	MONITOR	MacOSX_...	Monitor does not s
W	2..	2018-04-05 14:19...	danielas-...	BOOM_A...	MONITOR	MacOSX_...	Monitor does not s
C	2..	2018-04-05 14:20...	SPVSPT2A	SAP	R3Syslog	T2A::SPVS...	AbapErrorInUpdate
C	4..	2018-04-05 14:20...	SPVSPT2A	SAP	Backgrou...	T2A::SPVS...	AbortedJobs: Job (
C	9..	2018-04-05 14:20...	SPVSPT2A	SAP	System Er...	T2A::SPVS...	Kurzdump: [(I/O e
C	4..	2018-04-05 14:20...	SPVSPT2A	SAP	R3Syslog	T2A::SPVS...	BasisSystem: Betrie

Active: 718/718 Closed: 60025 Selected: 0

C:/boom/boomgui_5.5/dak_boomdemo.bes-intern.com/tmp/srv/packages/MySQLMPI_UNI...

```
#!/bin/sh
cd mysqlmpi
if [ -f mysqlspi.jar ]; then
 rm mysqlspi_run.jar
 mv mysqlspi.jar mysqlspi_run.jar
fi
PATH=.:$PATH
export PATH
nohup java -cp mysqlspi_run.jar:mysql-connector-java-5.0.5-bin.jar com.blixx.ov
```

83M of 283M   2018-04-05 14:19 CEST(+0200) Server online

Integrated File Editor context menu:

Right-click in the Integrated File Editor window opens the context menu:

Undo Typing	Cancel the previous typing
Revert File	Go back to the original file content
Save	Save all modifications to file
Cut	Cut selected part
Copy	Copy selected part
Paste	Paste selected part
Find/Replace	Find/Replace the specified term

6.6.4. Inventory Package

During startup or runtime the agent checks the inventory data automatically every 24 hours and in case of differences from the previous state it sends a message to the server. The *boom* server automatically fetches the newest inventory data from the agent.

If the Inventory package is **not deployed**, the agent reports back only a **minimal set of inventory data** like:

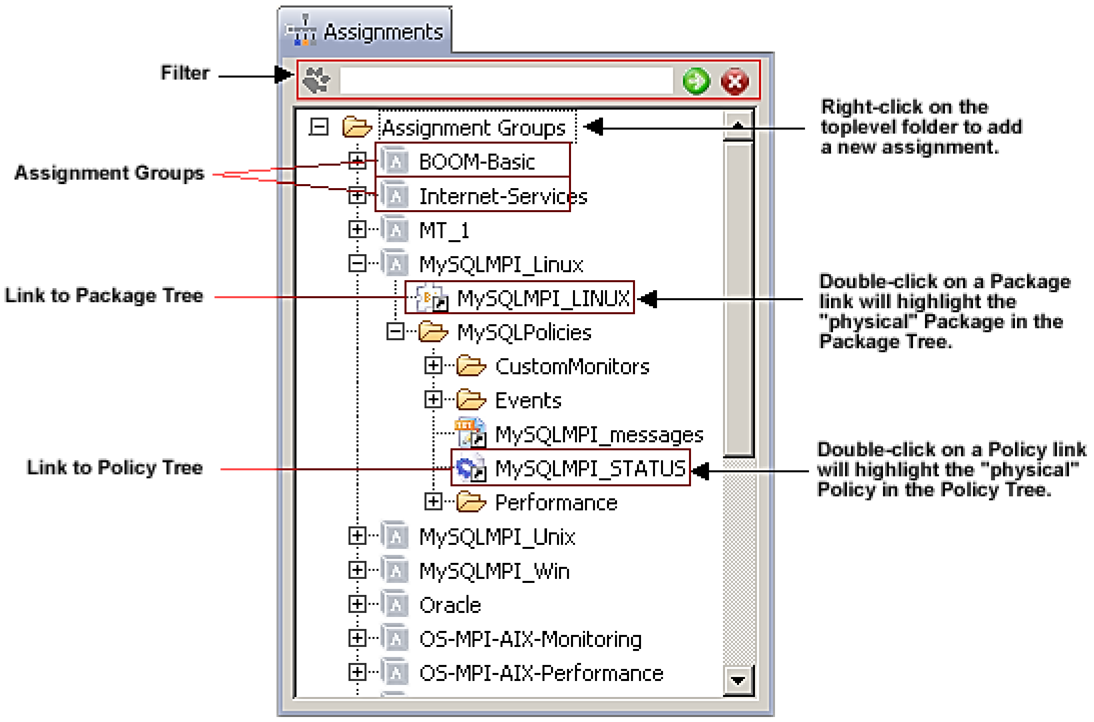
ID	agent ID
VERSION	agent version
HOSTNAME	agent hostname
IP	agent IP address
OS.NAME	OS name (i.e."Windows 7")
OS.VERSION	OS version (i.e."6.1")
OS.ARCH	Architecture of the system (i.e. "amd64")
HW.CPUN	number of CPUs
JAVA.HOME	home directory of JVM running *boom* agent
JAVA.VMNAME	Java virtual machine implementation name
JAVA.VMVENDOR	JVM vendor name
JAVA.VMVER	JVM version

Deploy the Inventory package only if you need more **detailed information** about the system e.g. hard disks, network cards ...

6.7. Assignments

6.7.1. General Information

Assignment Groups are used to group policies and packages together as needed. They can be grouped i.e. by operating systems, test and productive systems and so on. The assignment tree holds all user specified assignment groups. An assignment group can contain subfolders, links to monitor/indication policies or links to packages.



For full agent functionality, deploy the assignment group **"BOOM"** or **"BOOM_Basics"** to the agent, which contains the policy "BOOM_Messages" for internal messages and the package "BoomJavaMonitors" for agent specific Java classes.

6.7.2. Add Policy/Package Links to an Assignment

Add Policy link:

Links can be added to an assignment group by drag & drop a policy from the policy List to an assignment group. You can select one or one or more policies or even a complete policy group and drag & drop them to the assignment group.

Add Package link:

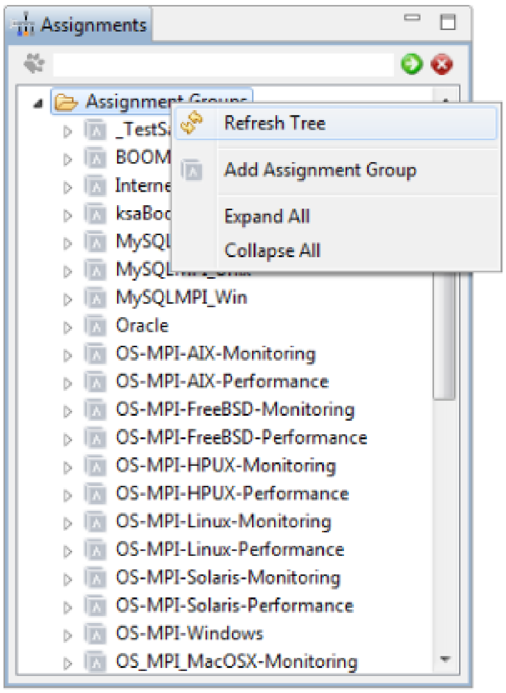
Package links can also be added to the assignments by drag & drop one or more packages from the package list to an assignment group.

Adding a policy/binary link to an assignment group will automatically enqueue the deployment on all *boom* agents where the assignment group is currently deployed. For more details about the 'Deployment Queue' see chapter Assignments Summary.

6.7.3. Assignment Operations

Right-click on an Assignment List element opens appropriate context menu:

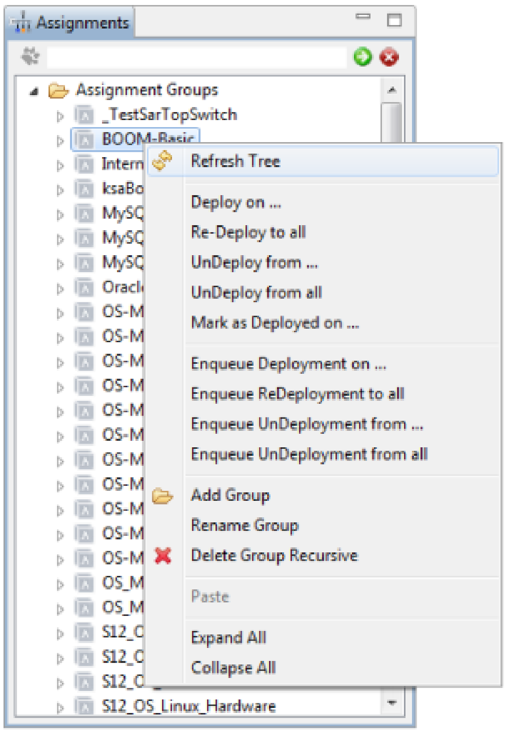
Context menu on top folder level



Add Assignment Group:

Assignment groups can only be added to the top level folder. It is not possible to create an assignment group inside existing assignment groups.

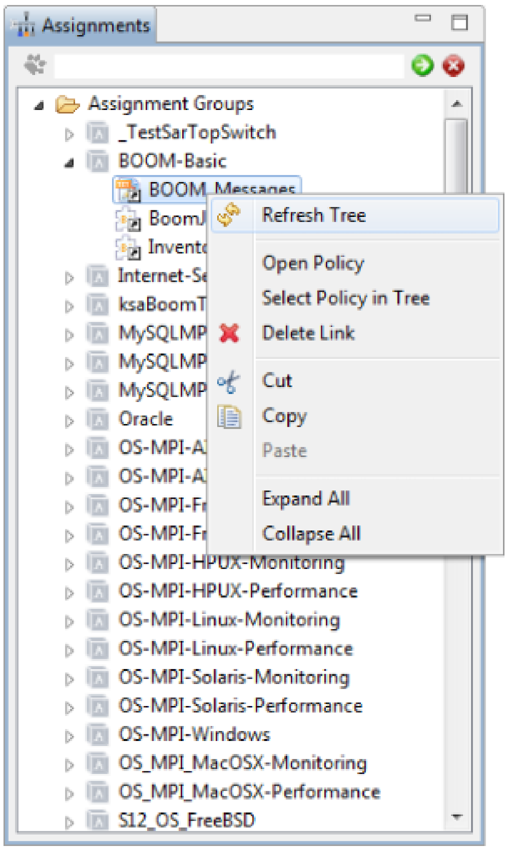
Context menu on Assignment level



Deploy on:	Deploys the selected assignment group and all child elements to one ore more *boom* agents. It also starts necessary monitor tasks. During the deployment process you have to select the *boom* agents where the selected assignments should be deployed. For more information see the paragraph Deploy on/Undeploy below.

Re-Deploy to all:	This will trigger the re-deployment of the selected assignment group and all it’s child elements. The assignment group will be re-deployed to all *boom* agents where the group was previously deployed on. Please see Assignments Summary for more information.
UnDeploy from:	This removes the assignment group and all it’s child elements from the selected *boom* agent. The monitor tasks will be stopped. During the undeployment process you have to select the *boom* agents from where the selected assignments should be undeployed. For more information see the paragraph Deploy on/Undeploy below.
UnDeploy from all:	This removes the assignment group and all it’s child elements from all boomagents where the group was previously deployed on.
Mark as Deployed on ...:	
Enqueue Deployment/UnDeployment/ReDeployment:	All "Enqueue Deployment" actions will be added to the 'Deployment Queue' and must be triggered manually! These actions will **NOT** be started automatically! For more details about the 'Deployment Queue' see chapter Assignments Summary.
Add Group:	A new assignment group can only be added to the top level folder, it is not possible to create an assignment group inside an existing assignment group.
Rename Group:	All *boom* agents were the assignment group is deployed on will automatically recognize the renaming of the assignment group.
Delete Group Recursive:	The selected assignment group and all its child elements (policy links and subfolders) will be completely removed from the assignment tree. You can select multiple assignment groups to delete them. The deletion of an assignment group is automatically put in deployment queue for all *boom* agents where the group is deployed on. Real deployment will start after 'Trigger Deployment' action in 'Assignment Summary' view.

Context menu on Policy/Package link level



Open Policy:	This will open the appropriate 'Policy Details' in the 'Policy Details View'.
Select Policy in Tree:	Points you to the specified policy in the policy view.
Delete Link:	Select one or more links to remove them from the assignment group.

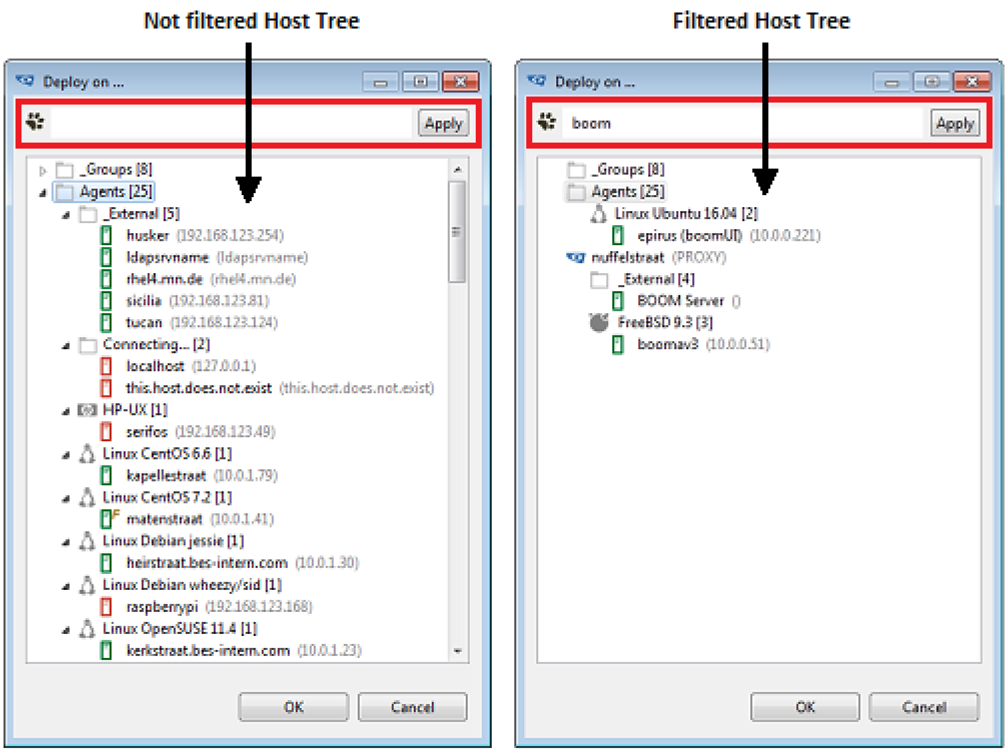
This will automatically enqueue undeployment of the selected element from all *boom* agents were the assignment group is currently deployed on. Do not forget that the enqueued actions must be triggered manually!

For more details about the 'Deployment Queue' see chapter Assignments Summary.

Cut/Copy/Paste: Copy/Paste gives you the possibility to copy a single assignment within the same assignment group or to another assignment group. Cut/Paste allows you to move a single or multiple assignments to another assignment group.

Deploy on / Undeploy from

During the deployment/undeployment process you need to select the *boom* agents where you want to deploy/undeploy the assignments. A filter function inside the 'Deployment Dialog' helps to find the appropriate *boom* agents. Just enter some letters that matches with the *boom* agents you need for the deployment process and press the 'Apply' button. To remove an existing filter from the tree you have to clean the filter input field and press the 'Apply' button.



```
_Groups [8]
Agents [25]
  _External [5]
    husker (192.168.123.254)
    ldapsrvname (ldapsrvname)
    rhel4.mn.de (rhel4.mn.de)
    sicilia (192.168.123.81)
    tucan (192.168.123.124)
  Connecting... [2]
    localhost (127.0.0.1)
    this.host.does.not.exist (this.host.does.not.exist)
  HP-UX [1]
    serifos (192.168.123.49)
  Linux CentOS 6.6 [1]
    kapellestraat (10.0.1.79)
  Linux CentOS 7.2 [1]
    matenstraat (10.0.1.41)
  Linux Debian jessie [1]
    heirstraat.bes-intern.com (10.0.1.30)
  Linux Debian wheezy/sid [1]
    raspberrypi (192.168.123.168)
  Linux OpenSUSE 11.4 [1]
    kerkstraat.bes-intern.com (10.0.1.23)
```

```
boom
_Groups [8]
Agents [25]
  Linux Ubuntu 16.04 [2]
    epirus (boomUI) (10.0.0.221)
nuffelstraat (PROXY)
  _External [4]
    BOOM Server ()
  FreeBSD 9.3 [3]
    boomav3 (10.0.0.51)
```

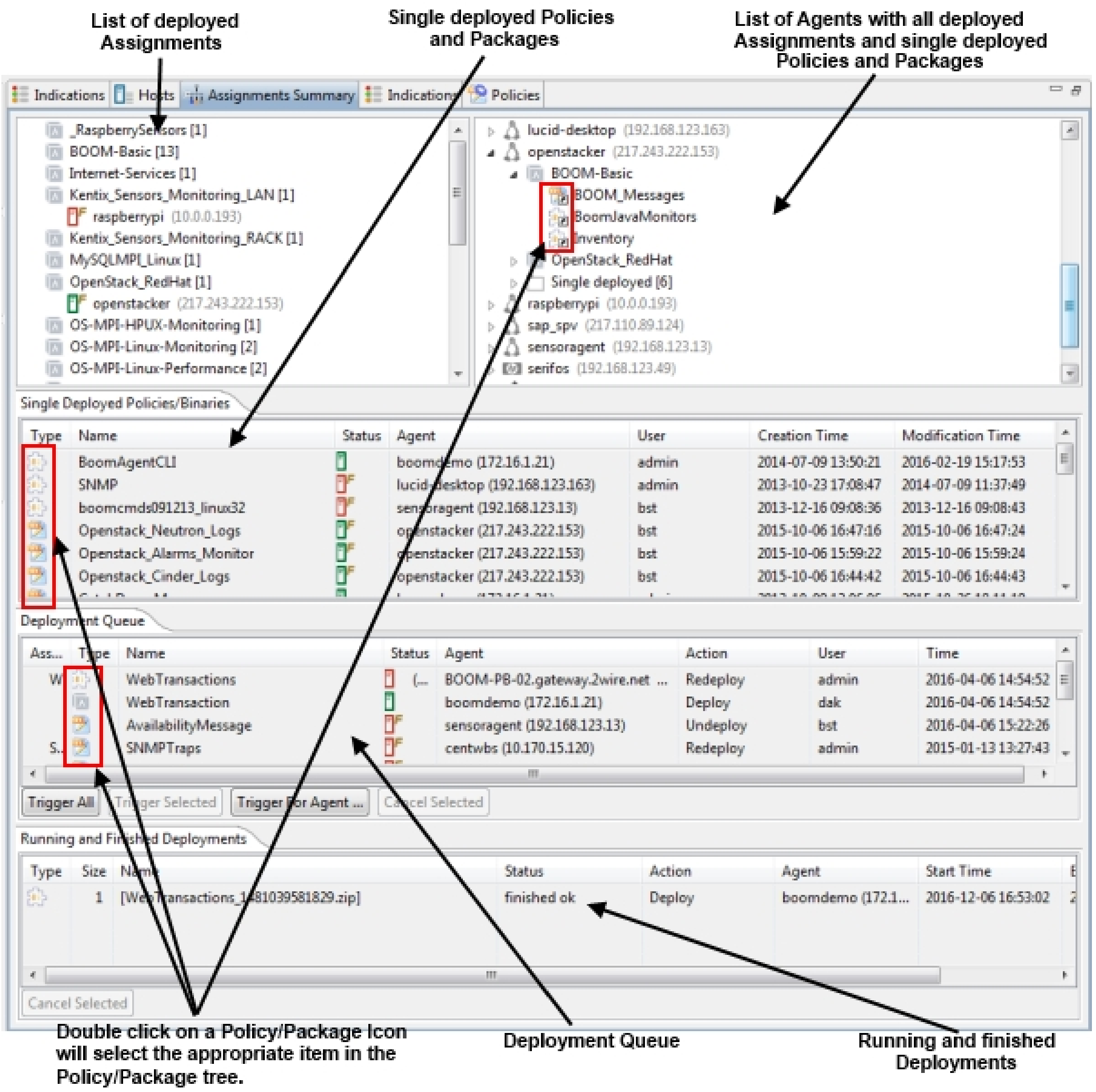
See also chapter Assignments Concept for further information about assignments.

6.8. Assignments Summary

6.8.1. General Information

The Assignments Summary View gives an overview over all deployed assignments groups and the single deployed policies and binary packages.

Please note: This view is focused on the *boom* server inventory and reflects the stored assignments. It does not trigger checking on the remote agents for existing differences. To check for any differences use the 'Hosts' view.


List of deployed Assignments
Single deployed Policies and Packages
List of Agents with all deployed Assignments and single deployed Policies and Packages
Indications
Hosts
Assignments Summary
Indications
Policies
Trigger All
Trigger Selected
Trigger For Agent ...
Cancel Selected
Cancel Selected
Double click on a Policy/Package Icon will select the appropriate item in the Policy/Package tree.
Deployment Queue
Running and finished Deployments

_RaspberrySensors [1]
BOOM-Basic [13]
Internet-Services [1]
Kentix_Sensors_Monitoring_LAN [1]
raspberrypi (10.0.0.193)
Kentix_Sensors_Monitoring_RACK [1]
MySQLMPI_Linux [1]
OpenStack_RedHat [1]
openstacker (217.243.222.153)
OS-MPI-HPUX-Monitoring [1]
OS-MPI-Linux-Monitoring [2]
OS-MPI-Linux-Performance [2]

lucid-desktop (192.168.123.163)
openstacker (217.243.222.153)
BOOM-Basic
BOOM_Messages
BoomJavaMonitors
Inventory
OpenStack_RedHat
Single deployed [6]
raspberrypi (10.0.0.193)
sap_spv (217.110.89.124)
sensoragent (192.168.123.13)
serifos (192.168.123.49)

Single Deployed Policies/Binaries

Type	Name	Status	Agent	User	Creation Time	Modification Time
	BoomAgentCLI	🟩	boomdemo (172.16.1.21)	admin	2014-07-09 13:50:21	2016-02-19 15:17:53
	SNMP	🟥	lucid-desktop (192.168.123.163)	admin	2013-10-23 17:08:47	2014-07-09 11:37:49
	boomcmds091213_linux32	🟥	sensoragent (192.168.123.13)	bst	2013-12-16 09:08:36	2013-12-16 09:08:43
	Openstack_Neutron_Logs	🟩	openstacker (217.243.222.153)	bst	2015-10-06 16:47:16	2015-10-06 16:47:24
	Openstack_Alarms_Monitor	🟩	openstacker (217.243.222.153)	bst	2015-10-06 15:59:22	2015-10-06 15:59:24
	Openstack_Cinder_Logs	🟩	openstacker (217.243.222.153)	bst	2015-10-06 16:44:42	2015-10-06 16:44:43

Deployment Queue

Ass...	Type	Name	Status	Agent	Action	User	Time
W		WebTransactions	🟥 (...	BOOM-PB-02.gateway.2wire.net ...	Redeploy	admin	2016-04-06 14:54:52
		WebTransaction	🟩	boomdemo (172.16.1.21)	Deploy	dak	2016-04-06 14:54:52
		AvailabilityMessage	🟥	sensoragent (192.168.123.13)	Undeploy	bst	2016-04-06 15:22:26
S..		SNMPTraps	🟥	centwbs (10.170.15.120)	Redeploy	admin	2015-01-13 13:27:43

Running and Finished Deployments

Type	Size	Name	Status	Action	Agent	Start Time
	1	[WebTransactions_1481039581829.zip]	finished ok	Deploy	boomdemo (172.1...	2016-12-06 16:53:02

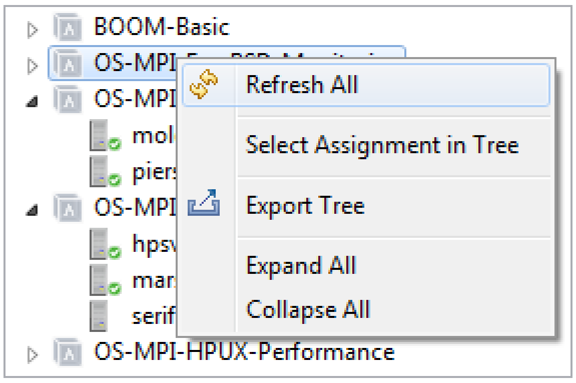
Assignment list:	The list shows all "deployed" assignment groups with all the agents they are currently deployed on.
Agent list:	It shows all known *boom* agents with their deployed assignment groups and single deployed policies.
Single Deployed Policy/Binaries:	This table contains all single deployed policies and binaries with the following information: type, name, *boom* agent were the policy is deployed on, the user who deployed the policy, creation time and modification time. The table contains multiple entries of the same policy/binary package if the policy/binary package is deployed on multiple *boom* agents.
Deployment Queue:	Shows all deployments that are enqueued but not active yet, e.g. as preparation for some version move or rollout of a new service.
Running and Finished Deployments:	Displays a list of currently running deployment jobs as well as finished deployment jobs with their result status. Please note that deployment failures will also trigger according error indications.

Double-click on a policy icon in the tree or the table will activate the policy tree view and select the appropriate policy.

6.8.2. Assignment Operations and Filtering

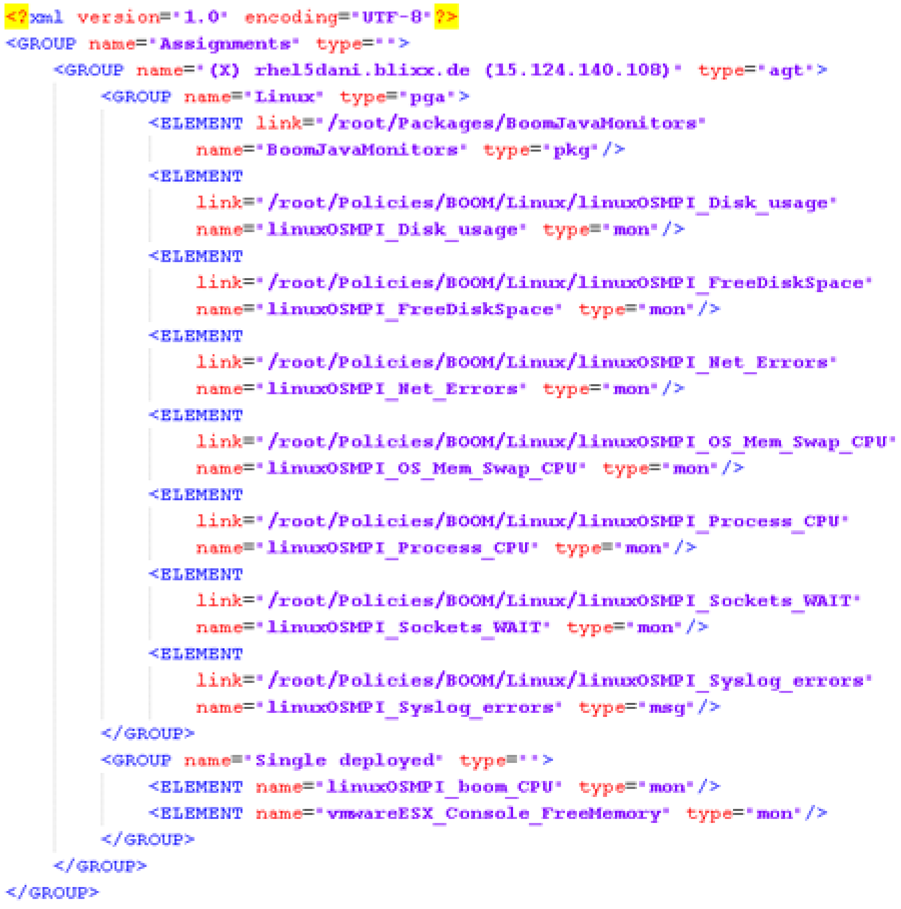
Right-click on one of the lists or tables will open the following context menus:

Assignments List



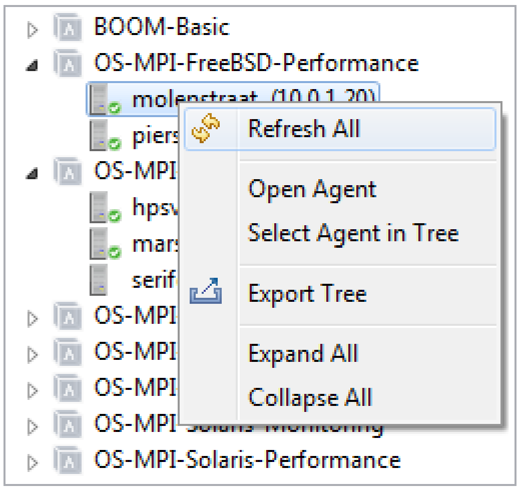
Refresh All:	The 'Refresh All' function will refresh the complete 'Assignments Summary View'.
Select Assignment in Tree:	Points you to the specified assignment in the assignment view.
Expand All:	This will expand the complete tree.
Collapse All:	This will collapse the complete tree.
Export Tree:	The assignments summary tree can be exported to an XML formatted file.

Example:



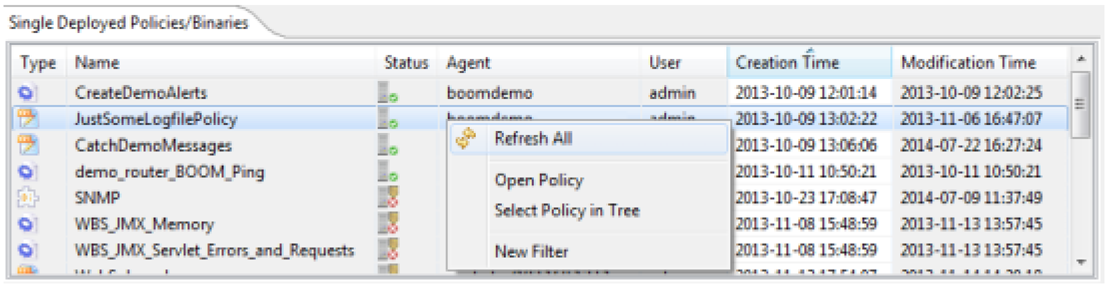
```
<?xml version="1.0" encoding="UTF-8"?>
<GROUP name="Assignments" type="">
    <GROUP name="(X) rhel5dani.blixx.de (15.124.140.108)" type="agt">
        <GROUP name="Linux" type="pga">
            <ELEMENT link="/root/Packages/BoomJavaMonitors"
                name="BoomJavaMonitors" type="pkg"/>
            <ELEMENT
                link="/root/Policies/BOOM/Linux/linuxOSMPI_Disk_usage"
                name="linuxOSMPI_Disk_usage" type="mon"/>
            <ELEMENT
                link="/root/Policies/BOOM/Linux/linuxOSMPI_FreeDiskSpace"
                name="linuxOSMPI_FreeDiskSpace" type="mon"/>
            <ELEMENT
                link="/root/Policies/BOOM/Linux/linuxOSMPI_Net_Errors"
                name="linuxOSMPI_Net_Errors" type="mon"/>
            <ELEMENT
                link="/root/Policies/BOOM/Linux/linuxOSMPI_OS_Mem_Swap_CPU"
                name="linuxOSMPI_OS_Mem_Swap_CPU" type="mon"/>
            <ELEMENT
                link="/root/Policies/BOOM/Linux/linuxOSMPI_Process_CPU"
                name="linuxOSMPI_Process_CPU" type="mon"/>
            <ELEMENT
                link="/root/Policies/BOOM/Linux/linuxOSMPI_Sockets_WAIT"
                name="linuxOSMPI_Sockets_WAIT" type="mon"/>
            <ELEMENT
                link="/root/Policies/BOOM/Linux/linuxOSMPI_Syslog_errors"
                name="linuxOSMPI_Syslog_errors" type="msg"/>
        </GROUP>
        <GROUP name="Single deployed" type="">
            <ELEMENT name="linuxOSMPI_boom_CPU" type="mon"/>
            <ELEMENT name="vmwareESX_Console_FreeMemory" type="mon"/>
        </GROUP>
    </GROUP>
</GROUP>
```

Agents List



Refresh All:	The 'Refresh All' function will refresh the complete 'Agents Summary View'.
Open Agent:	Opens the 'Agent Details View'.
Select Agent in Tree:	Points you to the specified agent in the agent view.
Expand All:	This will expand the complete tree.
Collapse All:	This will collapse the complete tree.
Export Tree:	The assignments summary tree can be exported to an XML formatted file.

Single Deployed Policies/Binaries



Type	Name	Status	Agent	User	Creation Time	Modification Time
	CreateDemoAlerts		boomdemo	admin	2013-10-09 12:01:14	2013-10-09 12:02:25
	JustSomeLogfilePolicy		boomdemo	admin	2013-10-09 13:02:22	2013-11-06 16:47:07
	CatchDemoMessages				2013-10-09 13:06:06	2014-07-22 16:27:24
	demo_router_BOOM_Ping				2013-10-11 10:50:21	2013-10-11 10:50:21
	SNMP				2013-10-23 17:08:47	2014-07-09 11:37:49
	WBS_JMX_Memory				2013-11-08 15:48:59	2013-11-13 13:57:45
	WBS_JMX_Servlet_Errors_and_Requests				2013-11-08 15:48:59	2013-11-13 13:57:45

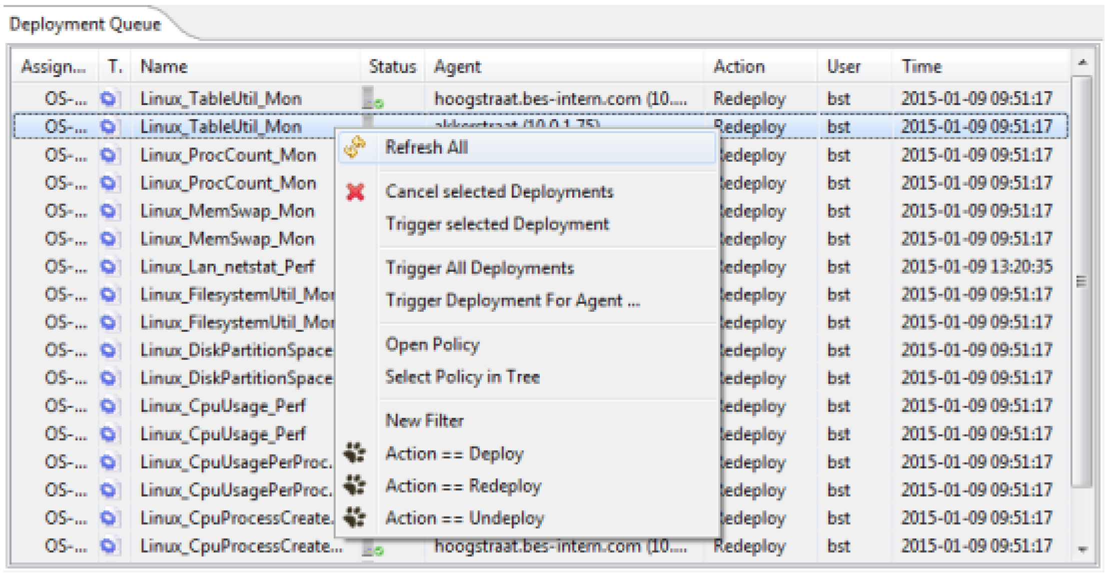
Refresh All: This will refresh the complete 'Assignments Summary View'.

Open Policy: Open the policy details view.

Select Policy in Tree: Points you to the specified policy in the policies view.

New Filter: For more information about filtering see below.

Enqueue Deployment



Assign...	T.	Name	Status	Agent	Action	User	Time
OS-...		Linux_TableUtil_Mon		hoogstraat.bes-intern.com (10....	Redeploy	bst	2015-01-09 09:51:17
OS-...		Linux_TableUtil_Mon		akkerstraat (10.0.1.75)	Redeploy	bst	2015-01-09 09:51:17
OS-...		Linux_ProcCount_Mon			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_ProcCount_Mon			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_MemSwap_Mon			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_MemSwap_Mon			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_Lan_netstat_Perf			edeploy	bst	2015-01-09 13:20:35
OS-...		Linux_FilesystemUtil_Mo			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_FilesystemUtil_Mo			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_DiskPartitionSpace			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_DiskPartitionSpace			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_CpuUsage_Perf			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_CpuUsage_Perf			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_CpuUsagePerProc.			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_CpuUsagePerProc.			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_CpuProcessCreate.			edeploy	bst	2015-01-09 09:51:17
OS-...		Linux_CpuProcessCreate...		hoogstraat.bes-intern.com (10....	Redeploy	bst	2015-01-09 09:51:17

Refresh All: Refreshes the complete assignments summary view.

Cancel selected Deployment: Removes the selected enqueued deployment actions from the 'Deployment Queue'.

Trigger selected Deployment: Triggers the un-/deployment of the selected queue element.

Trigger All Deployments: Triggers the un-/deployment of all queue elements.

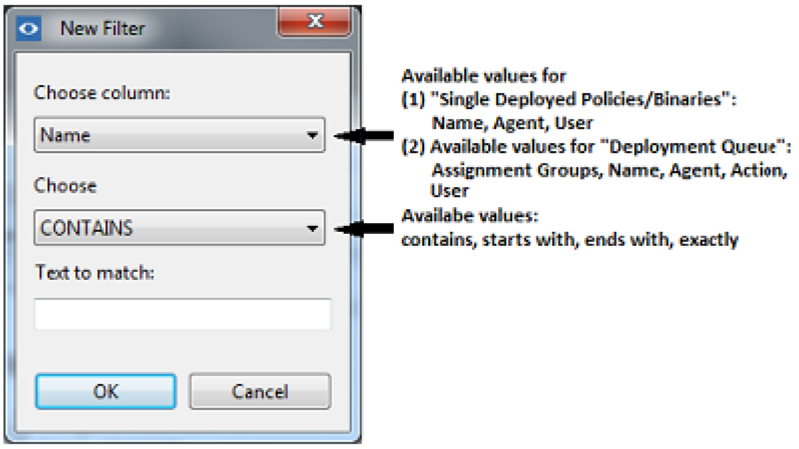
Trigger Deployment For Agents: Triggers the un-/deployment of the selected queue element for the specified agent.

Open Policy: Opens the policy details view.

Select Policy in Tree: Points you to the specified policy in the policies view.

New Filter:

Both tables are displaying by default all single deployed policies/binaries and enqueued deployment actions. The filtering function gives you now the possibility to filter the table content and display only the relevant data by adding one or more filters to the table.



Available values for
(1) "Single Deployed Policies/Binaries":
Name, Agent, User
(2) Available values for "Deployment Queue":
Assignment Groups, Name, Agent, Action, User
Availabe values:
contains, starts with, ends with, exactly

Predefined Filters:
The following predefined filters available for the deployment queue table:
Action==Deploy
Action==ReDeploy
Action==UnDeploy

A check mark in front of a filter indicates if that the filter is active. To activate/deactivate a predefined filter simply click on that filter in the context menu.

Please note: Custom filters will be also added the context menu but they cannot be deactivated! If you click on a custom filter in the context menu it will be deleted!

6.8.3. Deployment Queue

Successful Deploying of Policies and Packages

Every un-deploy/deploy operations is scheduled to the **Deployment Queue** and processed.

- For every Un-deploy/deploy operation a normal indication is created:
  `"UnDeploy task is scheduled for Agent" or "Deployment is scheduled for Agent .."`
- During the operation the deployment task is visible in the Deployment Queue of the "Assignment Summary" view.
- The deployment queue entry is removed. A normal indication is created:
  `"Policy <pol> was deployed" or "Removed policy …"`

Unsuccessful Deploying of Policies and Packages

Every un-deploy/deploy operations is scheduled to the **Deployment Queue** and processed.

- For every Un-deploy/deploy operation a normal indication is created:
  `"UnDeploy task is scheduled for Agent" or "Deployment is scheduled for Agent .."`
- During the operation the deployment task is visible in the Deployment Queue of the "Assignment Summary" view.

- These operations are retried for 10 minutes.
- If the deploy/un-deploy is not successful for 10 minutes, the operation times out. The Deployment Queue entry is **not** removed.
  - In case the agent is **offline** when the deployment starts an error indication is created:

    `"Deployment failed on Agent: <agent>"`

  - In case the agent is **online (not firewalled)** when the deployment starts but the deployment task could not be finished (timed out after 10 minutes) an error indication is created:

    `"Deployment failed on Agent: <agent>"`

  - In case the agent was ` when the deployment starts but the deployment task could not be finished (timed out after 10 minutes) an Error Indication is created:

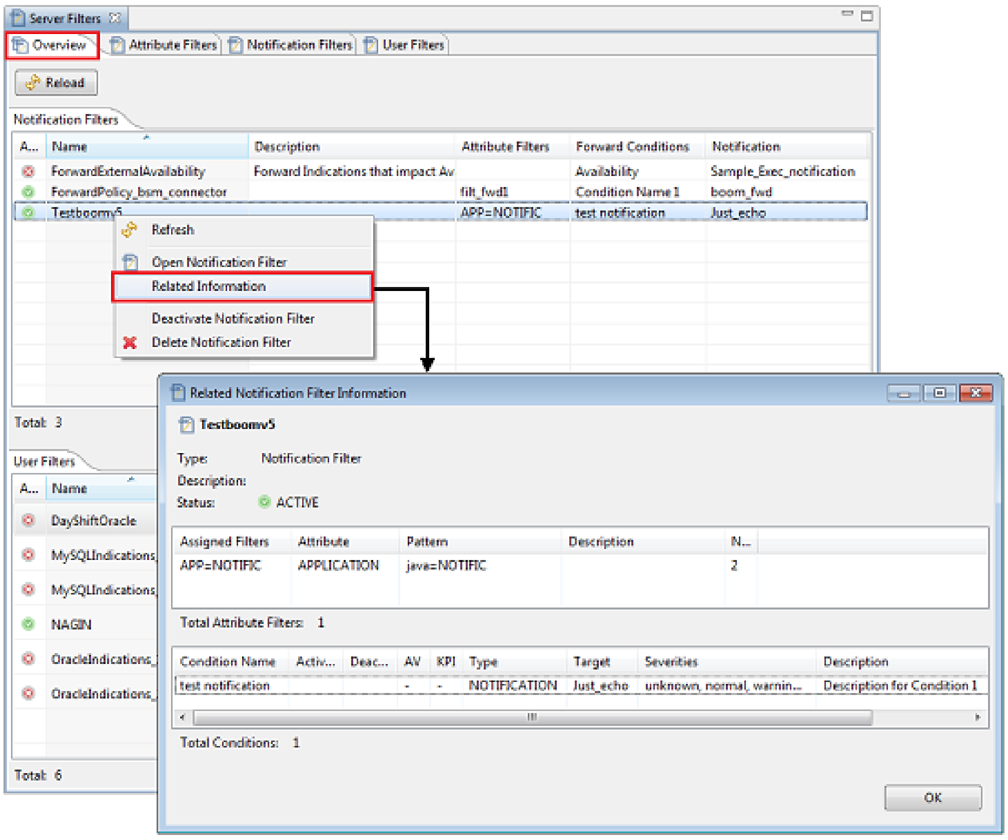
    `"Deployed Task has expired and was stopped for Agent"`

- **Note:** When the deployment operations timed out it will neither be restarted nor the queue will be emptied automatically
- The Deployment Queue has to be checked and the corresponding deployment task has to be **triggered** or **cancelled** manually
- The queue entry will be removed only after a new deploy/un-deploy operation.

6.9. Server Filters

6.9.1. General Information

The Server filters are used to create user roles, responsibility areas, expert centers, manager views, dedicated external integrations and more by defining restrictions for the forwarding of indications to multiple target *boom* servers and/or local user roles or external programs like notification and trouble ticket systems.



Notification Filters

A...	Name	Description	Attribute Filters	Forward Conditions	Notification
🟥	ForwardExternalAvailability	Forward Indications that impact Av		Availability	Sample_Exec_notification
🟩	ForwardPolicy_bsm_connector		filt_fwd1	Condition Name 1	boom_fwd
🟩	Testboomv5		APP=NOTIFIC	test notification	Just_echo

Total: 3

User Filters

A...	Name
🟥	DayShiftOracle
🟥	MySQLIndications_
🟥	MySQLIndications_
🟩	NAGIN
🟥	OracleIndications_
🟥	OracleIndications_

Total: 6

Related Notification Filter Information

Testboomv5

Type: Notification Filter
Description:
Status: ACTIVE

Assigned Filters	Attribute	Pattern	Description	N...
APP=NOTIFIC	APPLICATION	java=NOTIFIC		2

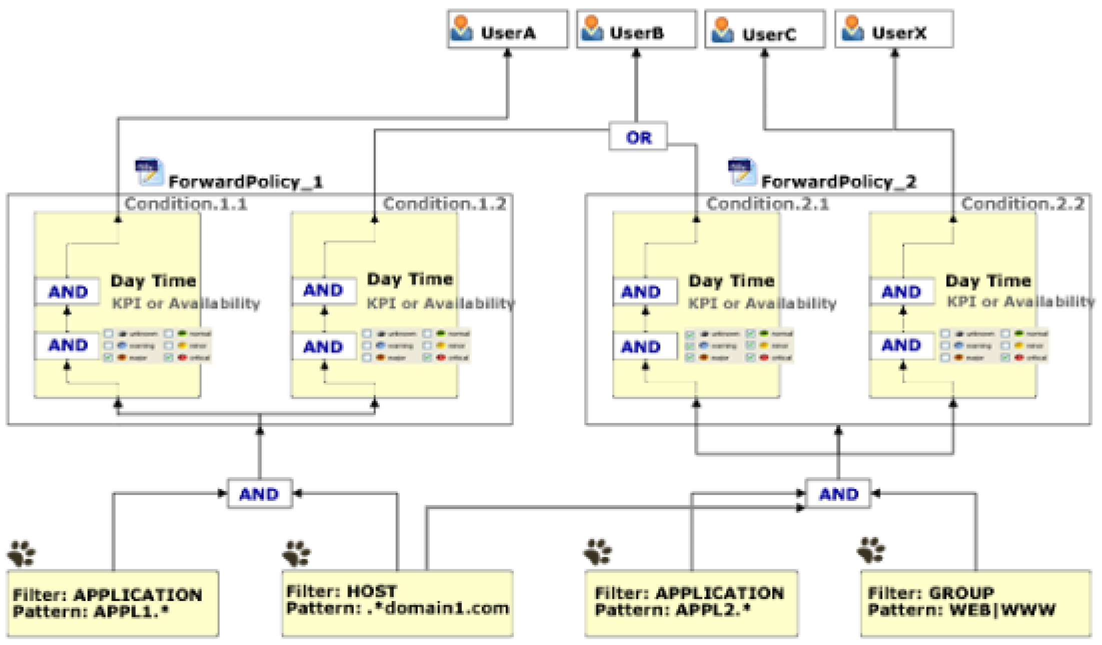
Total Attribute Filters: 1

Condition Name	Activ...	Deac...	AV	KPI	Type	Target	Severities	Description
test notification			-	-	NOTIFICATION	Just_echo	unknown, normal, warnin...	Description for Condition 1

Total Conditions: 1

The architecture of the *boom* filters is based on the following rules:

- A 'Attribute Filter' defines a low level pattern for an attribute of the indication
- The 'Attribute Filters' are used in the 'ForwardPolicy' of the Notification Filters and/or in the 'User Filters'.
- 'Attribute Filters' can be used in multiple 'Forward Policies' and 'User Filters'.
- A Notification 'Forward Policy' contains one or more 'Forward Conditions' that specify the targets (notification service which can be external programs or an email) and additional filters on severity, daytime, availability and KPI attributes
- The 'Forward Policy' and 'User Filters' use boolean 'AND' between the 'Attribute filters' and additional attributes except availability and KPI flags
- The 'Forward Policy' based on the assigned 'Forward Filters' and the 'User Filter' filter each incoming indication and forward matches to the specified targets (notification services or user roles).
- A logical 'OR' is used if multiple 'Forward Policies' or 'User Filters' are related to the same targets



Indication Attributes used in Forward Filters

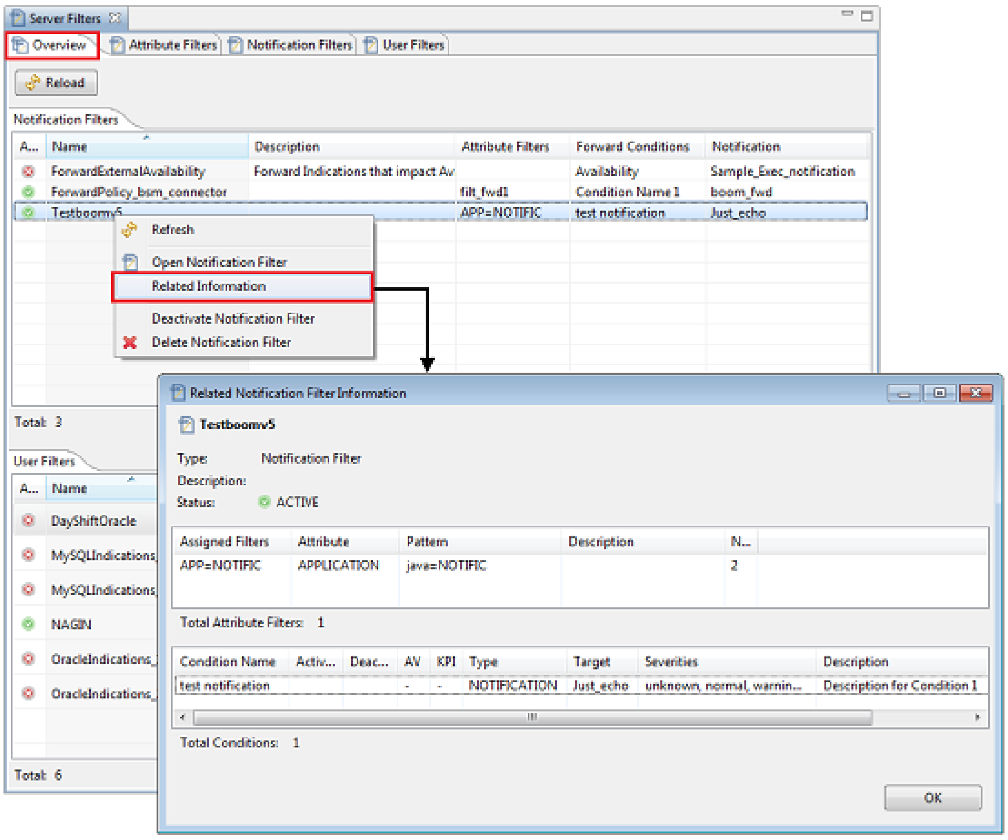
- Application
- Group
- Object
- Host
- Host Group
- Severity
- Source: this attribute is constructed as <Monitor|Message>:<Policyname>:<ConditionID> (check e.g. the Message Details to see the Source attribute), e.g. a Filter ".:**Oracle.**:.*" would match all indications coming from Policies which names start with Oracle
- Text
- Key
- AgentID
- Agent Host
- Slave Server

- Custom Attributes [1...15]
- Service (reserved)

> Starting with *boom* v4 the Target 'EXEC' and specifying rules for single Users was deprecated. The configuration for Target EXEC is replaced by the Notification Configuration of type EXEC, so that there is a consistent configuration for all notification services.

6.9.2. Server Filter Overview

The 'Server Filter Overview' table gives a summary of all available server filters.



Notification Filters

A...	Name	Description	Attribute Filters	Forward Conditions	Notification
🔴	ForwardExternalAvailability	Forward Indications that impact Av		Availability	Sample_Exec_notification
🟢	ForwardPolicy_bsm_connector		filt_fwd1	Condition Name 1	boom_fwd
🟢	Testboomv5		APP=NOTIFIC	test notification	Just_echo

Total: 3

User Filters

A...	Name
🔴	DayShiftOracle
🔴	MySQLIndications
🔴	MySQLIndications
🟢	NAGIN
🔴	OracleIndications_
🔴	OracleIndications_

Total: 6

Related Notification Filter Information

Testboomv5

Type: Notification Filter
Description:
Status: 🟢 ACTIVE

Assigned Filters	Attribute	Pattern	Description	N...
APP=NOTIFIC	APPLICATION	java=NOTIFIC		2

Total Attribute Filters: 1

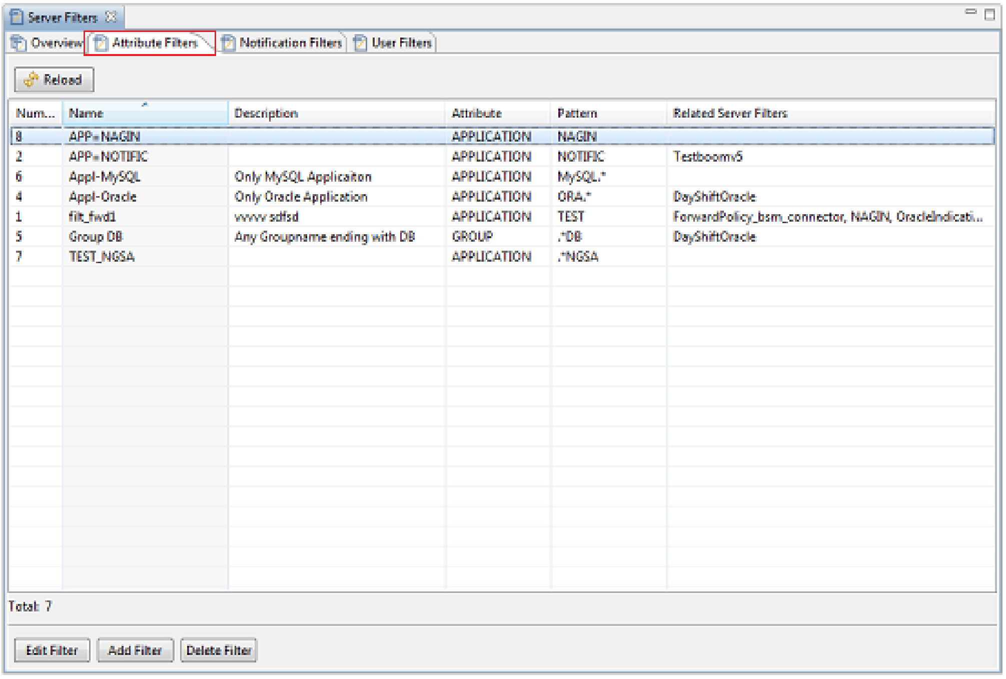
Condition Name	Activ...	Deac...	AV	KPI	Type	Target	Severities	Description
test notification			-	-	NOTIFICATION	Just_echo	unknown, normal, warnin...	Description for Condition 1

Total Conditions: 1

Show Related Information gives an overview of the assigned filters and the defined conditions.

6.9.3. Attribute Filters

The 'Attribute Filters' are the building blocks for the 'Notification Filters' and 'User Filters'. Each filter defines a low level pattern for a single attribute of the indication.



Server Filters

Overview | Attribute Filters | Notification Filters | User Filters

Reload

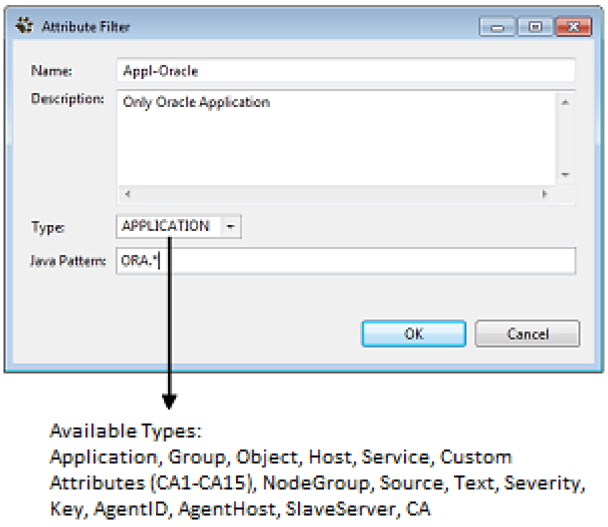
Num...	Name	Description	Attribute	Pattern	Related Server Filters
8	APP=NAGIN		APPLICATION	NAGIN	
2	APP=NOTIFIC		APPLICATION	NOTIFIC	Testboomv5
6	Appl-MySQL	Only MySQL Applicaiton	APPLICATION	MySQL.*	
4	Appl-Oracle	Only Oracle Application	APPLICATION	ORA.*	DayShiftOracle
1	filt_fwd1	vvvvv sdfsd	APPLICATION	TEST	ForwardPolicy_bsm_connector, NAGIN, OracleIndicati...
5	Group DB	Any Groupname ending with DB	GROUP	.*DB	DayShiftOracle
7	TEST_NGSA		APPLICATION	.*NGSA	

Total: 7

Edit Filter | Add Filter | Delete Filter

Edit Filter

The Forward Filter name can be any name that clarifies its purpose. It does not need to be unique because the Filter will be identified by its 'Number'. If the Forward Filter name will be changed, all Forward Policies that are linked to this filter will be automatically updated.



Attribute Filter

Name: Appl-Oracle

Description: Only Oracle Application

Type: APPLICATION

Java Pattern: ORA.*

OK | Cancel

Available Types:
Application, Group, Object, Host, Service, Custom Attributes (CA1-CA15), NodeGroup, Source, Text, Severity, Key, AgentID, AgentHost, SlaveServer, CA

The following filter types are available. The Java Pattern defines the filter criteria itself.

Type field	Description
APPLICATION	Application name
GROUP	Indication Group name
OBJECT	Object name
HOST	Node from where the Indication has been issued
SERVICE	reserved
CA1 - CA15	match the value of one specific CA from 1 to 15
NODEGROUP	NodeGroup name - use "Select" to filter one or more Node Groups

Type field	Description
SOURCE	source policy
TEXT	Indication text
SEVERITY	severity of the Indication
KEY	Indication Key
AGENTID	Unique number to identify the *boom* agent - use "Select" to filter one or more Agents and/or Node Groups
AGENT_HOST	Host Name of the *boom* agent
SLAVE_SERVER	Name of the *boom* Slave Server
CA	match the value of any CA from CA1 to CA15

Delete Filter

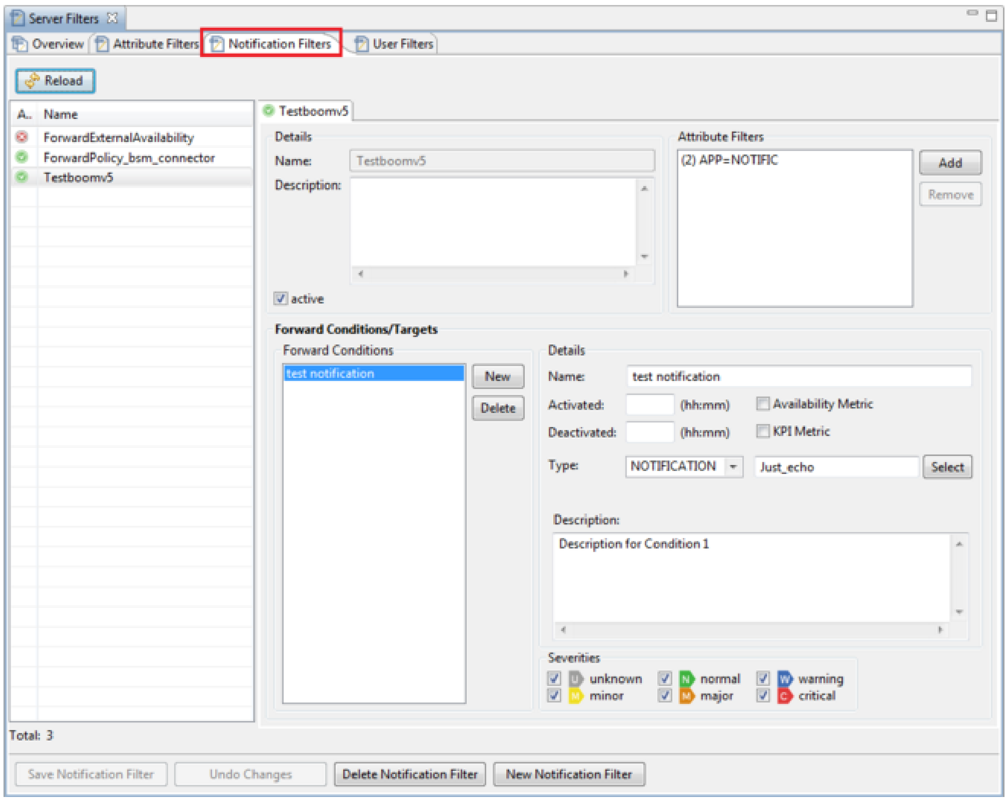
All selected Attribute Filters will be deleted from the Filters table. Please note that an Attribute Filter can not be deleted if it is still used by an Notification Filter or User Filter.

Add Filter

Create a new Attribute Filter.

6.9.4. Notification Filters

The tab **"Notification Filters"** tab shows the following information



ForwardExternalAvailability
ForwardPolicy_bsm_connector
Testboomv5

Name: Testboomv5
Description:
☑ active

(2) APP=NOTIFIC

test notification

Name: test notification
Activated: (hh:mm) ☐ Availability Metric
Deactivated: (hh:mm) ☐ KPI Metric
Type: NOTIFICATION Just_echo

Description:
Description for Condition 1

☑ unknown ☑ normal ☑ warning
☑ minor ☑ major ☑ critical

Total: 3

The tab lists all available **Notification Filters**.
Select **New Notification Filter** or **Delete Notification Filter** buttons to add a new filter or delete the selected filter.
Select an existing **Notification Filter** from the list to modify the filter.

Notification Filter Details:

Name

This is the **unique name** of the Notification Filter.

Description

A description of the Notification Filter

active

This will set the Notification Filter status to active. If the filter is not active, no notification will be triggered.

Attribute Filters:

Add

Opens a pop-up dialog that allows to select from the defined Attribute Filters which ones should be used by the Notification Filter. The Attribute Filters are connected with a logical "And" when the indications get filtered, meaning all Attribute Filters must be matched by an indication to trigger the Notification Service.
Please note that the filter has to be saved to finally link the Attribute Filters to the Notification Filter.

Forward Conditions/Targets (each Notification Filter might have multiple Targets)

Name

The Forward Condition name is a **unique name**.

Activated

This is the activation time.

Deactivated

This is the deactivation time.

Availability Metric

Only indications having the Availability Flag set will match this condition.

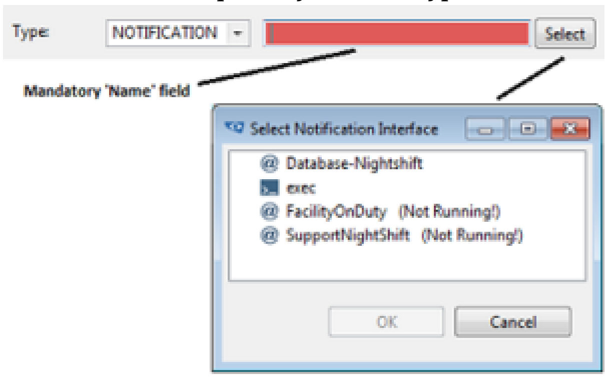
KPI Metric

Only indications having the KPI Metric Flag set will match this condition.

Type

Possible types are: NOTIFICATION

'NOTIFICATION'
The selection of an existing notification service is mandatory. Make sure that this notification service is created first or choose temporarily another Type to be able to save the Forward Policy.


Type:
NOTIFICATION
Select
Mandatory 'Name' field
Select Notification Interface
Database-Nightshift
exec
FacilityOnDuty (Not Running!)
SupportNightShift (Not Running!)
OK
Cancel

Severities

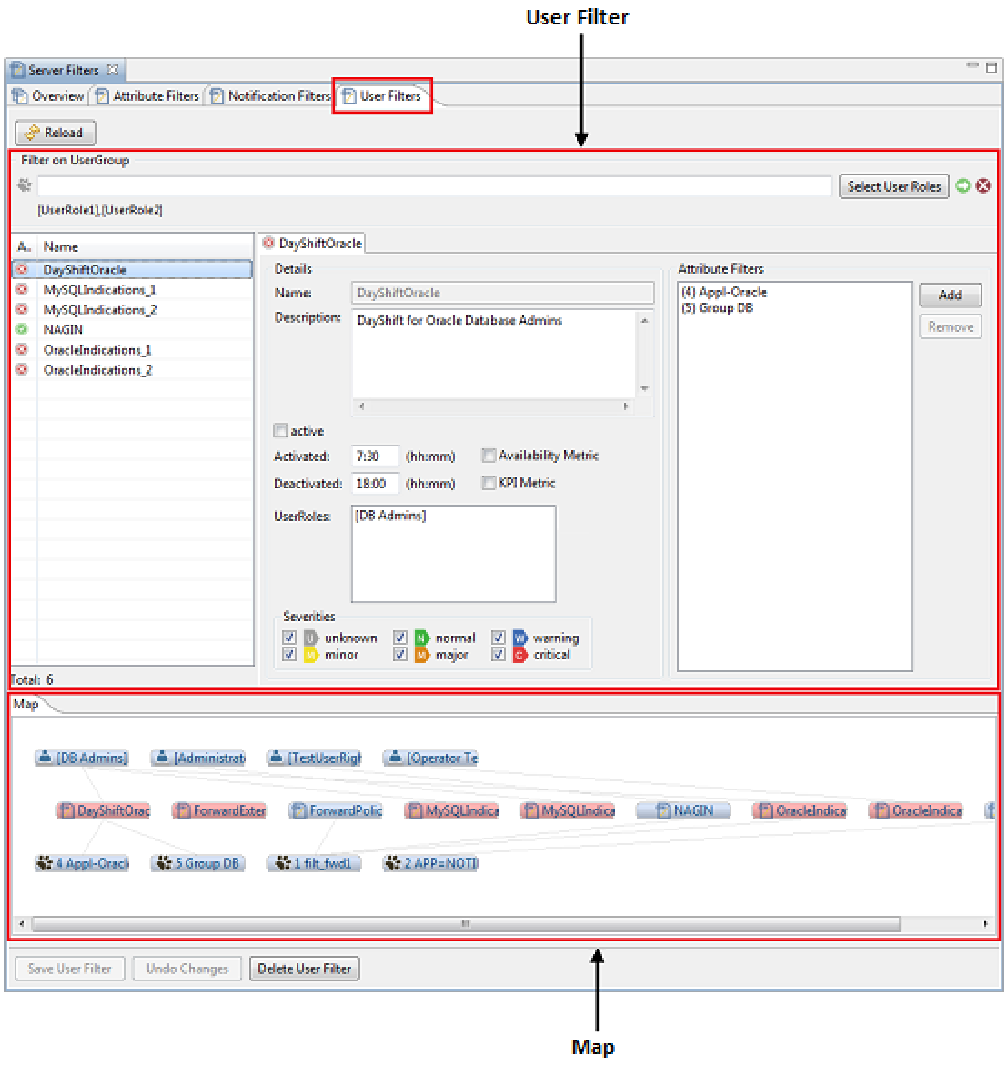
Only indications with selected severities match with the condition.

Description

A description of the Forward Condition.

6.9.5. User Filters

The **"User Filters"** tab shows the following information:



Filter on UserGroup: [UserRole1],[UserRole2]

A..	Name
	DayShiftOracle
	MySQLIndications_1
	MySQLIndications_2
	NAGIN
	OracleIndications_1
	OracleIndications_2

Total: 6

DayShiftOracle

Details
Name: DayShiftOracle
Description: DayShift for Oracle Database Admins
☐ active
Activated: 7:30 (hh:mm) ☐ Availability Metric
Deactivated: 18:00 (hh:mm) ☐ KPI Metric
UserRoles: [DB Admins]

Severities
☑ unknown ☑ normal ☑ warning
☑ minor ☑ major ☑ critical

Attribute Filters
(4) Appl-Oracle
(5) Group DB

Map: [DB Admins], [Administrat, [TestUserRigh, [Operator Te, DayShiftOrac, ForwardExter, ForwardPolic, MySQLIndica, MySQLIndica, NAGIN, OracleIndica, OracleIndica, 4 Appl-Oracl, 5 Group DB, 1 filt_fwd1, 2 APP=NOTI

The tab lists all available **User Filters**.
Select **Delete User Filter** buttons to delete the selected filter.
Select an existing User **Filter** from the list to modify the filter.

> Adding of a **new** filter can only be done via User Management’s User Filters window.
> For detailed information see chapter User Filters.

User Filter Details:

Name

This is the unique name of the User Filter.

Description

A description of the User Filter.

active

This will set the User Filter status to active. A disabled User Filter will not change the visibility of indications.

Activated

This is the activation time.

Deactivated

This is the deactivation time.

Availability Metric

Only indications having the Availability Flag set will match this condition.

KPI Metric

Only indications having the KPI Metric Flag set will match this condition.

User Roles

That user role(s) will be found here assigned by the User Management.

Severities

Only indications with selected severities match with the condition.

Attribute Filters:

Add

Opens a pop-up dialog that allows to select from the defined Attribute Filters which ones should be used by the User Filter. The Attribute Filters are connected with a logical "And", meaning all Attribute Filters must be matching.

Remove

Removes selected Attribute Filter.

Map:

Tab

The Map view gives an overview about the relation between User Filters, Attribute Filters and affected user roles.

6.10. Notifications

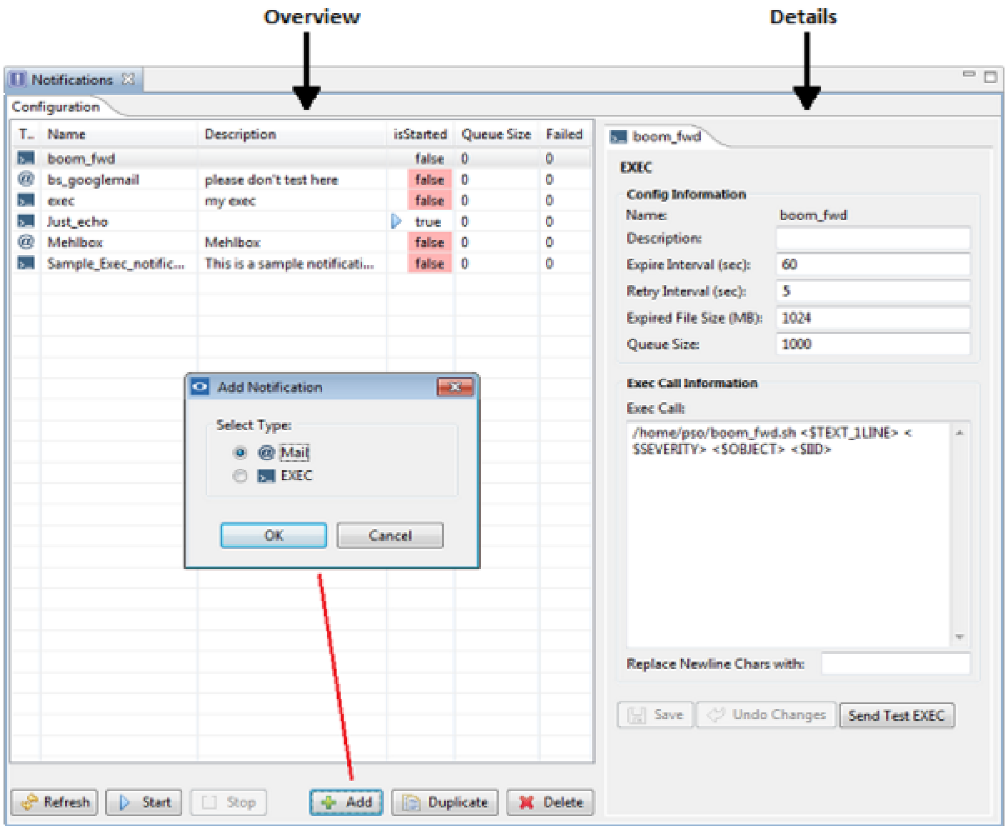
6.10.1. General Information

The Notifications View is used to configure notification services. These notification services can be used as targets for Server Filters to send indication information automatically via Email or by a program call (exec) for indications that matched the according filter rules. It is also possible to manually trigger a notification service for an selected indication directly from the Indication browser. The Notification functionality can be used to implement beside many others use cases like

- automatic Email alerting on critical problems to on call duty operators during the off-shift hours
- automatic Email alerting of critical service availability problems to specialists
- automatic Email alerting of end customer specific problems to on-site technician
- manual escalation of problems to other teams or specialists
- manual escalation of problems to help desk or ticketing systems
- integration with an umbrella system, help desk or ticketing system via script or program
- automatic initiation of an work flow in an external system via script or program

6.10.2. Notification Overview

The Notifications View shows a summary of all configured notification services and the configuration details for these services.



T..	Name	Description	isStarted	Queue Size	Failed
	boom_fwd		false	0	0
@	bs_googlemail	please don't test here	false	0	0
	exec	my exec	false	0	0
	Just_echo		true	0	0
@	Mehlbox	Mehlbox	false	0	0
	Sample_Exec_notific...	This is a sample notificati...	false	0	0

Select Type: [✓ Mail] EXEC

boom_fwd

EXEC

Config Information

Name:	boom_fwd
Description:	
Expire Interval (sec):	60
Retry Interval (sec):	5
Expired File Size (MB):	1024
Queue Size:	1000

Exec Call Information

Exec Call:

```
/home/pso/boom_fwd.sh <$TEXT_1LINE> <
$SEVERITY> <$OBJECT> <$SID>
```

Replace Newline Chars with:

The Overview section shows all configured services and their current state.

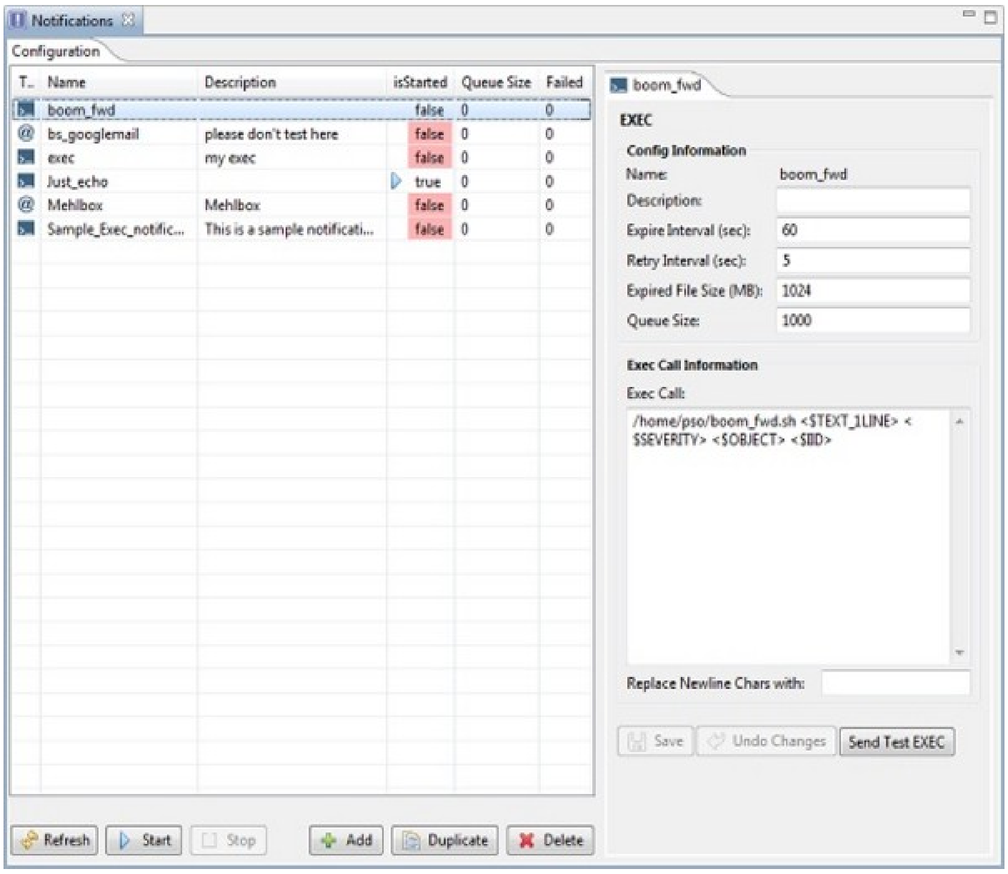
Type	Mail or Exec
Name	Choose a descriptive name that allows to identify the intended usage as this name will be used to select the target for manually as well Server Filter based notifications. No special characters are allowed in this field.
Description	Use this field to provide a more detailed description for this notification service.
isStarted	This flag indicates if this notification service is running (true) or stopped (false). A stopped notification service will not accept any requests. Trying to manually trigger a notification will return an error message telling that the service is stopped, automatic notifications that should be triggered by Server Filters are ignored.
queueSize	Shows how many requests are in the queue and will be send in the next interval.
failedQueueSize	Shows how many requests could not be send and will be retried with the next interval. Retries will be performed until the Expire Interval is reached.
Refresh	Reload the Notification configuration.
Start	Starts the selected email service.
Stop	Stops the selected email service.
Duplicate	Creates a new email service with the parameters of the selected service.

Delete	Removes the selected email service. Before removing the email service itself first remove any target conditions in the Server Filters that use this email service as target.

6.10.3. Notification Configuration

The Details sections allows to configure a new notification service or modify the service that is selected in the in the overview.

Email Notifications



T..	Name	Description	isStarted	Queue Size	Failed
	boom_fwd		false	0	0
@	bs_googlemail	please don't test here	false	0	0
	exec	my exec	false	0	0
	Just_echo		true	0	0
@	Mehlbox	Mehlbox	false	0	0
	Sample_Exec_notific...	This is a sample notificati...	false	0	0

EXEC

Config Information

Name:	boom_fwd
Description:	
Expire Interval (sec):	60
Retry Interval (sec):	5
Expired File Size (MB):	1024
Queue Size:	1000

Exec Call Information

Exec Call:

```
/home/pso/boom_fwd.sh <$TEXT_1LINE> <
$SEVERITY> <$OBJECT> <$IID>
```

Replace Newline Chars with:

<u>**Config Information**</u>

Name	Choose a descriptive name that allows to identify the email receiver or intended usage as this name will be used to select the target for manually as well Server Filter based email notifications. No special characters are allowed in this field.
Description	Use this field to provide a more detailed description for this email service.
Expire Interval (sec)	If an email can not be send the *boom* server will retry to send this message until it gets older than this limit. Emails than cannot be send in time (expired emails) will be written to the log file: `<boom_server>/srv/etc/ext/notifications/<email service name>/mail.failed.expired<timestamp>.log`
Retry Interval (sec)	This interval specifies how long the server will wait between its attempts to send the emails in the new email and failed email queues.
Expired File Size (MB)	This parameter allows to limit the size of the log file for expired Emails.

Queue Size	Shows how many emails the server will buffer for this email server.

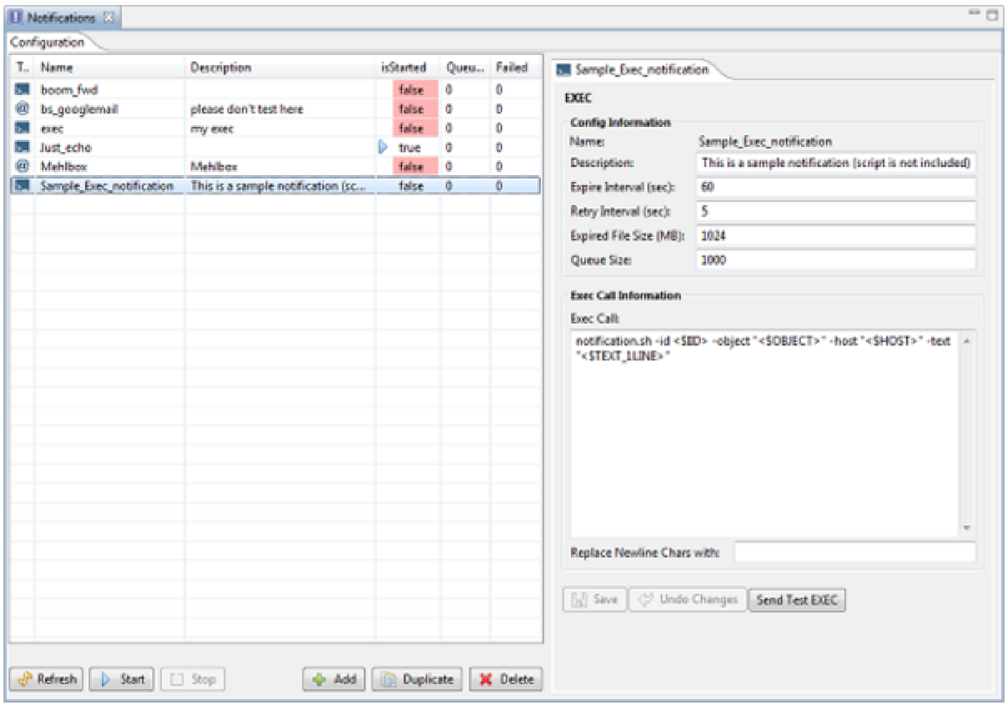
Message Information

From	Specify a valid "From" email address that will show up as sender address of the emails.
To	Sepcify the email recipients.
Subject	Specify the Subject Line of the email. Indication Variables can be used to construct the email subject. For the list of available variables please refer to the table below.
Text	Specify the email body text. Indication Variables can be used to construct the text. For the list of available variables please refer to the table below. The email can be sent in HTML format. To enable the HTML handling put the tag <html> on the first line and </html> as last line of the text.

Server Information

Outgoing Host (SMTP)	The *boom* server acts as SMTP client to send emails using an existing SMTP server (which can be an company internal server like Microsoft Exchange, Postfix or a cloud server like googlemail, hotmail). Specify the hostname of the SMTP Server in this field.
Port	Specify the port on which the SMTP Server is listening for incoming SMTP requests. The standard port for SMTP is 25, however servers are often configured to use alternate ports for security reasons.
STARTTLS	The checkmark indicates that an encrypted secured connection (TLS = Transport Layer Security) has to be used for communication with the SMTP server. This flag should only be unchecked for very good reason and in secured environments as it means that the logon credentials (and the email contents) are transferred clear text over the network.
SMTP Authentication	In almost every environment the SMTP server will require to authenticate before it accepts an email for delivery. This check mark allows to switch on/off the authentication.
User	Specify the username for the SMTP account.
Password	Specify the password for the SMTP account.
Save	Save the changes for the email service.
Undo	Undo the changes for the email service.
Send Test Mail	Send a test email to the SMTP Host to verify the configuration settings.

Exec Notifications



T..	Name	Description	isStarted	Queu...	Failed
	boom_fwd		false	0	0
	bs_googlemail	please don't test here	false	0	0
	exec	my exec	false	0	0
	Just_echo		true	0	0
	Mehlbox	Mehlbox	false	0	0
	Sample_Exec_notification	This is a sample notification (sc...	false	0	0

Sample_Exec_notification

EXEC

Config Information

Name: Sample_Exec_notification
Description: This is a sample notification (script is not included)
Expire Interval (sec): 60
Retry Interval (sec): 5
Expired File Size (MB): 1024
Queue Size: 1000

Exec Call Information

Exec Call:

```
notification.sh -id <$IID> -object "<$OBJECT>" -host "<$HOST>" -text
"<$TEXT_1LINE>"
```

Replace Newline Chars with:

Config Information is the same as described above for the email notification service

Exec Call	Enter the script or program that should be called. The working directory is the *boom* server directory.

Both notification types can use following variables to construct message parts or command parameters that will be automatically resolved by the *boom* server:

Variable	Description
<$AGENT_IP>	*boom* agent IP address
<$AGENT_ID>	*boom* agent ID
<$AGENT_HOST>	*boom* agent host name
<$HOST>	Host attribute of the indication
<$APPLICATION>	Application attribute
<$CA1>, …, <$CA15>	Custom Attributes
<$DUPLICATES>	Number of duplicates of indication
<$GROUP>	Group attribute
<$IID>	Indication ID
<$KEY>	Indication Key Attribute
<$AKEY>	Close Mask Attribute
<$OBJECT>	Object attribute
<$SEVERITY>	Severity as text
<$SEVERITY_INT>	Severity as integer

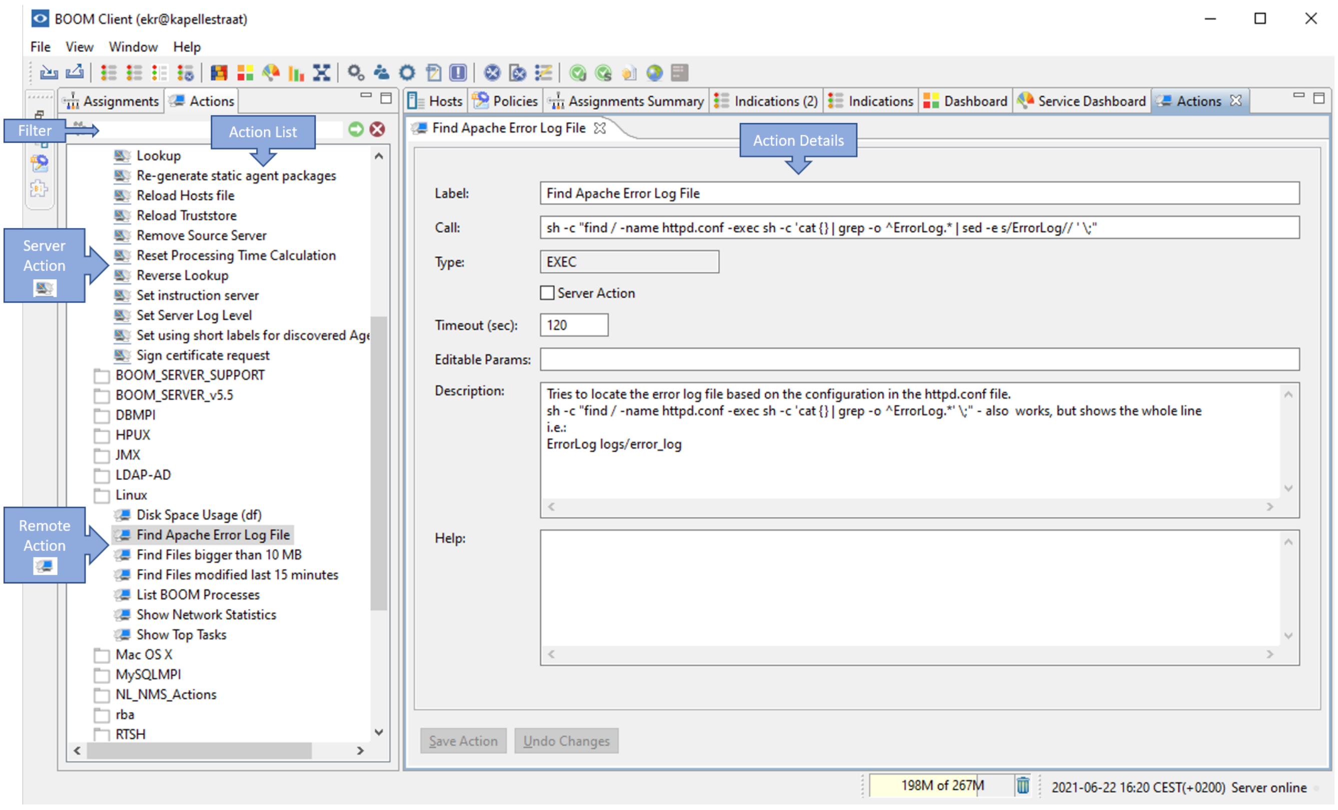
Variable	Description
<$SOURCE>	Specifies the Policy Source e.g. + "Monitor:IS_HTTPMonitor:579204a7-dbc4-4182-a185-61737aa3496e"
<$NAME>	Monitor name (for messages - empty string)
<$VALUE>	Monitor value (for messages = -1)
<$TIME>	Receiving time on the *boom* agent side (milliseconds).
<$STIME>	Receiving time on the *boom* server side (milliseconds)
<$TIME_STR>	Receiving time on the *boom* agent side ("yyyy-MM-dd HH:mm:ss")
<$STIME_STR>	Receiving time on the *boom* server side ("yyyy-MM-dd HH:mm:ss")
<$TYPE_A>	Availability flag (>0 if an availability metric)
<$TYPE_K>	KPI flag (>0 if a KPI metric)
<$TEXT>	Full indication text *
<$TEXT_1LINE>	First line of indication text
<$TEXT_255>	First 256 chars of indication text [0...255] *
<$TEXT_80>	First 81 chars of indication text [0...80] *
<$TEXT_40>	First 41 chars of indication text [0...40] *
<$BOOM_SERVER>	*boom* server host name
<$BOOM_SERVER_IP>	*boom* server IP address

* - new line characters will be replaced with spaces.

6.11. Actions

6.11.1. General Information

The *boom* GUI provides multiple types of actions. This chapter describes the *boom* GUI functionalities of remote actions and server actions in detail.



Label: Find Apache Error Log File

Call: sh -c "find / -name httpd.conf -exec sh -c 'cat {} | grep -o ^ErrorLog.* | sed -e s/ErrorLog// ' \;"

Type: EXEC

[ ] Server Action

Timeout (sec): 120

Editable Params:

Description:

```
Tries to locate the error log file based on the configuration in the httpd.conf file.
sh -c "find / -name httpd.conf -exec sh -c 'cat {} | grep -o ^ErrorLog.*' \;" - also  works, but shows the whole line
i.e.:
ErrorLog logs/error_log
```

Help:

See also the chapter Actions for more information about the action concepts.

6.11.2. Actions Overview

The *boom* GUI provides multiple types of actions. This chapter gives an overview of the available predefined *boom* actions. The actions are stored in different directories dependent on their type. Please define your own action if no predefined action exists for an available remote command.

BOOM_AGENTS

Remote Command (no predefined action)	Description
DISABLE ACTIONS	Disables remote actions on the agent. Use the "add action" function to add your own action. Command: BOOM_AGENT DISABLE_ACTIONS **Attention:** There is no ENABLE_ACTIONS since the agent would reject to execute it once the actions are disabled. See the agent.conf description how to enable remote actions again.
BOOM_AGENT OD2ACN_ON BOOM_AGENT OD2ACN_OFF	Sets OFFER_DATA_TO_ANY_CLUSTER_NODE=<true|false>. This flag allows the agent to offer data to any node from the cluster list (CLUSTER_NODES) that actually heartbeats the agent if the main server is not reachable from the agent (the agent can't send any data to the configured main server but the main server sends heartbeats to the agent).
BOOM_AGENT GET_DETAILS	Gives detailed inventory information about agents e.g. hard disks, network cards etc.
BOOM_AGENT@ENABLE BOMM_AGENT@DISABLE	Remote action to enable/disable agents. Actions can be executed by users without administrator rights.
GET_HB_CTIMEOUT	Get the Heartbeat connection timeout parameter in milliseconds.

Remote Command (no predefined action)	Description
GET_HB_RTIMEOUT	Get the Heartbeat read timeout parameter in milliseconds.
GET_HB_INTERVAL	Get the Heartbeat interval parameter in seconds.
SET_HB_CTIMEOUT	Set the Heartbeat connection timeout parameter in milliseconds.
SET_HB_RTIMEOUT	Set the Heartbeat read timeout parameter in milliseconds.
SET_HB_INTERVAL	Set the Heartbeat interval parameter in seconds.

Action	Description
Disable Policies	Disables a single or multiple policies (does not store this state to the disk)
Enable Policies	Enables a single or multiple policies (does not store this state to the disk)
Get Agent Config File	Show the current configuration settings of the agent.
Get Agent ID	Returns agent’s unique ID. Do not change this field manually. It is an integer value and is created automatically during first clean start of the *boom* agent.
Get Agent IP	Returns agent’s IP address.
Get Agent Log Level	Returns the agent’s current log level. Default log level = 1.
Get Agent PID	Returns agent’s process id.
Get Agent STATUS	The "enabled" parameter shows the agent’s runtime status. In disabled mode the agent only responds to action requests from the server.
Get Agent version	Displays the version of the agent’s software.
Get Agent’s ClusterNodes	Returns configured CLUSTERNODES. CLUSTERNODES - configured hostnames and IPs that can access the agent and perform actions.
Get Environment (Unix)	List the agent’s environment. Only valid for Unix nodes.
Get Environment (Windows)	List the agent’s environment. Only valid for Windows nodes.
Get Message Storm Detection Settings	Returns the active message storm protection settings like: Maximum message per minute: 1000 Blocking period (minutes): 5
List Instrumentation (Unix)	Lists the packages (scripts, executables, configuration files) which were deployed to the agent’s default packet directory %BOOM_ROOT%/spi/.
List Instrumentation (Windows)	Lists the packages (scripts, executables, configuration files) which were deployed to the agent’s default packet directory %BOOM_ROOT%/spi/.
List Policies Status	Returns status of the policies in the format "xxx (version)" where xxx - 3 chars: 1st character: 'E' enabled policy, 'D' - disabled 2nd char: '-' active (not inside deactivation scheduled interval) 'D' not active (! inside deactivation scheduled interval - keeping silence) 3d char '-' PLOG not active (policy specific log) 'P' PLOG active (policy specific log)

Action	Description
Move to a new Server	Agent will be moved to a new *boom* server. After moving the agent sends approval request to the new server but the old server has access to perform actions. If it's necessary perform this action with the old server name to rollback. Ensure that new server has approved the request and delete agent card from the old server. Specify the new server name in the optional parameters field.
Remove Agent's Global Variable	Removes agent's global variable. To print a list of defined variables execute the action with an empty optional parameters field.
Set Agent Log Level	Sets agent's log level (use 1..4).
Set Agent Log Limit	Sets LOGCOUNT and LOGSIZE values for logging facility of the *boom* agent. LOGCOUNT - defines maximal number of logfiles LOGSIZE - defines max size in MB of each logfile
Set Agent's ClusterNodes	Replaces configured CLUSTERNODES. CLUSTERNODES - configured hostnames and IPs that can access the agent and perform actions.
Set Agent's Global Variable	Defines a new global variable on the selected agent. A global variable can be used in any deployed policy. To print a list of defined variables execute the action with an empty optional parameters field.
Set Message Storm Detection settings	Sets the MaxMessagesPerMinute and BlockTimeMinutes parameters. For more information see chapter Message Storm Detection
Show Last Monitor Values	Shows the last values that the agent has received for each policy

BOOM_AGENTS_SUPPORT

Action	Description
Clear Agent Indication Buffer	Clear all buffered indications on the Agent
Clear Agent Monitor values Buffer	Clear all buffered monitor values on the Agent
Clear Agent Performance data Buffer	Clear all buffered performance data on the Agent
Emulate Java Monitor Call	Executes the specified java monitor call and prints the results
Reset Threshold States	Flushes the threshold states for all monitors. Basically this will cause the agent to resend indications for all monitors that exceed a threshold just as if the threshold is crossed the first time.
Set Agent→ Server communication OFF	Switches the selected agent to a passive, listen only mode. The agent won't actively send any heartbeats and data to the server. The server will take over the responsibility to send heartbeats and poll the data from the agent.
Set Agent→ Server communication ON	Switches the selected agent to an active mode. The agent will actively communicate with the server. The agent tries to recognize if the server is online, offline or firewalled and submits data to the server.
Set AgentID-based communication OFF	Instructs the selected agent NOT to use its AgentID for the communication with the *boom* server. The AgentID is only necessary for NAT based environments.

Action	Description
Set AgentID-based communication ON	Instructs the selected agent to include its ID in all communication to the *boom* server, so that the server can differentiate the agents behind a NAT device.
Set PLOG OFF	Sets policy specific logging off. Please see policy specific logfiles
Set PLOG ON	Sets policy specific logging on. Please see policy specific logfiles
Trigger Policy Now	Will trigger the specifed policy to run on the agent

BOOM_Server

Command (no predefined action)	Description
AGENTS_STATUS	Lists the inventory information of all agents. Use the "add action" function to add your own action. Command: AGENTS_STATUS

Action	Description
Add License	Add a new license to the *boom* server.
Add new Source Server	Adds a new Slave Server. The server, user, password and scenario must be specified. This action requires a valid boom2_boom_ license
Get Environment	Lists the server environment.
Get instruction Server	Gets the instruction server URL. This URL is used to replace the variable <$INSTRUCTION_SERVER> for each incoming indication with the variable defined in the Instruction URL field.
Get monitored hosts	Returns a list of all monitored hosts. According to the indication field "Host".
Get Server Log Level	Returns the server’s current loglevel.
Get Server PID	Returns the *boom* server PID.
Lookup	Returns the resolved IP of the specified host name. Please note - this action will use the java lookup mechanism and allows to check how the server will resolve an agent name.
Re-generate static agent packages	This action (re)generate all agent packages to /srv/deploy/pkg directory. List of available static packages are available under: https://your.boom.server/deploy
Reload Hosts file	Reloads <boomServer>/etc/hosts file from the disk.
Remove Source Server	Removes Source Server.
Reset Processing Time Calculation	Resets Max Processing Time status variable for the Statistics Calculation.
Reverse Lookup	Returns the resolved hostname of a given IP. Please note - this action will use the java lookup mechanism.

Action	Description
Set instruction Server	Sets the instruction server URL (HTTP/HTTPS) for instructions defined in the policies. The variable <$INSTRUCTION_SERVER> will be replaced with the specified string for each incoming indication with the variable defined in the Instruction URL field. The value will be stored in boom.props
Set Server Log Level	Set loglevel of the *boom* server. Use [1…4]
Set using short labels for discovered Agent cards	If true - server will use the short host name as label for each new connected Agent. The value will be stored in the boom.props

BOOM_Server_Support

Action	Description
DB disable	Puts the *boom* database into virtual "offline" mode. Means, the *boom* server stops writing data into the database. The server will buffer all data and needs to have enough free memory. !!! Please don’t use this action unless you know what you are doing !!!
DB enable	Switches the *boom* database back to normal mode.
DB Perf disable	Puts the *boom* performance database into virtual "offline" mode. Means, the *boom* server stops writing data into the database. !!! Please don’t use this action unless you know what you are doing !!!
DB Perf enable	Puts the *boom* performance database back to normal mode.
Disable performance data processing	Force the *boom* server to ignore incoming performance data from Agents. All incoming performance data from agents will be dropped.
Enable performance data processing	Enable processing and storing of performance data from the agents.
Get heartbeats connection timeout	Gets current connection timeout used by heartbeats.
Get heartbeats interval	Gets current interval used by heartbeats. (Interval between heartbeats in seconds)
Get heartbeats SO_TIMEOUT	Gets current socket operations timeout used by heartbeats.
Set (force) Agent online/offline	Force Agent status to online or offline. !!! Please don’t use this action unless you know what you are doing !!!
Set heartbeats connection timeout	Sets Socket connection timeout for the heartbeat tasks. Default value is 3000 milliseconds. The value will be stored in boom.props
Set heartbeats interval	Sets interval for heartbeat tasks. The value will be stored in boom.props
Set heartbeats SO_TIMEOUT	Sets Socket operations timeout for the heartbeat tasks. Default value is 10000 milliseconds. The value will be stored in boom.props

Action	Description
Set maximum Agent threads	Sets maximum threads used by server for communication with Agents. (please don’t change if you are not sure what it is). The value will be stored in boom.props
Set maximum deployment threads	Sets maximum number of threads used by server for deployments. (please don’t change if you are not sure what it is) The value will be stored in boom.props
Set maximum UI threads	Sets maximum number of UI Threads used by server. Can not be less then 10. Default value is 200. Please note, that each connected UI can use more than one server thread. The value will be stored in boom.props

HPUX

Action	Description
Cluster Package Information	Shows which cluster packages are running on which cluster node.
Disk Utilization (bdf)	Shows the disk utilization on a specified HP-UX node.
Net Interfaces Statistics	Executes netstat -i on a specified HP-UX node.
SAR CPU & Swap Queue	sar -q checks the queue activity. Requires the sar (System Activity Reporter) utility.
SAR CPU Load	sar -u shows the CPU utilization. Requires the sar (System Activity Reporter) utility.
SAR Disk Activity	The output of sar -d shows various disk-related statistics. Requires the sar (System Activity Reporter) utility.
SAR Disk IO	Reports disk I/O and transfer rate statistics. Requires the sar (System Activity Reporter) utility.
SAR Process, Inode and File Table Utilization	sar -v checks the system table status. Requires the sar (System Activity Reporter) utility.
SAR Swap Activity	Displays the RAM memory switching and swapping activities. Requires the sar (System Activity Reporter) utility.
SAR System Calls	Displays process creation activity. Requires the sar (System Activity Reporter) utility.
Swap Utilization	Prints information about device and file system paging space.
Volume Groups	Displays LVM Volume Group summary and detail information.

JMX (Java Management Extensions)

Action	Description
Jmx check Memory	Returns memory usage for objects with type=Memory.
Jmx check MemoryPool (Perm Gen only)	Returns memory usages for Permanent Generation. Filtered by attribute=Name and valueMask=Perm*: … -o java.lang:*,type=MemoryPool -a Name -v Perm*

Action	Description
Jmx check MemoryPools	Returns memory usage for MemoryPools (not filtered). ... -o java.lang:*,type=MemoryPool
Jmx check Servlets	... -o :,j2eeType=Servlet

LDAP-AD

Action	Description
LDAP Add User	Adds an LDAP user to the *boom* server. Multiple LDAP users can be added (separated by space). Specify the new user in the optional parameters field: e.g. newuser@boomserver1.mycompany.de
LDAP Delete configuration	Will delete the LDAP configuration for a given URL.
LDAP Import Users	Import users from Active Directory. To limit the import to a certain group of users, add the Common Name Hierarchy as LDAP Search filter: e.g. LDAP_SEARCH_FILTER=(?
LDAP List configurations	Returns a list of LDAP servers configured in the *boom* server. Select a *boom* server int the server field.
LDAP Reload	Reload the configured LDAP/Active Directory Server.
LDAP Save configuration	Will create the LDAP configuration for a given URL. Optional parameters can be: LDAP_BASE=base LDAP_USER_ATTR= LDAP_EXPIRE= to re-authorize users
LDAP Test User	Test user credentials against a given LDAP URL. (!) This action makes independent direct request to the LDAP server (*boom* server configuration ignored). Optional parameters: LDAP_BASE=base LDAP_USER_ATTR=userAttribute

Linux

Action	Description
Disk Space Usage (df)	Displays free disk space for all your mounted filesystems in 1K blocks.
Find Apache Error Log File	Tries to locate the error log file based on the configuration in the httpd.conf file.
Find Files bigger than 10 MB	Find files bigger than 10 MB on the specified Linux host.
Find Files modified last 15 minutes	Finds all files that were modified during the last 15 minutes.
List *boom* Processes	List all *boom* processes running on the specified system.
Show Network Statistics	Display summary statistics for each protocol of the specified node.
Show Top Tasks	Display the tasks consuming most CPU.

MAC OS X

Action	Description
Kernel IO Statistics	Display Kernel IO Statistics on the specified host.
Net Errors	Display Statistics from the network devices.
Show launched Services	List all services that are registered with launchd.
Swap Pageouts	Report page-out activity (the number of pages paged out).
Virtual Memory Statistics	Display virtual memory statistics.

SNMP

Action	Description
SNMPWalk	SNMP walk implementation.

SSH

Action	Description
Remote SSH Command	This action requires that the SSH package was deployed. Allows you to execute a command on a remote host via a SSH connection. Select a node and complete the optional parameter field. e.g. boomnode.mycompany.de root pwd "ps -ef

SunOS

Action	Description
Disk Utilization (df)	Display the disk utilization for all mounted filesystems. Requires the sar (System Activity Reporter) utility.
SAR CPU ?	sar -q checks the queue activity. Requires the sar (System Activity Reporter) utility.
SAR CPU Load	sar -u shows the CPU utilization. Requires the sar (System Activity Reporter) utility.
SAR Disk Activity	The output of sar -d shows various disk-related statistics. Requires the sar (System Activity Reporter) utility.
SAR IPC	sar -m checks the interprocess communication. Requires the sar (System Activity Reporter) utility.
SAR System Table Status	sar -v checks the system table status. Requires the sar (System Activity Reporter) utility.

Unix

Action	Description
List *boom* Processes	List all *boom* processes running on the specified system.
Print System Information	List basic information currently available from the system.
Show Free Diskspace	Shows the statistics for the available file systems.

Action	Description
Show Listening Ports	List all ports that are currently in state LISTEN. Requires the lsof utility.
Show Network Statistics	Shows per-protocol network statistics.
Show Processes	Show all processes that are running on the system.
Show Routing Configuration	Displays the routing table.
Show Top 10 busy Processes	Displays the top 10 busy processes in descending order (most busy process at the top): PID CPU% MemSize(Kb) Command Arguments
System Utilization	Displays the current system utilization

Windows

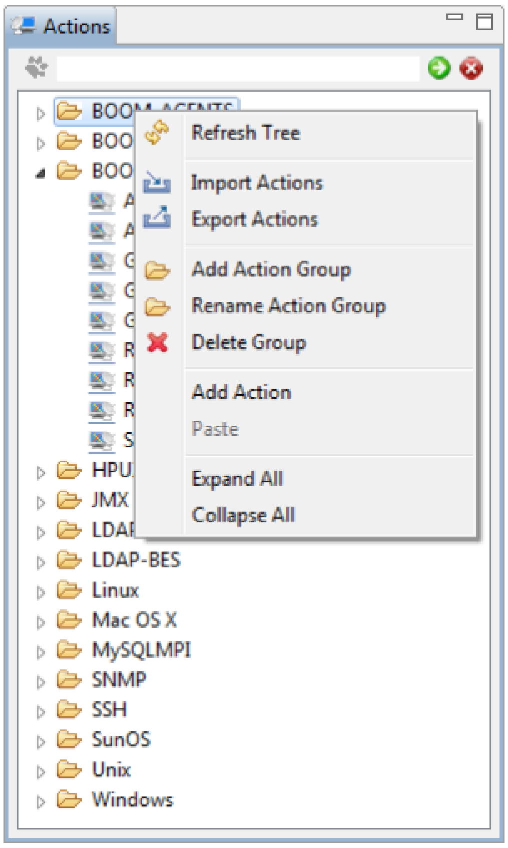
Action	Description
Available Shares	Displays information about all of the resources that are shared on the specified host.
Show Hostname	Displays the windows host name.
Show IP Configuration	Displays the IP configuration details of all available adapters.
Show Network Connections	Displays all currently open network connections.
Show Network Statistics	Displays the ethernet statistics of the specified host.
Show Processes	Displays all processes that are running on the specified host.
Show Remotely Open Files	Displays the names of all open shared files on the specified host and the number of file locks. You can use NET FILE to close individual files and release the locks. Use net file /close to close an open file.
Show Routing Configuration	Displays the routing table of the specified host.
Show running Services	Displays a list of services that are currently running.
SMB Client Information	Displays the configuration for the workstation service.
SMB Client Statistics	Displays statistics for the workstation service.
SMB Server Information	Displays the configuration for the server service.
SMB Server Statistics	Displays statistics for the server service.
View Sessions	Displays information about all sessions with the local computer.

See also chapter Actions for more information about Actions.

6.11.3. Action Operations

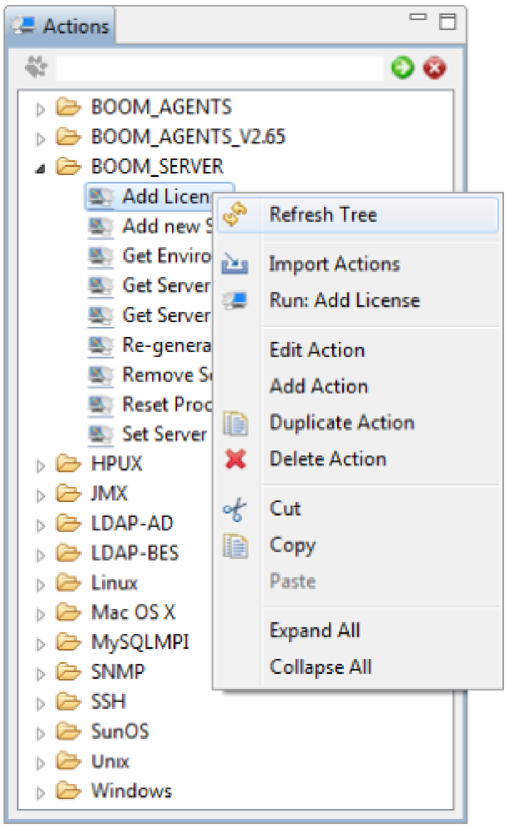
Action groups are used to group actions together as needed. An action group can contain both remote and server actions, but no subfolders! All action specific functions can be found in the context menu (right-click on a list element):

Action group context menu:



Rename Group:	Rename the selection action group.
Import Actions:	You have the possibility to import actions from an XML file.
Export Actions:	This will export all actions to an XML file.
Rename Group:	Rename an existing action group.
Add Group:	Create a new action group.
Delete Group:	The action group with all actions inside will be removed from the action list.
Add Action:	A new action will be opened in the 'Action Details View'. All changes to to the action specific attributes have to be saved before running the action.

Action context menu:



Import Actions:	You have the possibility to import actions from an XML file.
[Run: <action_name>]:	This opens the 'Run Action Dialog' (see below) from where the selected action can be executed.
Edit:	The action will be opened in the 'Action Details View' where all action specific attributes can be changed.
Add Action:	Creates a new action. This new action will be opened in the 'Action Details View' where all action specific attributes can be specified.
Duplicate Action:	If you want to create a new action based on an existing one.
Delete Action:	Deletes the selected action from the tree.

6.11.4. Action Details

The 'Action Details View' contains all action specific data. All changes that are made to an actions have to be saved before executing the action: The following attributes can be specified:

Label	Action name
Call	The Action Command. If the Call is started with BOOM_AGENT an agent internal command is performed, if the call is started with the word "IAction" an JAVA_ACTION is performed.
Type	Type is by default EXEC.
Server Action	Flag indicates an action that can be executed on the *boom* server.

Timeout (sec)	Defines the timeout in seconds.
Editable Parameters	Optional call parameters.
Description	Description of the action.
Help	Help text for the action.

6.11.5. Run an Action

boom provides several ways to start an action.

- Open the **Actions** view in the GUI:
  To execute an action select **Run:<action>** in the action context menu or double-click on the action itself in the action list. The **Run Action** Dialog opens before the action is executed. In case of a remote action you have to select the *boom* agent where you want to run the action. Any changes that are made in this dialog will not be saved! Changing the action details can only be done in the **Action Details** View (select the "Edit" function from the context menu).
- Select the **Hosts** view in the GUI: To execute an action select one or more agents in the **Hosts** view. Right-click to open the context menu. Select **Run Action on selected …** to open the **Select Agent Action** Dialog. Select the action you want to execute.
- Select the **Agent Overview** view in the GUI: To execute an action select one or more agents in the **Agent Overview** view. Right-click to open the context menu. Select **Run Action on selected …** to open the **Select Agent Action** Dialog. Select the action you want to execute.

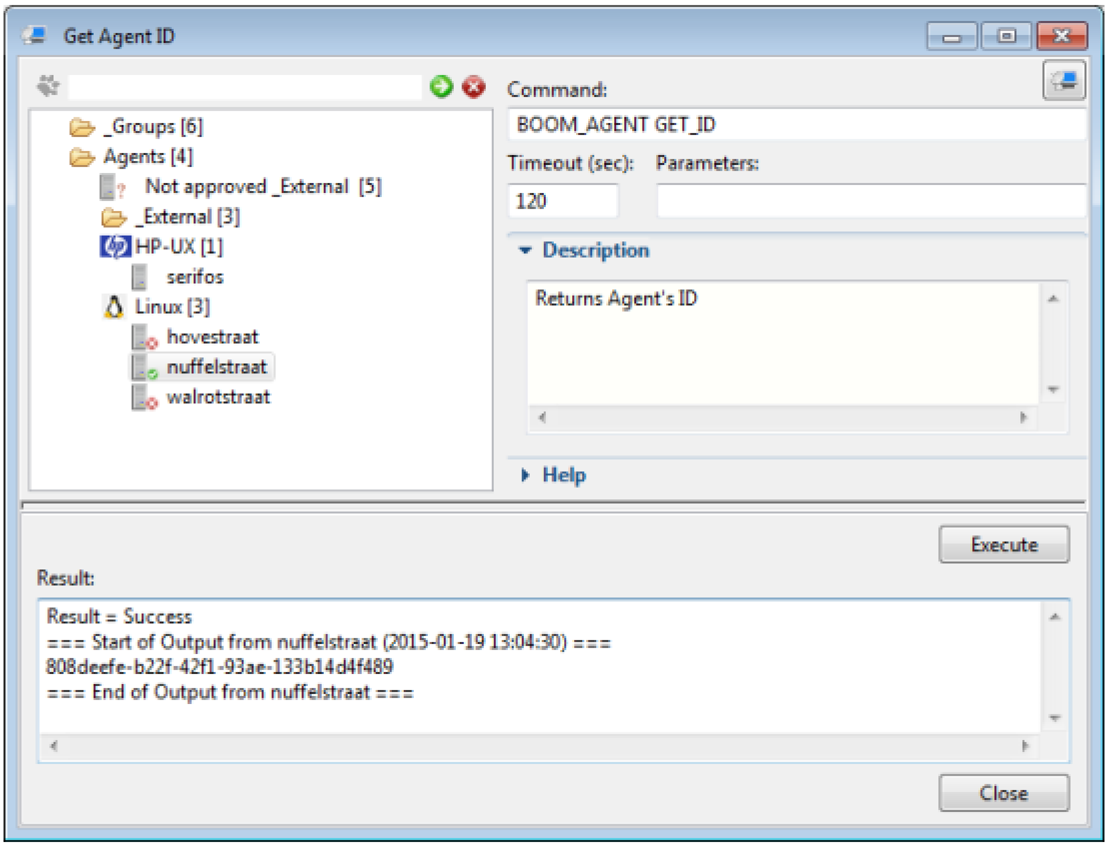
'OWNER' rights are necessary to be able to create, edit or delete an Action. To be able to execute an Actions the user needs at least 'GUEST' rights. A user with GUEST rights is allowed to change the Parameters field of the Action.

The agent sets the following environment variables that can be used in an action:

BOOM_AGENT_HOST	the agents host name.
BOOM_AGENT_ID	the agent's unique ID (e.g. 07f24f82-2d44-4c4c-98c0-1ac7e3814a77).
BOOM_AGENT_IP	the agent's IP address.
BOOM_AGENT_PID	the agent's process id.
BOOM_AGENT_PORT	the port number the agent is listening on (i.e. 23021).

6.11.6. Remote Action

A remote action, which can be any script or executable, can be executed on any remote system where the *boom* agent is installed.



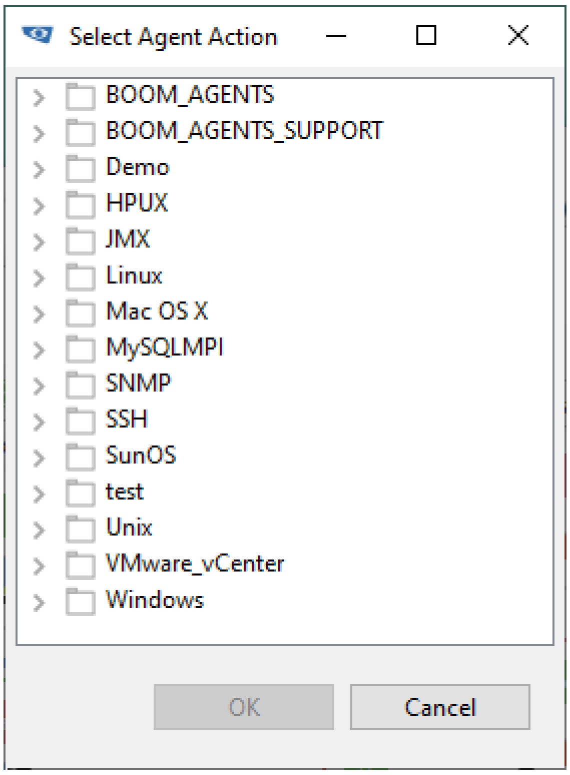
Command: BOOM_AGENT GET_ID

Timeout (sec): 120

Returns Agent's ID

```
Result = Success
=== Start of Output from nuffelstraat (2015-01-19 13:04:30) ===
808deefe-b22f-42f1-93ae-133b14d4f489
=== End of Output from nuffelstraat ===
```

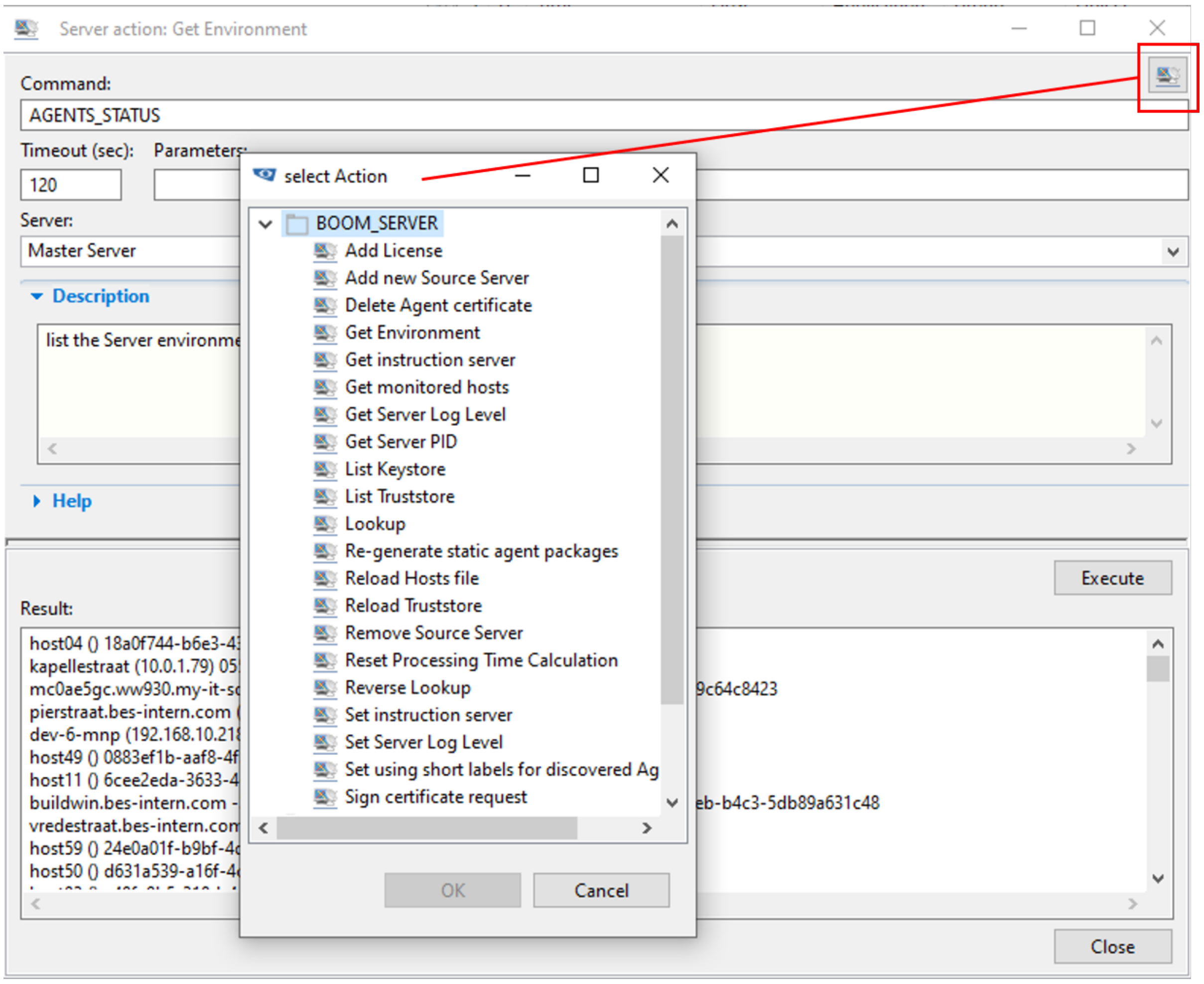
Use the select Action button (top right) to open the select Action view. Select a further remote action you want to execute.



6.11.7. Server Action

If an action is defined as a "Server Action" it will always be executed on the *boom* server. In this case no *boom* agent needs to be selected in the 'Run Dialog'.

Use the "select Action" button (top right) to open the "select Action" view. Select a further server action you want to execute.



```
host04 () 18a0f744-b6e3-4
kapellestraat (10.0.1.79) 05
mc0ae5gc.ww930.my-it-so
pierstraat.bes-intern.com (
dev-6-mnp (192.168.10.21
host49 () 0883ef1b-4f
host11 () 6cee2eda-3633-4
buildwin.bes-intern.com -
vredestraat.bes-intern.com
host59 () 24e0a01f-b9bf-4c
host50 () d631a539-a16f-4
```

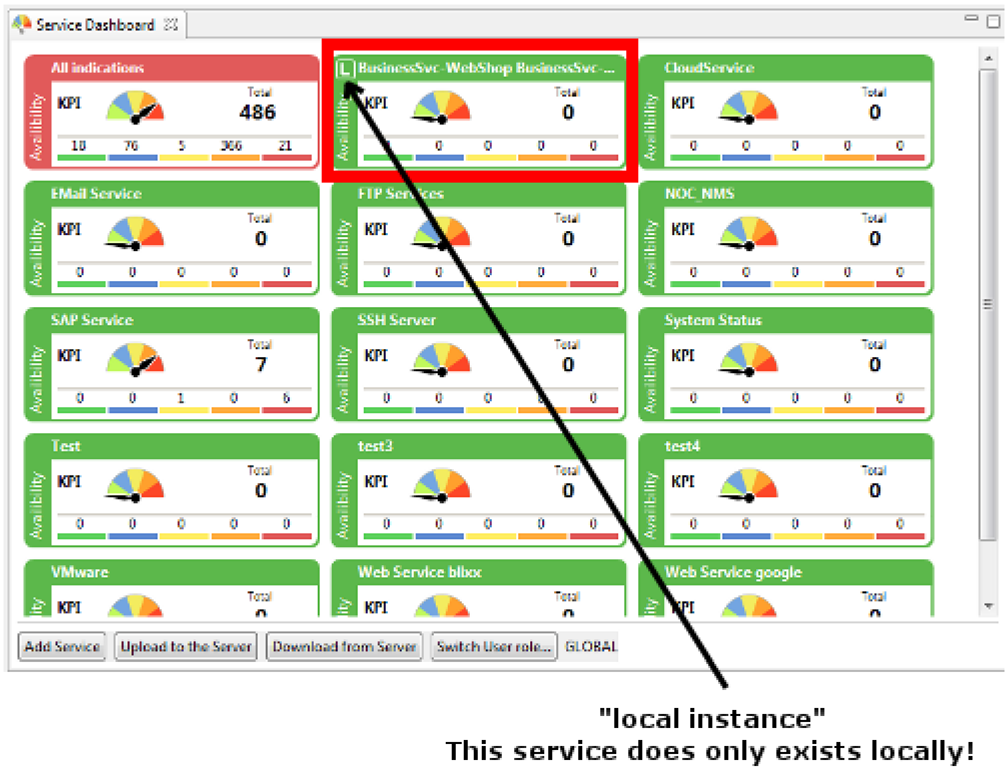
```
9c64c8423
eb-b4c3-5db89a631c48
```

6.12. Service Dashboard

6.12.1. General Information

The 'Service Dashboard' provides a quick overview about the 'Service Availability' and KPI state.
Configuring the dashboard is very simple. A set of filters defines which indications might impact a certain service. In the according monitor policy conditions the flags 'Availability' and KPI differentiates indications that are relevant to 'Service Availability', 'Service KPI' (the monitored value or message has impact on an SLO), and not service relevant general indications. Use the IS MPI monitor policies as configuration example.

For each service three sections are displayed: The Availability section shows the maximum severity of all filtered indications with set Availability flag. The KPI section shows the weighted average of all filtered KPI-flagged indications. The last section lists the total counts of all filtered indications independent from the flags.



6.12.2. Local/Global Services

The *boom* Service Dashboard is based on global and local definitions. The server-side definitions (global definitions) create a general list of available services for all *boom* users. Users have the possibility to create local service definitions as add-on to the global list or to override any global service definitions for their own needs. The local configurations provide the possibility to have flexible and independent definitions allowing to implement team specific service views or user and company POIs.

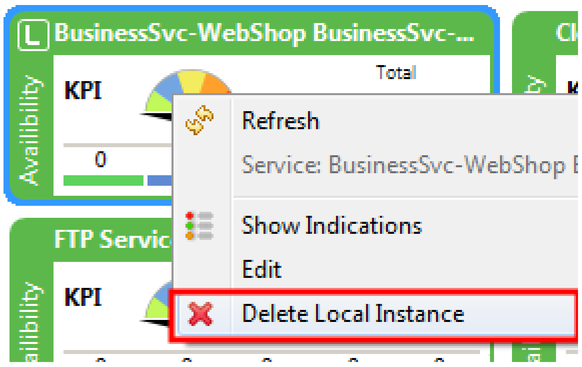
Changes that are done in the user interface are not reflected in the server side global service definitions as long as they are NOT explicitly uploaded to the server. Local service definitions take precedence over the global definitions, so users will always see services based on their local definitions. Such local service definitions and overrides will be flagged accordingly.

Service Operations

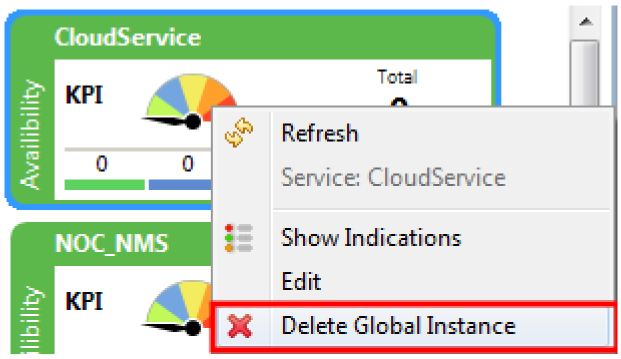
Add Service:	This will create a new service. After the creation a set of filters/filter groups need to be added to this service. Please note: A new created service will be saved locally. To make the service a global service you have to upload the service definition to the *boom* server (see below). For more information see "Add Dashboard Service" below.
Upload to the Server:	A new created service is saved locally. To make the service a global service you have to upload the service to the server. Only user with "Owner" rights on "Service Dashboard" have permissions to change the global service definitions.
Download from the Server:	This will download all global services to the local system. The Local Services will NOT be touched.

Switch User role...:	Select another user role.

Right-click on a Service to open the context menu:



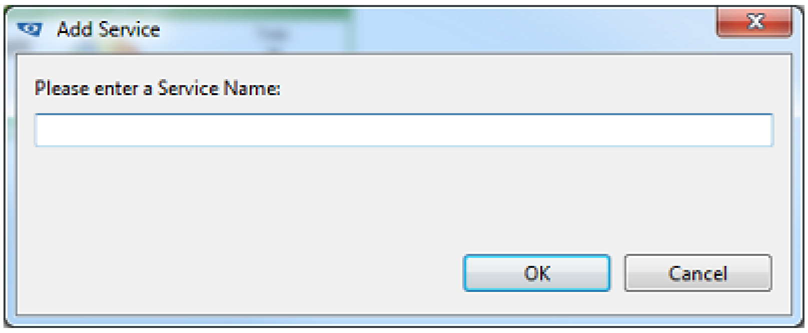
Show Indications:	This will open a new indication tab displaying all active indications for the selected service. → You can also double-click on a service to display all appropriate indications.
Edit:	The service details can be changed in the service dialog. This is the same dialog that is used to create a new service. For more information about the service dialog see "Add Dashboard Service" below.
Delete local instance:	This will delete the local instance of the selected service.



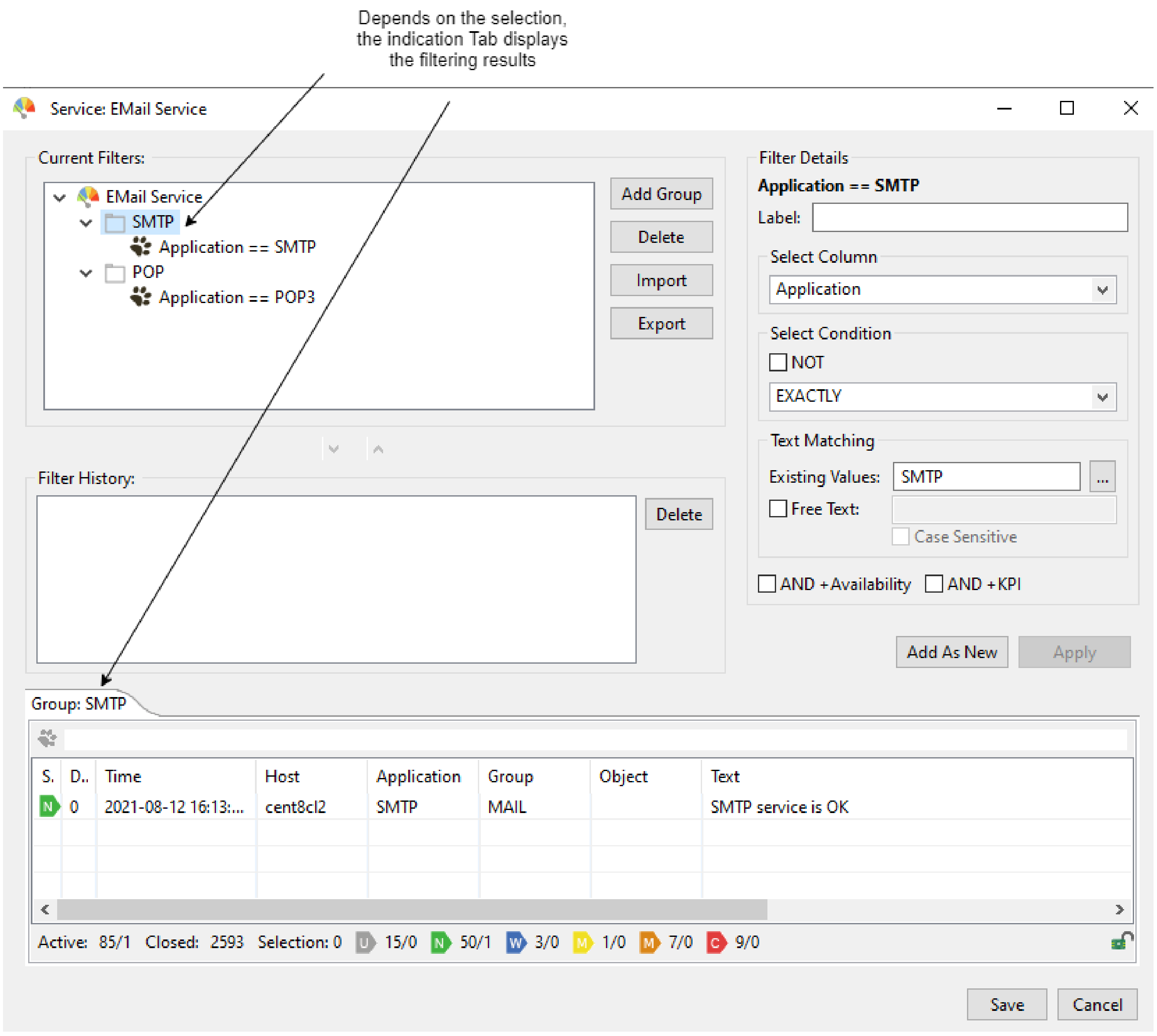
Delete global instance:	This will delete the global instance of the selected service. Only user with "Owner" rights on "Service Dashboard" have permissions to change the global service definitions.

6.12.3. Add Dashboard Service

If you want to create a new service, you first need to provide a service name:



A new service object with the given name will be created and opened in the 'Service Dialog':



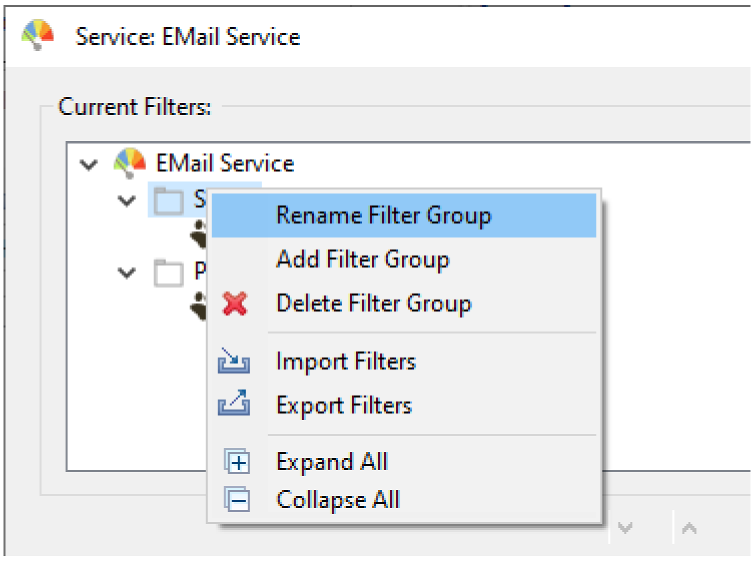
S.	D..	Time	Host	Application	Group	Object	Text
N	0	2021-08-12 16:13:...	cent8cl2	SMTP	MAIL		SMTP service is OK

Active: 85/1 Closed: 2593 Selection: 0 U 15/0 N 50/1 W 3/0 M 1/0 M 7/0 C 9/0

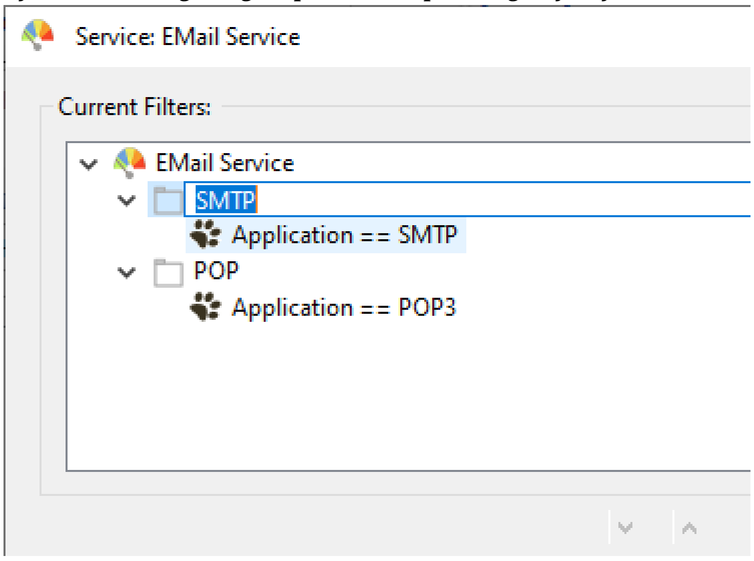
Please note:

- A filter can only be added to a group and not directly to a folder. The new created service does not have any groups or filters. In order to add filters to the service you have first to create a group!
- The filter results will be displayed in the indication tab at the bottom. The results are displayed depending on the selection. Select a group to see all indications matching to the filter belonging to this group. Select the service itself to see all indications matching to all filters belonging to this service.
- Any filter group uses boolean **'AND'** between defined filters. In other words: A filter group matches with an indication only if all filters in this group are matching with the given indication.
- A service uses boolean **'OR'** between filter groups. In other words: An indication matches with a service if at least one filter group matches with the given indication.

Right-Click on a Filter Group to open the content menu:



Rename Filter Group: The name of the filter group can be changed. You can also change the group name by first selecting the group and than pressing any key.



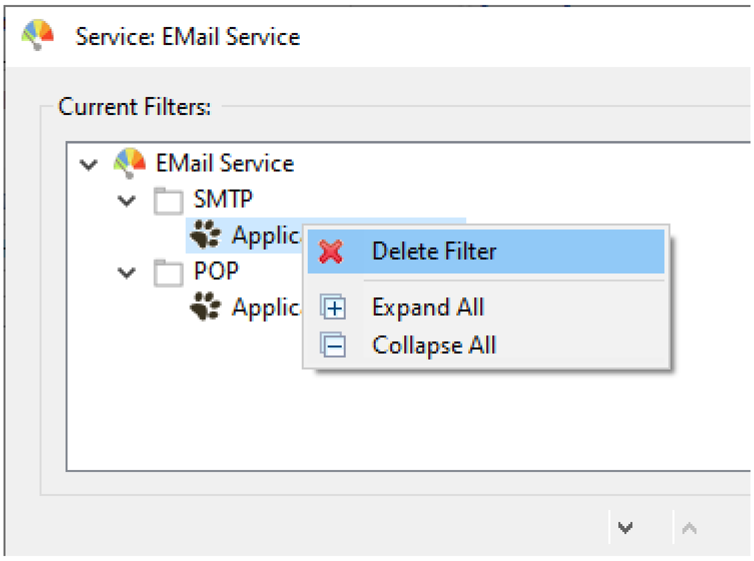
Add Filter Group: A filter can only be added to a group and not directly to a service. Therefore a service must have at least one group.

Delete Filter Group: This will delete a group with all its filters.

Import Filters: Import Filters from an XML file.

Export Filters: Export Filters to an XML file. Please note: Only filter groups can be exported.

Right-Click on a Filter Group to open the content menu:

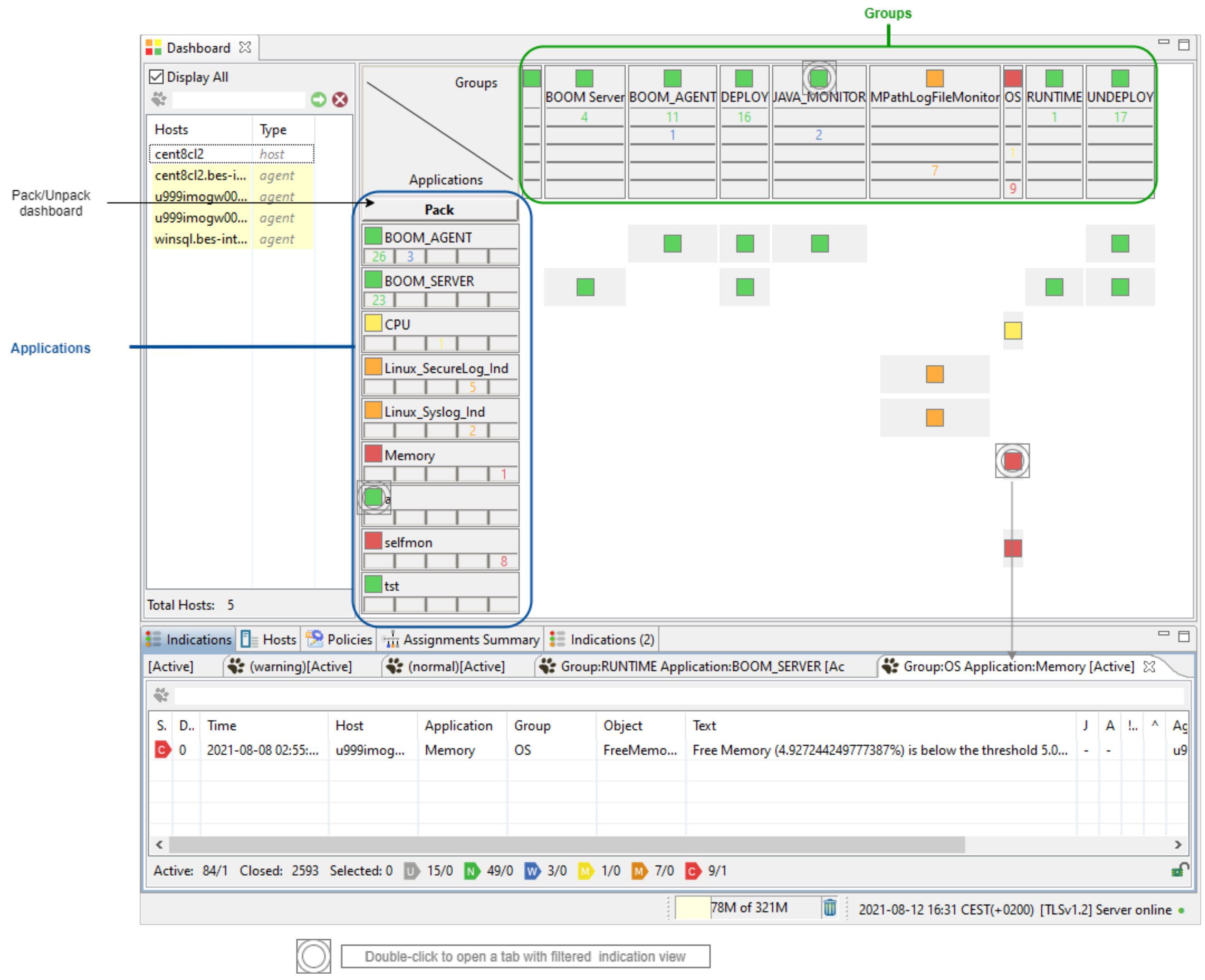


Delete Filter: This will delete the selected filter.

6.13. Operator Dashboard

General Information

The operator dashboard gives an quick overview about the actual state and problem areas. Its graphical representation shows at one glance the overall status of applications and groups, allowing to quickly identify hot spots in the environment and problem impacts.



S.	D..	Time	Host	Application	Group	Object	Text	J	A	!..	^	Ag
C	0	2021-08-08 02:55:...	u999imog...	Memory	OS	FreeMemo...	Free Memory (4.927244249777387%) is below the threshold 5.0...	-	-			u9

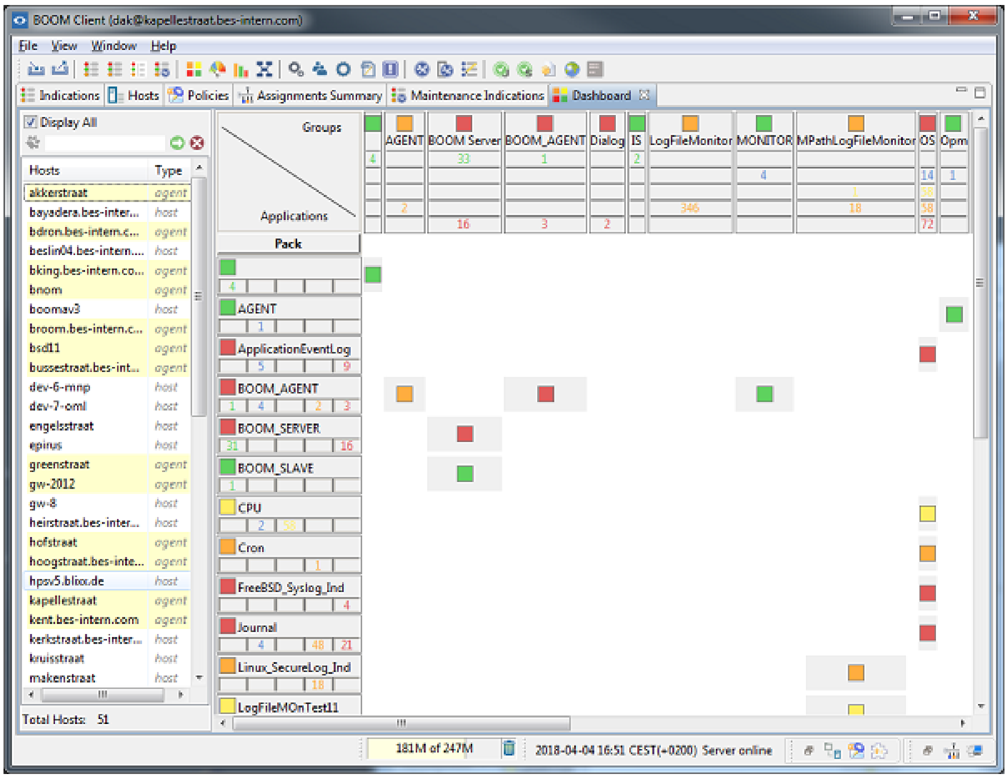
Active: 84/1 Closed: 2593 Selected: 0 U 15/0 N 49/0 W 3/0 M 1/0 M 7/0 C 9/1

78M of 321M 2021-08-12 16:31 CEST(+0200) [TLSv1.2] Server online

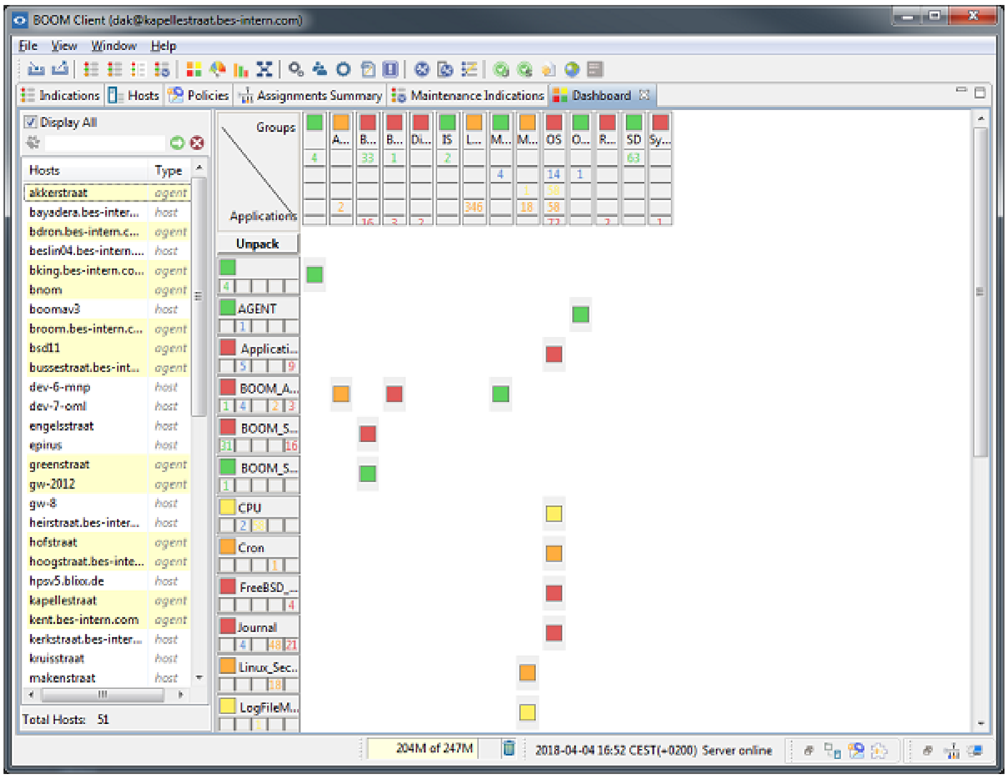
Pack/Unpack Dashboard

Packing of the dashboard will limit the width of the displayed group columns reducing the dashboard width.

Unpacked Operator Dashboard


BOOM Client (dak@kapellestraat.bes-intern.com)
File View Window Help
Indications Hosts Policies Assignments Summary Maintenance Indications Dashboard
Display All
Groups
Applications
Pack
AGENT BOOM Server BOOM_AGENT Dialog IS LogFileMonitor MONITOR MPathLogFileMonitor OS Opm
AGENT
ApplicationEventLog
BOOM_AGENT
BOOM_SERVER
BOOM_SLAVE
CPU
Cron
FreeBSD_Syslog_Ind
Journal
Linux_SecureLog_Ind
LogFileMOnTest11
Hosts Type
Total Hosts: 51
181M of 247M
2018-04-04 16:51 CEST(+0200) Server online

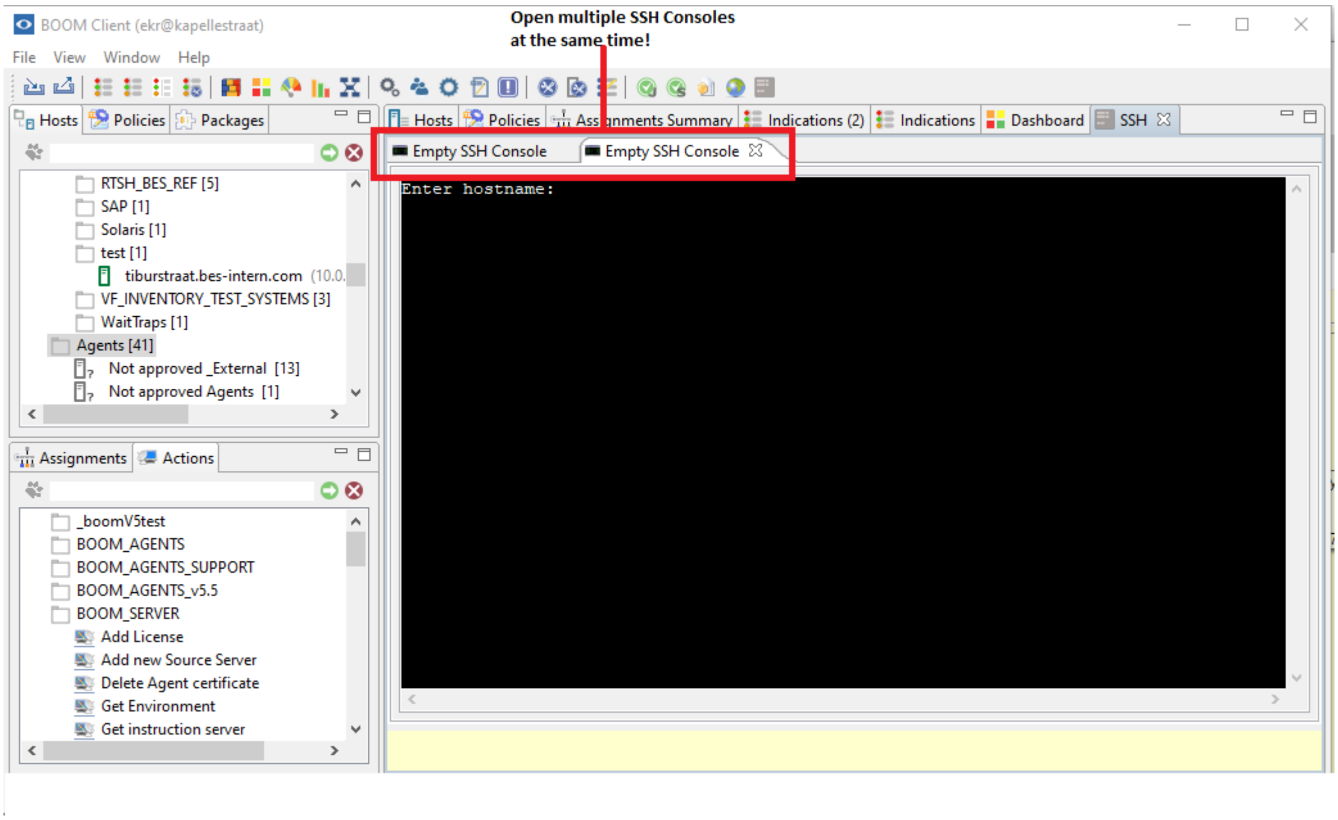
Packed Operator Dashboard


BOOM Client (dak@kapellestraat.bes-intern.com)
File View Window Help
Indications Hosts Policies Assignments Summary Maintenance Indications Dashboard
Display All
Groups
Applications
Unpack
Hosts Type
Total Hosts: 51
204M of 247M
2018-04-04 16:52 CEST(+0200) Server online

6.14. SSH Terminal

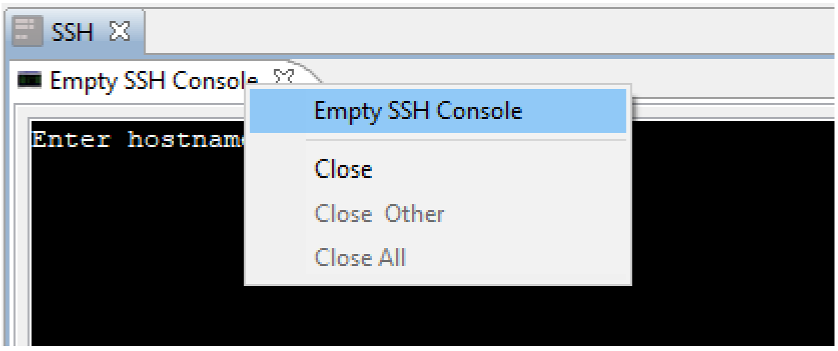
General Information

Build-in ssh terminal allows an operator to check remote systems or to make some corrective actions without switching to another application.



```
Enter hostname:
```

Right-click on the tab heading to open the context menu:



```
Enter hostname
```

Empty SSH Console:	This will open a new SSH Console Tab

6.15. Server Jobs

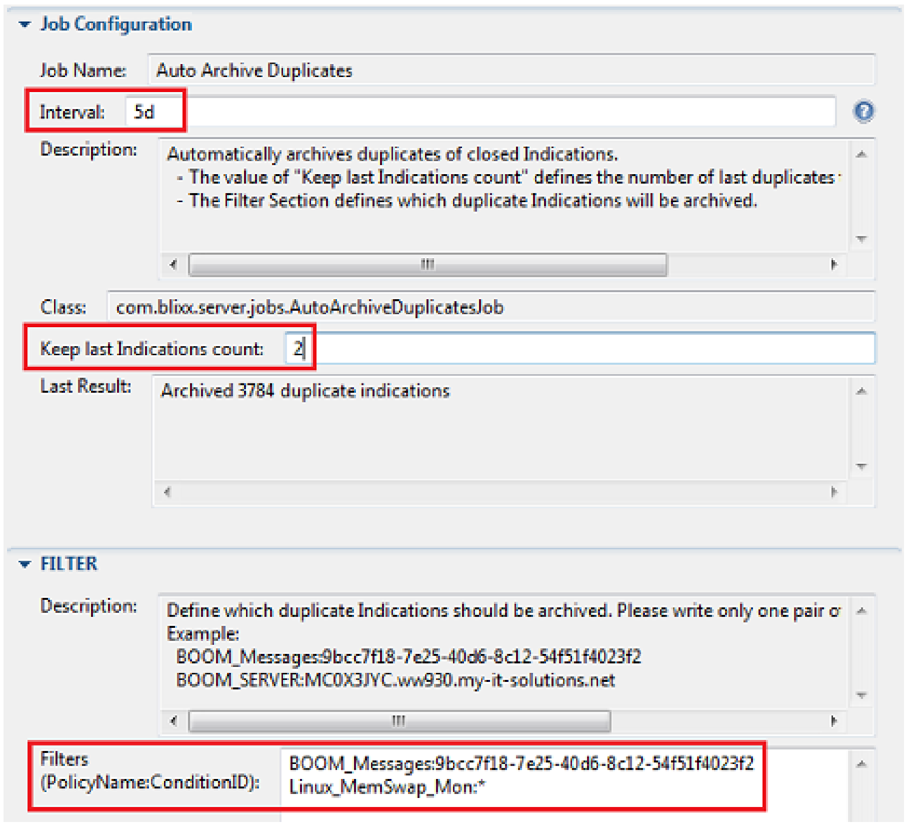
6.15.1. General Information

The *boom* server provides several jobs to automatically check and cleanup the system. According to your individual requirements these jobs can be enabled/disabled or modified.

The following **Server Jobs** are available:

Auto Archive Duplicates

Automatically identifies and archives duplicate indications for the closed indication browser. With this job you can reduce the number ob closed indications. For example if you want to reduce the online/offline Agent indiations+ One example is to reduce online/offline Agent indications.


Job Configuration

Job Name: Auto Archive Duplicates

Interval: 5d

Description: Automatically archives duplicates of closed Indications.
- The value of "Keep last Indications count" defines the number of last duplicates
- The Filter Section defines which duplicate Indications will be archived.

Class: com.blixx.server.jobs.AutoArchiveDuplicatesJob

Keep last Indications count: 2

Last Result: Archived 3784 duplicate indications

FILTER

Description: Define which duplicate Indications should be archived. Please write only one pair o
Example:
BOOM_Messages:9bcc7f18-7e25-40d6-8c12-54f51f4023f2
BOOM_SERVER:MC0X3JYC.ww930.my-it-solutions.net

Filters (PolicyName:ConditionID): BOOM_Messages:9bcc7f18-7e25-40d6-8c12-54f51f4023f2
Linux_MemSwap_Mon:*

Serverjob **configuration examples**:

Archive duplicate closed indications for a special conditionID → **<PolicyName>:<CondID>**

```
BOOM_Messages:9bcc7f18-7e25-40d6-8c12-54f51f4023f2
```

Archive duplicate closed indications sent by the boom server for a special node → **BOOM_SERVER:<nodeName>**

```
BOOM_SERVER:MC0X3JYC.ww930.my-it-solutions.net
```

Archive ALL duplicate closed indications sent by the boom server:

```
BOOM_SERVER:*
```

Auto Archive Indications Job

Automatically archives closed indications older than specified amount of "Max days". Value of "Max days" must be bigger 0. To deactivate this task set "Max days" value to 0 or -1.

Auto Close Indications Job

Automatically closes active indications older than specified amount of "Max days". Value of "Max days" must be bigger 0. To deactivate this task set "Max days" value to 0 or -1

Auto Delete Indications

Automatically deletes archived indications older than specified amount of "Max days"

CleanPerf DB

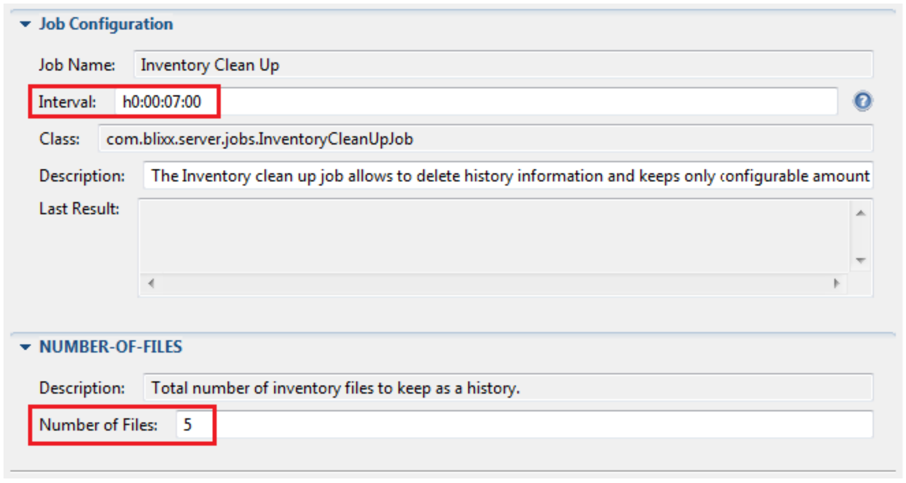
Cleaning of the *boom* performance database. "Keep maximum history of (days)" parameter specifies oldest time to keep in DB. A logfile is available on:
`<server_install_dir>/<configured working directory>/CleanBoomPerfDB_<date>_<number>.log`

Find_Lost_Policies

Automatically detects lost policies and adds them to the policy tree folder "_Lost and found". e.g. If the server job detects a policy which is not known in his policyGroups.xml file then the policy will be added to the _Lost and Found folder. A user can decide whether the policy should be deleted or copied to another folder.

Inventory Clean Up Job

The Inventory Clean Up server job allows to "delete historical versions" of inverntory files and keeps only configurable amout of versions per Agent.


Job Configuration

Job Name: Inventory Clean Up

Interval: h0:00:07:00

Class: com.blixx.server.jobs.InventoryCleanUpJob

Description: The Inventory clean up job allows to delete history information and keeps only configurable amount

Last Result:

NUMBER-OF-FILES

Description: Total number of inventory files to keep as a history.

Number of Files: 5

Synchronize_PROXY_Slaves Job

Automatically synchronize binary packages, policies,assignment groups,host groups, and actions on PROXY *boom* servers if PROXY slaves are connected to this master server. Per default this job is scheduled every 15 min.

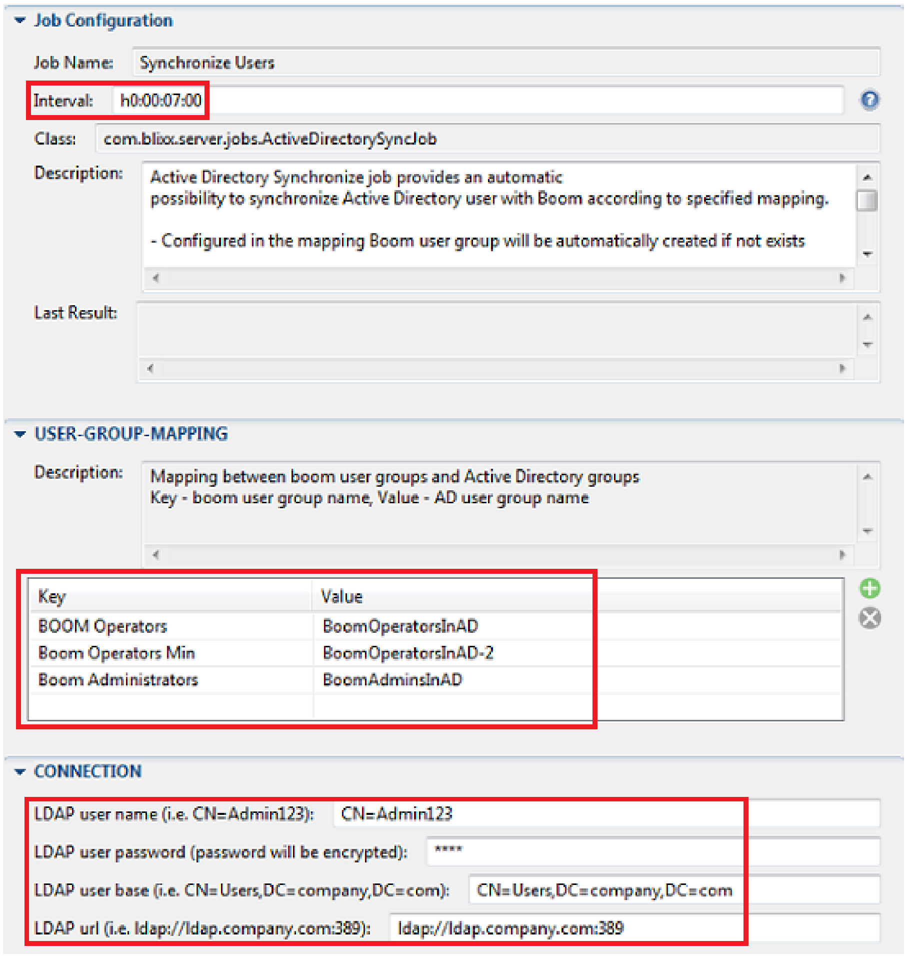
Every modification of policies, assignment groups, host groups and actions is directly synchronized to the slaves.

Each modification of packages via the user interface like file change, create file, delete file, create folder etc. is forwarded to the slaves immediately. Package modifications which were made on the file system (`<boom_dir>/srv/packages`) itself are synchronized either on demand (Push Package to all Proxy Server) or with the help of the Synchronize_PROXY_SLAVES job. The schedule time for the job is configurable.

Synchronize Users Job

The server job "Synchronize Users" is responsible for regular user synchronization of Active Directory Users with boom users. The Active Directory data is the "master" data, so the job will ensure that the list of boom users is same matching the Active Directory information.

- This job will not change any local boom users that might be present in the target boom user role unless there exists a local user and Active Directory user with the same name.
- If the specified boom user group does not exist, the job will automatically create it.
- A user in boom can only be member of one user role. Therefore if an Active Directory user is member of multiple user groups that are configured to be synchronized only the first found occurrence will be taken onto account.
- Active Directory users will be automatically created or removed from boom or their boom user role membership changed according to the information retrieved from the Active Directory information.
- Do not use the boom Tools for LDAP (i.e. LDAP Import/Reload) if you want to use the Synchronization Job. If there are changes that need to be synchronized at once, use the "Run once" Feature of the job as described later.



Job Configuration

Job Name: Synchronize Users

Interval: h0:00:07:00

Class: com.blixx.server.jobs.ActiveDirectorySyncJob

Description: Active Directory Synchronize job provides an automatic possibility to synchronize Active Directory user with Boom according to specified mapping.

- Configured in the mapping Boom user group will be automatically created if not exists

Last Result:

USER-GROUP-MAPPING

Description: Mapping between boom user groups and Active Directory groups
Key - boom user group name, Value - AD user group name

Key	Value
BOOM Operators	BoomOperatorsInAD
Boom Operators Min	BoomOperatorsInAD-2
Boom Administrators	BoomAdminsInAD

CONNECTION

LDAP user name (i.e. CN=Admin123): CN=Admin123

LDAP user password (password will be encrypted): ****

LDAP user base (i.e. CN=Users,DC=company,DC=com): CN=Users,DC=company,DC=com

LDAP url (i.e. ldap://ldap.company.com:389): ldap://ldap.company.com:389

In the **USER-GROUP-MAPPING** configuration section of the job the mapping between Active Directory user groups and boom user roles is configured. Only Active Directory user groups configured in this section are synchronized to boom. The synchronization of multiple user groups is possible. You have to define the mapping between BOOM User Role name and Active Directory User Group name.

```
Key   -> Name of the BOOM User Role
Value -> Name of the Active Directory User Group
```

In the **CONNECTION** section you can configure all connection properties for the Active Directory Service.

```
LDAP user must have according *rights* to list user groups and users.
```

```
URL -> for secure connection use: *ldaps://hostname:port*
```

Agent Export Job

Export agents and external hosts to a csv formatted file. The job can be configured to export to include only Agents or external hosts which are online or offline.
A logfile is available on:
`<server_install_dir>/<configured` working directory>/agentexport_<date>_<number>.log

User Export Job

Export user to a csv formatted file. A logfile is available on:
`<server_install_dir>/<configured working directory>/userexport_<date>_<number>.log`

Auto Close Filtered indications Job

Closes all active indications which match with one of the defined patterns. The key of the filter map is a template that supports the following variables:

```
<$AGENT_HOST>
<$AGENT_ID>
<$BOOM_SERVER> - if indication from slave server
<$APPLICATION>
<$GROUP>
<$OBJECT>
<$HOST>
<$TEXT>
<$SEVERITY>
<$STIME_STR> - Server time of indication/last Duplicate (yyyy-MM-dd HH:mm:ss)
<$TIME_STR> - Agent time of last duplicate indication (yyyy-MM-dd HH:mm:ss)
<$FTIME_STR> - Agent time of first indication  (yyyy-MM-dd HH:mm:ss)
<$KEY> - indication key
<$SOURCE>
<$CA1> - <$CA15> custom attributes in format name=value
<$name> - custom attribute by defined name
```

Auto Archive Filtered Indications Job

Archives all closed indications which match with one of the defined patterns. The key of the filter map is a template that supports the following variables:

```
<$AGENT_HOST>
<$AGENT_ID>
<$BOOM_SERVER> - if indication from slave server
<$APPLICATION>
<$GROUP>
<$OBJECT>
<$HOST>
<$TEXT>
<$SEVERITY>
<$STIME_STR> - Server time of indication/last Duplicate (yyyy-MM-dd HH:mm:ss)
<$TIME_STR> - Agent time of last duplicate indication (yyyy-MM-dd HH:mm:ss)
<$FTIME_STR> - Agent time of first indication  (yyyy-MM-dd HH:mm:ss)
<$KEY> - indication key
<$SOURCE>
<$CA1> - <$CA15> custom attributes in format name=value
<$name> - custom attribute by defined name
```

Import Virtual Agents

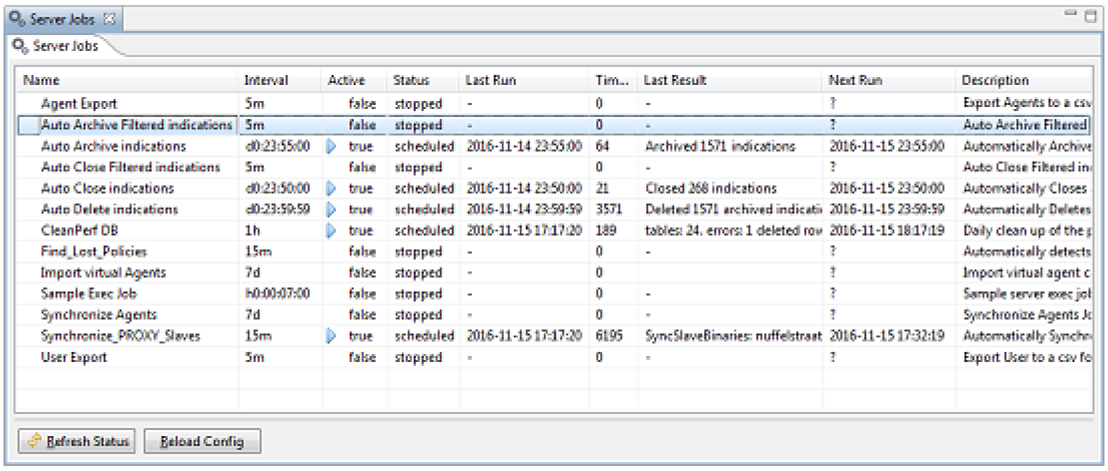
Allows to import external hosts from a comma separated file.
The file needs to be in the following format:
`AGENT_ID;LABEL;HOSTNAME;IP;DESCRIPTION`
i.e.:

```
;;NewHostName;1.1.1.1;
a9f14a8a-770c-483c-9a80-a719fa096feb;MyLabel;UpdatedHostName;1.1.1.1;"Updated Description"
```

If external host configurations should be updated it is necessary to specify the AGENT_ID field.

The *boom* **Server Jobs View** gives useful information about every job:

Click on any column to sort the elements of the selected column.



Name	Interval	Active	Status	Last Run	Tim...	Last Result	Next Run	Description
Agent Export	5m	false	stopped	-	0	-	?	Export Agents to a csv
Auto Archive Filtered indications	5m	false	stopped	-	0	-	?	Auto Archive Filtered
Auto Archive indications	d0:23:55:00	true	scheduled	2016-11-14 23:55:00	64	Archived 1571 indications	2016-11-15 23:55:00	Automatically Archive
Auto Close Filtered indications	5m	false	stopped	-	0	-	?	Auto Close Filtered in
Auto Close indications	d0:23:50:00	true	scheduled	2016-11-14 23:50:00	21	Closed 268 indications	2016-11-15 23:50:00	Automatically Closes
Auto Delete indications	d0:23:59:59	true	scheduled	2016-11-14 23:59:59	3571	Deleted 1571 archived indicati	2016-11-15 23:59:59	Automatically Deletes
CleanPerf DB	1h	true	scheduled	2016-11-15 17:17:20	189	tables: 24. errors: 1 deleted row	2016-11-15 18:17:19	Daily clean up of the p
Find_Lost_Policies	15m	false	stopped	-	0	-	?	Automatically detects
Import virtual Agents	7d	false	stopped	-	0		?	Import virtual agent c
Sample Exec Job	h0:00:07:00	false	stopped	-	0	-	?	Sample server exec jol
Synchronize Agents	7d	false	stopped	-	0	-	?	Synchronize Agents Jc
Synchronize_PROXY_Slaves	15m	true	scheduled	2016-11-15 17:17:20	6195	SyncSlaveBinaries: nuffelstraat	2016-11-15 17:32:19	Automatically Synchr
User Export	5m	false	stopped	-	0	-	?	Export User to a csv fo

Name: Name of the server job.

Interval: Job schedule interval.

Active: Job status if job is activated or deactivated.

Status: Shows the status of the job e.g. scheduled.

Last Run: Last schedule time.

Next Run: Next schedule time.

Time spent: Last job running time in ms.

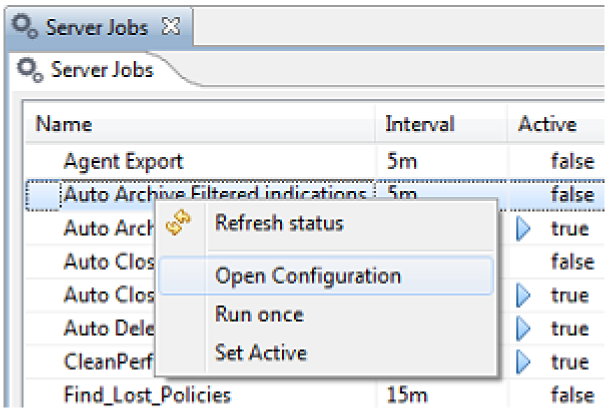
Last Result: Result of the last job schedule.

Description: Description of the job.

6.15.2. Server Jobs Operations and Configuration

Open the **Show Server Jobs View** and select a server job.

Right-click on a job element to open the context menu:



Name	Interval	Active
Agent Export	5m	false
Auto Archive Filtered indications	5m	false
Auto Arch		true
Auto Clos		false
Auto Clos		true
Auto Dele		true
CleanPerf		true
Find_Lost_Policies	15m	false

Refresh Status: Refreshes the status.

Open Configuration: Opens the configuration view for the selected server job.

Run Once:	Allows you to start the selected job manually.
Set Inactive:	Allows you to set an active job to inactive. The active flag in the "Show Server Jobs View" changes to false.
Set Active:	Allows you to set an inactive job to active. The active flag in the "Show Server Jobs View" changes to true.

Select "Open Configuration" to open the configuration view:

Job Name:	Name of the server job.
Interval:	Shows the job schedule interval. Examples of valid time intervals: 5m - 5 minutes 5h - 5 hours 5d - 5 days h0:00:11:12 - every hour at 11 min 12 sec d0:15:11:12 - daily at 15 hours 11 min 12 sec w4:15:11:12 - weekly every friday at 15 hours 11 min 12 sec m15:00:05:00 - every 15 minutes shifted from midnight 00:00:00 by offset of 5 min (offset must be less than interval)
Description:	Detailed description of the server job.
Class:	Used Java class.
Last Result:	Summary of the last run of the job.
Max Hours:	Only valid for the jobs "Auto Archive/Close/Delete indications". Indications will be archived/closed/deleted after the specified "Max Days".
Export working directory:	Only valid for the job "CleanPerf DB", "Agent Export" and "User Export". This directory is used for all outputs. The directory is also used for writing log files.
Keep maximum history of (days):	Only valid for the job "CleanPerf DB". All data older than "Keep maximum history of" are deleted from database.

6.16. Import/Export Management Plug-Ins

The *boom* agent provides the infrastructure to monitor and manage systems. The agent’s main responsibilities are the scheduling, thresholding, correlation, suppression and other necessary processing steps as well as the communication with the *boom* server. The actual monitoring is configured and performed by so called Management PlugIns (MPI). Such MPIs contain the knowledge what needs to be monitored and how to monitor it. Generally an MPI is a combination of `binaries/scripts/executables`, policies, remote actions and assignment configurations.
The import/export functionality is used to import additional Plug-Ins as well as rolling out monitoring configurations e.g. from *boom* development to production systems.

6.16.1. Import MPI

There are several Import MPI functions available:

Import MPI:

This is an earlier implementation and has been replaced by the new "Show Import MPI View".
Open the file menu and select "Import MPI".
Select MPI's directory (location of the MPI files) and click "OK" to import the MPI's. You can import policies, packages, assignment groups, actions and node groups according to your individual requirements.

Show Import MPI View:

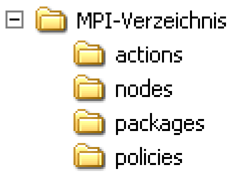
Policies/policy groups, assignment groups, binaries, actions and node groups can be imported according to your individual requirements.

1. Open the File Menu and select **Show Import MPI View.**

   There are two **Import Types** available:

   Import MPI

   The "Import MPI" type expects the following directory structure for loading (same structure as created by the MPI Export utility). If e.g. only actions have to be imported only the actions directory has to exist.

   Import Other

   "Import Other" allows to import other directory structures e.g. **HP Operation Manager configuration files**.

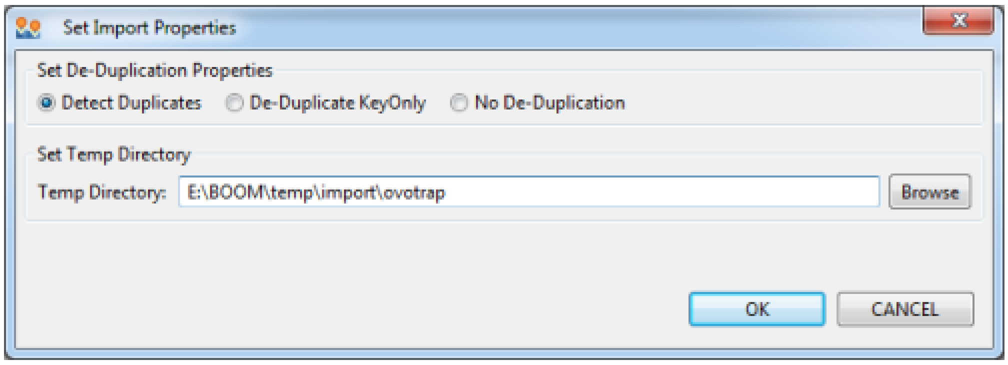
   The import of the following HP Operations Manager configuration data is supported:
   Host Groups
   Policy Groups
   Policies (Trap, Message, PerformanceMonitor, SchedulePolicy, LogfileMonitor)

Procedure:

- Download Operations Manager Policy. Hint: syntax version lower or equal 8 will be supported only!
- Transfer Data-File(s) of Download (<policy_id>_data) to a directory where UI can read it.
- Import directory mentioned above must be structured as following: <any_dir>\C\TEMPLATES\<policy_type>\<policy_id>_data; policy type must be in this range: TRAP|LOGFILE|SCHEDULE|MONITOR|INTERFACE
- **"Load Directory"** must point to <any_dir>
- Hint: imported policies will be opened but not saved automatically - save it when manual inspection is done.

Instruction Text Interface
TroubleTicket Flags as CMA

+ Additionally this import type asks you for the Import Properties. Select the type of De-Duplication you want to use for the selected policies.

+ 
Set Import Properties

Set De-Duplication Properties

(● Detect Duplicates) ○ De-Duplicate KeyOnly ○ No De-Duplication

Set Temp Directory

Temp Directory: E:\BOOM\temp\import\ovotrap

. Use **"Load Directory"** to select the directory which should be imported. . The configuration data will be loaded. The **"Preview Window"** shows all possible conflicts. All **green** marked elements have no conflict and can be imported. . Select **"Start Import"**

+ image::images/import_mpiview.png[width=500]

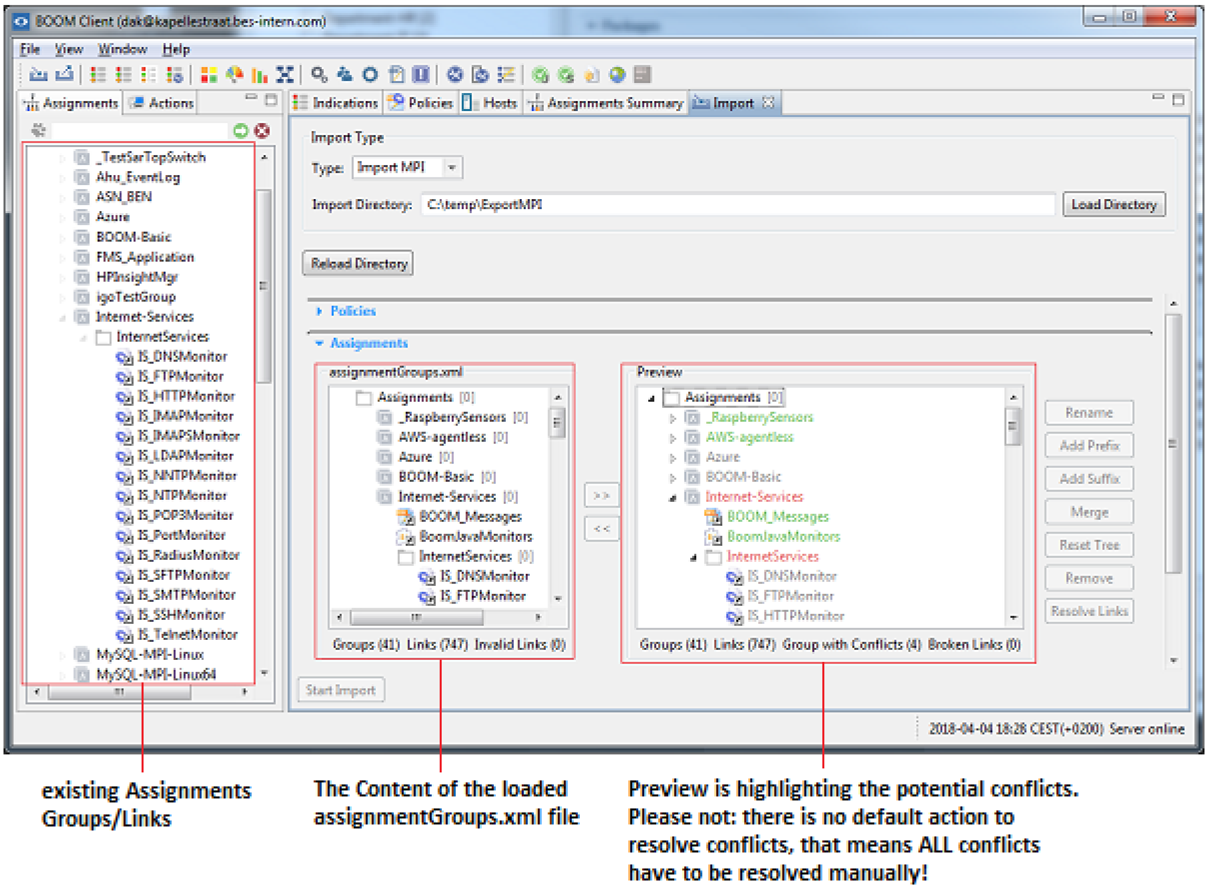
+ The configuration data can't be imported if there are any conflicts. In this case the conflicting elements are marked **red** in the **Preview window**. The user has to decide how to solve the conflict manually. In some cases there are more possibilities to solve the conflict e.g. policy conflict → rename policy or import policy with a higher version. If all conflicts are solved the import can be started. . Solving **import conflicts:**

+ **Solving policy import conflicts:**

+ image::images/import_mpi_policy.png[width=500]

+ Compare with server: :: When conflicts are reported use as first step the compare function. This will do an indepth comparison of the policy content to verify if there are changes between the incoming and existing policies. Policies that don't have changes will be grayed out as there is no need to load them again. Policies with real conflicts remain marked red and for them one of the further resolution actions need to be selected. Rename: :: Rename the name of the importing policy. Increase Version: :: The policy will be imported with a higher version. Use XML path: :: Policy already exists in another policy group on the server. Policy will be moved into the policy group specified in the MPI import. Use Server Path: :: Policy already exists in another policy group on the server. Policy won't be moved to another directory. Reset Tree: :: Reload the previous tree (review) Remove: :: Remove a single policy or policy tree from import

1. **Solving assignment group conflicts:**


BOOM Client (dak@kapellestraat.bes-intern.com)

Import Type

Type: Import MPI

Import Directory: C:\temp\ExportMPI

assignmentGroups.xml: Groups (41) Links (747) Invalid Links (0)

Preview: Groups (41) Links (747) Group with Conflicts (4) Broken Links (0)

2018-04-04 18:28 CEST(+0200) Server online

File
View
Window
Help
Assignments
Actions
Indications
Policies
Hosts
Assignments Summary
Import
Load Directory
Reload Directory
Policies
Assignments
Rename
Add Prefix
Add Suffix
Merge
Reset Tree
Remove
Resolve Links
Start Import
existing Assignments Groups/Links
The Content of the loaded assignmentGroups.xml file
Preview is highlighting the potential conflicts. Please not: there is no default action to resolve conflicts, that means ALL conflicts have to be resolved manually!

Rename:

Rename the assignment group

Add Prefix:

Add a prefix to the assignment group

Add Suffix:

Add a suffix to the assignment group

Merge:

Merges the element with the existing element

Reset Tree:

Reload the previous tree

Remove:

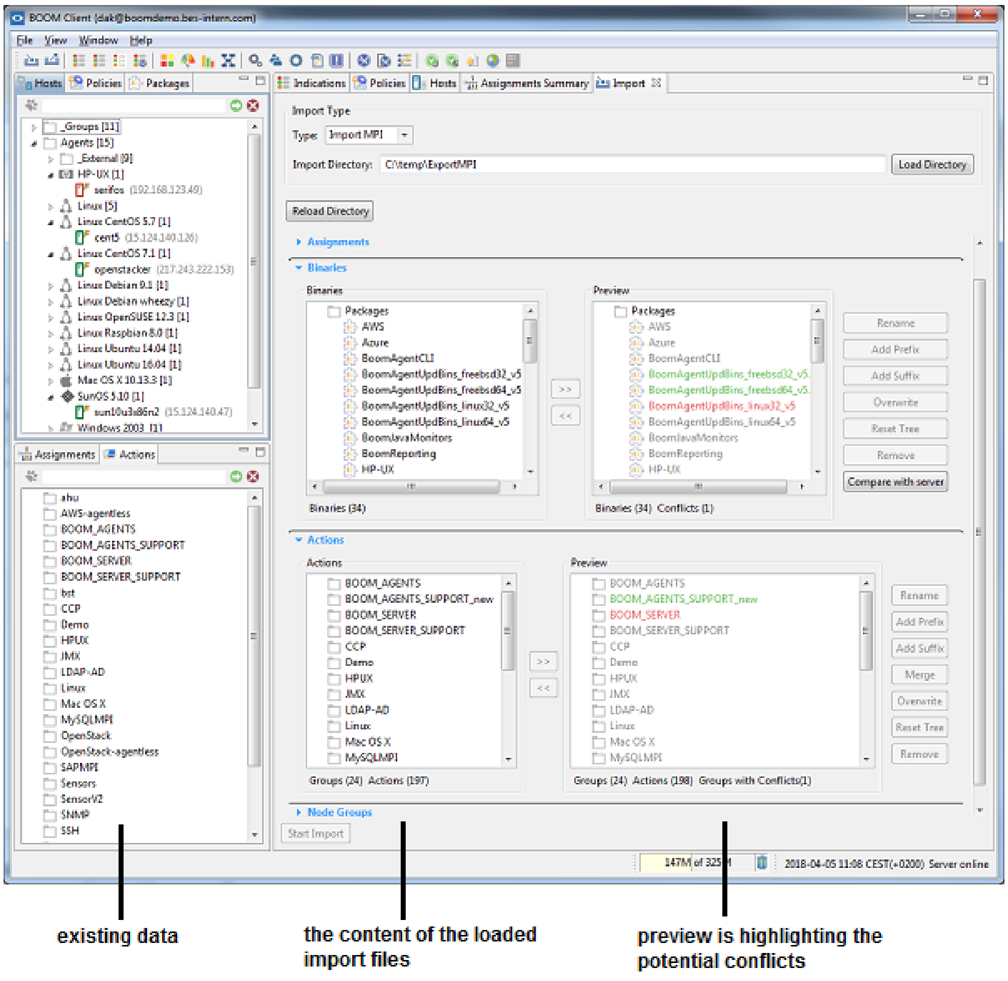
Remove element from import

Resolve Links:

All links that point to an element (Policy or Package) that is not yet selected for import, are marked with the following icon:

The 'Resolve Link' method will try to locate and add all missing elements in the import package. Imports with broken links can be performed, since the element might be already on the server or loaded later with a different package.

2. **Solving Packages or Action Conflicts**


BOOM Client
File View Window Help
Hosts Policies Packages
Indications Policies Hosts Assignments Summary Import
Import Type
Type: Import MPI
Import Directory: C:\temp\ExportMPI
Load Directory
Reload Directory
Assignments
Binaries
Preview
Rename
Add Prefix
Add Suffix
Overwrite
Reset Tree
Remove
Compare with server
Binaries (34)
Binaries (34) Conflicts (1)
Actions
Merge
Groups (24) Actions (197)
Groups (24) Actions (198) Groups with Conflicts(1)
Node Groups
Start Import
Assignments Actions
2018-04-05 11:08 CEST(+0200) Server online
existing data
the content of the loaded import files
preview is highlighting the potential conflicts

The conflict solving follows the same principles that are used before for policies and assignment groups.
For the binaries as first step perform the in-depth comparison with the 'Compare with server' functionality and decide for the remaining conflicts to rename, overwrite or skip the conflicting objects. For action groups you can also decide to 'Merge' the contents to get as result a mixture of old and new actions inside the action group while 'Overwrite' would remove the old content.
For Nodegroups it is only possible to 'Remove' conflicting groups.

The following signs are used to show the kind of conflict and conflict solution:

Common signs:

green element has no conflict
grey element already exists on the server and won't be imported

Policy conflict signs:

🗀 Policy alreday exists in the same directory on the server
🗀 Path conflict. Policy already exists in another policy group on the server

Conflict solution signs:

Ⓥ Policy will be imported with a higher version Ⓜ Policy will be moved to a new policy group
Ⓡ Element will be renamed Element will be merged with the existing element. This is possible for assignments and actions.

Informational signs:

BOOM_Ping BOOM_Messages

The \policy\assignmentGroups.xml file of the MPI package contains links pointing to non existing physical xml files (policy xml, package xml). The missing xml files exist already on the server hence the links can be imported.

BOOM_Ping BOOM_Messages

The `\policy\assignmentGroups.xml` file of the MPI package contains links pointing to non existing physical xml files (policy xml, package xml). The missing xml files do not exist on the server hence the links cannot be imported.

Assignment link has been selected for import but the corresponding policy or package has not been selected.
The import does not contain an element which consists on the server.

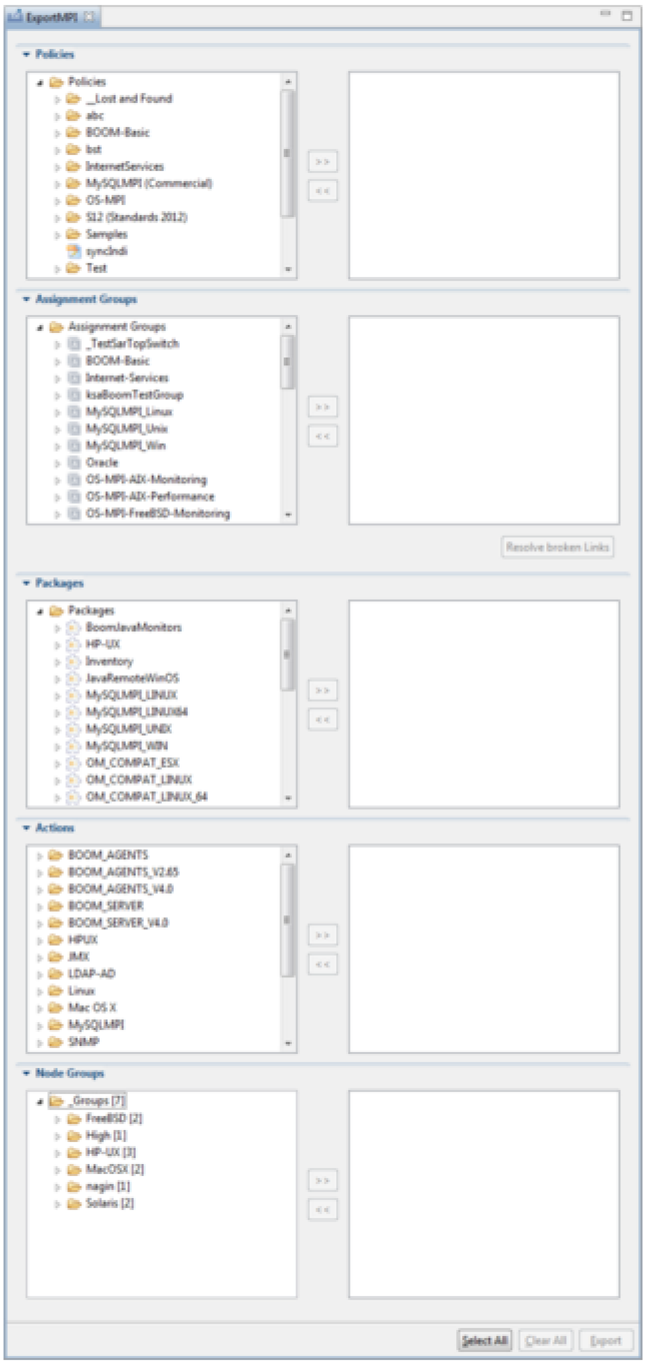
3. **Checking of the import log file**

   Open the import logfile `<boom_gui>/<user>_<server>/BoomPolicyConverter_date.log`
   Check for hints or warnings to verify the conversion results

6.16.2. Export MPI

Show Export MPI View: Policies/policy groups, assignment groups, binaries, actions and node groups can be exported according to your individual requirements.

1. Open the file menu and select **"Show Export MPI View"**.
2. The "Export MPI View" opens. Expand/collapse the appropriate section and **mark the policies, packages, assignment groups, actions and node groups** you want to export.
3. Press the **">>"** button to put the selected items into the right window for export.
4. Press the **"Export Button"** and **select** the destination directory for saving. The tree structure and all items are exported to xml formatted files.

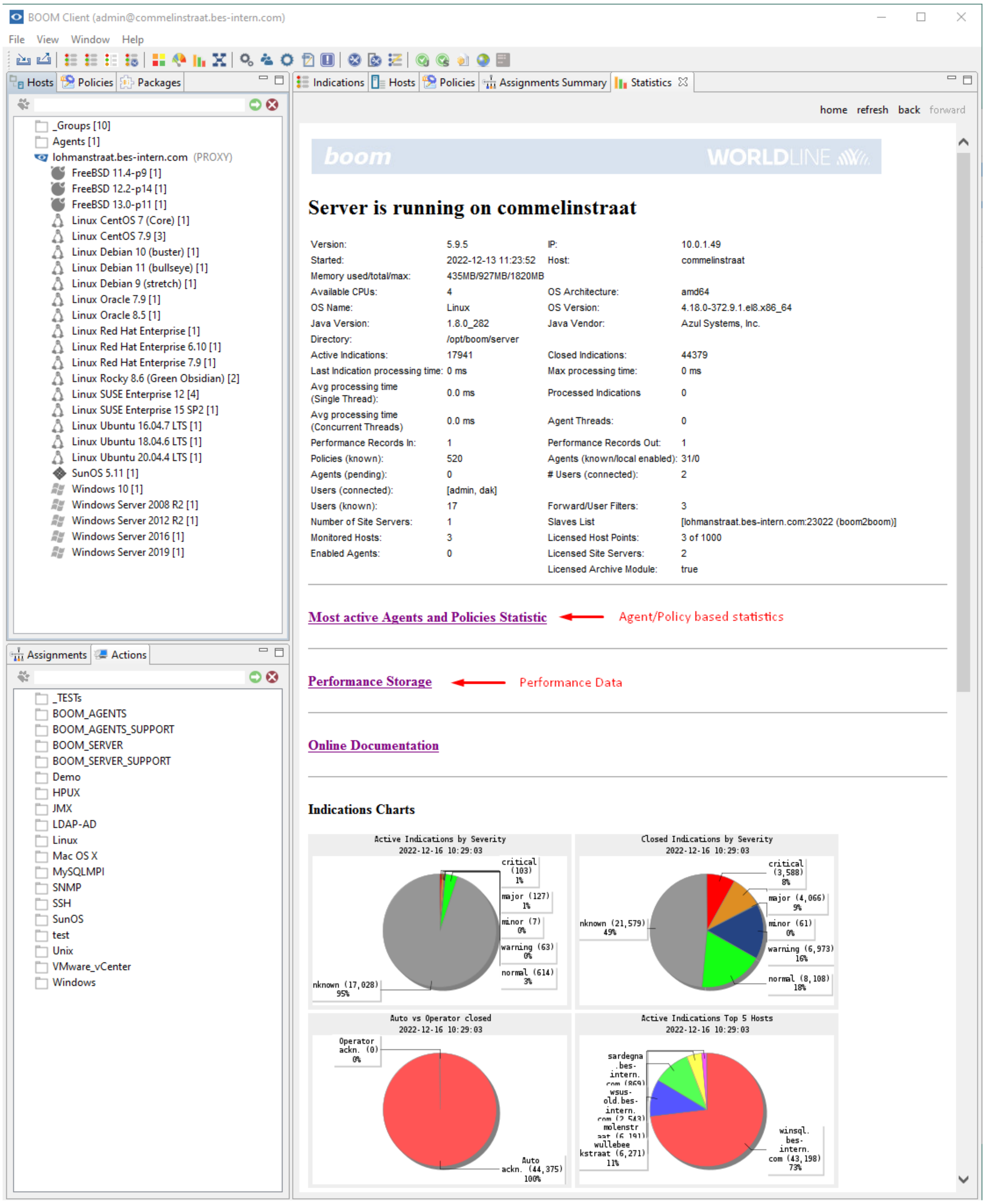


6.17. Statistics

The "Statistics View" contains some general information about the *boom* server, it contains some performance data and various "Indication Charts".
See also chapter Performance Data for more information about statistics.

6.17.1. General Server Information

Open the **Statistics** view in the *boom* User Interface to get general information about the boomserver.



_Groups [10]
Agents [1]
lohmanstraat.bes-intern.com (PROXY)
FreeBSD 11.4-p9 [1]
FreeBSD 12.2-p14 [1]
FreeBSD 13.0-p11 [1]
Linux CentOS 7 (Core) [1]
Linux CentOS 7.9 [3]
Linux Debian 10 (buster) [1]
Linux Debian 11 (bullseye) [1]
Linux Debian 9 (stretch) [1]
Linux Oracle 7.9 [1]
Linux Oracle 8.5 [1]
Linux Red Hat Enterprise [1]
Linux Red Hat Enterprise 6.10 [1]
Linux Red Hat Enterprise 7.9 [1]
Linux Rocky 8.6 (Green Obsidian) [2]
Linux SUSE Enterprise 12 [4]
Linux SUSE Enterprise 15 SP2 [1]
Linux Ubuntu 16.04.7 LTS [1]
Linux Ubuntu 18.04.6 LTS [1]
Linux Ubuntu 20.04.4 LTS [1]
SunOS 5.11 [1]
Windows 10 [1]
Windows Server 2008 R2 [1]
Windows Server 2012 R2 [1]
Windows Server 2016 [1]
Windows Server 2019 [1]

_TESTs
BOOM_AGENTS
BOOM_AGENTS_SUPPORT
BOOM_SERVER
BOOM_SERVER_SUPPORT
Demo
HPUX
JMX
LDAP-AD
Linux
Mac OS X
MySQLMPI
SNMP
SSH
SunOS
test
Unix
VMware_vCenter
Windows

Server is running on commelinstraat

Version:	5.9.5	IP:	10.0.1.49
Started:	2022-12-13 11:23:52	Host:	commelinstraat
Memory used/total/max:	435MB/927MB/1820MB		
Available CPUs:	4	OS Architecture:	amd64
OS Name:	Linux	OS Version:	4.18.0-372.9.1.el8.x86_64
Java Version:	1.8.0_282	Java Vendor:	Azul Systems, Inc.
Directory:	/opt/boom/server		
Active Indications:	17941	Closed Indications:	44379
Last Indication processing time:	0 ms	Max processing time:	0 ms
Avg processing time (Single Thread):	0.0 ms	Processed Indications	0
Avg processing time (Concurrent Threads)	0.0 ms	Agent Threads:	0
Performance Records In:	1	Performance Records Out:	1
Policies (known):	520	Agents (known/local enabled):	31/0
Agents (pending):	0	# Users (connected):	2
Users (connected):	[admin, dak]		
Users (known):	17	Forward/User Filters:	3
Number of Site Servers:	1	Slaves List	[lohmanstraat.bes-intern.com:23022 (boom2boom)]
Monitored Hosts:	3	Licensed Host Points:	3 of 1000
Enabled Agents:	0	Licensed Site Servers:	2
		Licensed Archive Module:	true

Most active Agents and Policies Statistic ← Agent/Policy based statistics

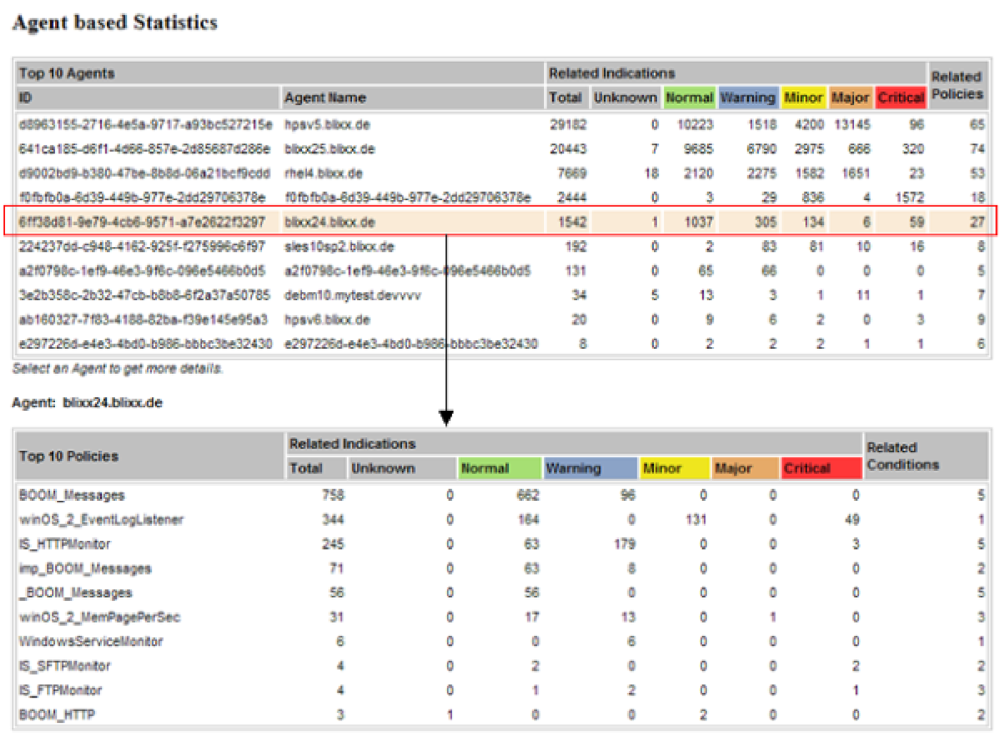
Performance Storage ← Performance Data

Online Documentation

Indications Charts

6.17.2. Agent based Statistics

Open the **Statistics** view in the *boom* User Interface.
Select **Most active Agent and Policies Statistics** to get the Agent based Statistics.


Agent based Statistics

Top 10 Agents		Related Indications							Related Policies
ID	Agent Name	Total	Unknown	Normal	Warning	Minor	Major	Critical	
d8963155-2716-4e5a-9717-a93bc527215e	hpsv5.blixx.de	29182	0	10223	1518	4200	13145	96	65
641ca185-d6f1-4d66-857e-2d85687d286e	blixx25.blixx.de	20443	7	9685	6790	2975	666	320	74
d9002bd9-b380-47be-8b8d-06a21bcf9cdd	rhel4.blixx.de	7669	18	2120	2275	1582	1651	23	53
f0fbfb0a-6d39-449b-977e-2dd29706378e	f0fbfb0a-6d39-449b-977e-2dd29706378e	2444	0	3	29	836	4	1572	18
6ff38d81-9e79-4cb6-9571-a7e2622f3297	blixx24.blixx.de	1542	1	1037	305	134	6	59	27
224237dd-c948-4162-925f-f275996c6f97	sles10sp2.blixx.de	192	0	2	83	81	10	16	8
a2f0798c-1ef9-46e3-9f6c-096e5466b0d5	a2f0798c-1ef9-46e3-9f6c-096e5466b0d5	131	0	65	66	0	0	0	5
3e2b358c-2b32-47cb-b8b8-6f2a37a50785	debm10.mytest.devvvv	34	5	13	3	1	11	1	7
ab160327-7f83-4188-82ba-f39e145e95a3	hpsv6.blixx.de	20	0	9	6	2	0	3	9
e297226d-e4e3-4bd0-b986-bbbc3be32430	e297226d-e4e3-4bd0-b986-bbbc3be32430	8	0	2	2	2	1	1	6

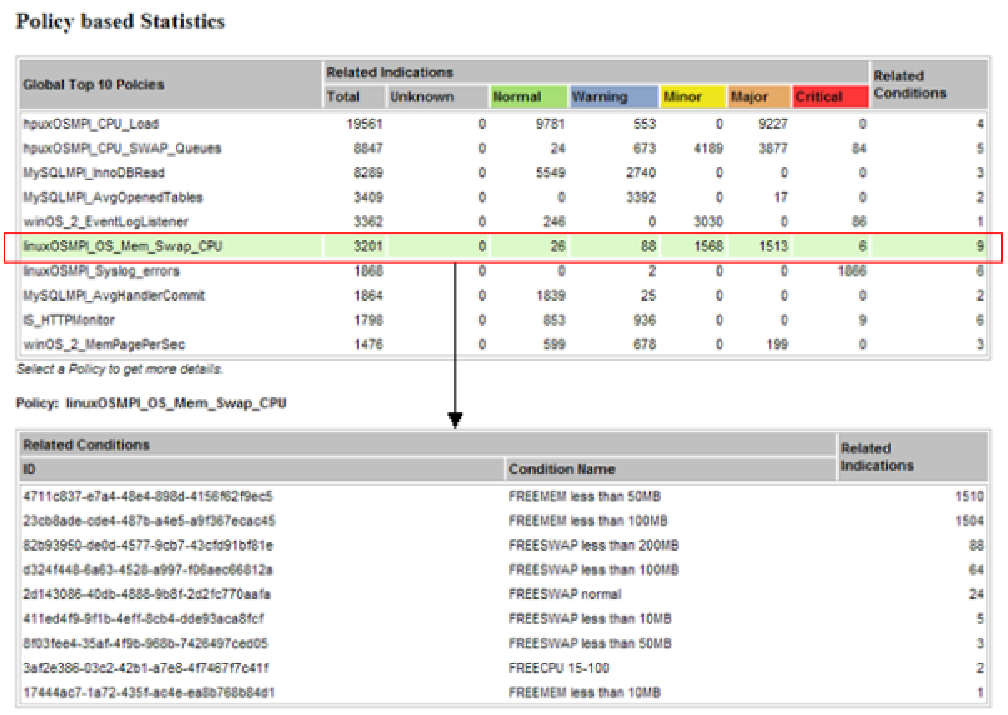
Select an Agent to get more details.

Agent: blixx24.blixx.de

Top 10 Policies	Related Indications							Related Conditions
	Total	Unknown	Normal	Warning	Minor	Major	Critical	
BOOM_Messages	758	0	662	96	0	0	0	5
winOS_2_EventLogListener	344	0	164	0	131	0	49	1
IS_HTTPMonitor	245	0	63	179	0	0	3	5
imp_BOOM_Messages	71	0	63	8	0	0	0	2
_BOOM_Messages	56	0	56	0	0	0	0	5
winOS_2_MemPagePerSec	31	0	17	13	0	1	0	3
WindowsServiceMonitor	6	0	0	6	0	0	0	1
IS_SFTPMonitor	4	0	2	0	0	0	2	2
IS_FTPMonitor	4	0	1	2	0	0	1	3
BOOM_HTTP	3	1	0	0	2	0	0	2

6.17.3. Policy based Statistics

Open the **Statistics** view in the *boom* User Interface.
Select **Most active Agent and Policies Statistics** to get the Policy based Statistics.


Policy based Statistics

Global Top 10 Policies	Related Indications							Related Conditions
	Total	Unknown	Normal	Warning	Minor	Major	Critical	
hpuxOSMPI_CPU_Load	19561	0	9781	553	0	9227	0	4
hpuxOSMPI_CPU_SWAP_Queues	8847	0	24	673	4189	3877	84	5
MySQLMPI_InnoDBRead	8289	0	5549	2740	0	0	0	3
MySQLMPI_AvgOpenedTables	3409	0	0	3392	0	17	0	2
winOS_2_EventLogListener	3362	0	246	0	3030	0	86	1
linuxOSMPI_OS_Mem_Swap_CPU	3201	0	26	88	1568	1513	6	9
linuxOSMPI_Syslog_errors	1868	0	0	2	0	0	1866	6
MySQLMPI_AvgHandlerCommit	1864	0	1839	25	0	0	0	2
IS_HTTPMonitor	1798	0	853	936	0	0	9	6
winOS_2_MemPagePerSec	1476	0	599	678	0	199	0	3

Select a Policy to get more details.

Policy: linuxOSMPI_OS_Mem_Swap_CPU

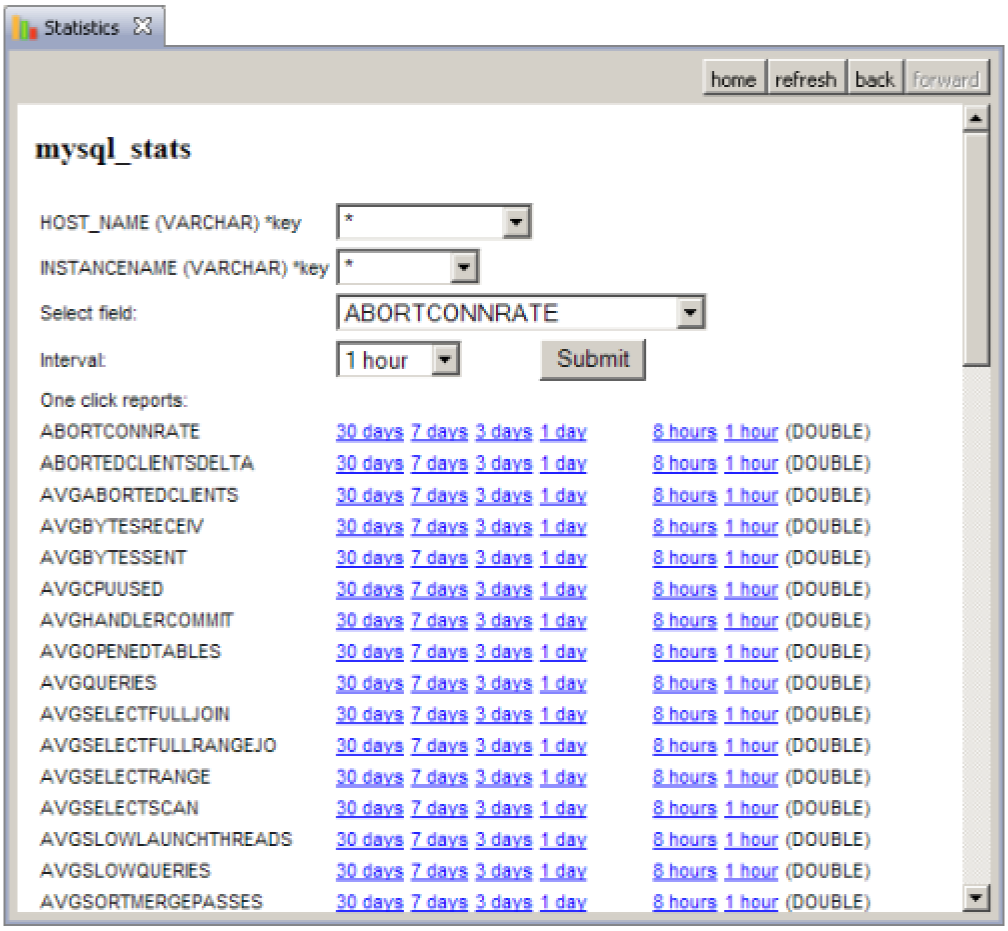
Related Conditions		Related Indications
ID	Condition Name	
4711c837-e7a4-48e4-898d-4156f62f9ec5	FREEMEM less than 50MB	1510
23cb8ade-cde4-487b-a4e5-a9f367ecac45	FREEMEM less than 100MB	1504
82b93950-de0d-4577-9cb7-43cfd91bf81e	FREESWAP less than 200MB	88
d324f448-6a63-4528-a997-f06aec66812a	FREESWAP less than 100MB	64
2d143086-40db-4888-9b8f-2d2fc770aafa	FREESWAP normal	24
411ed4f9-9f1b-4eff-8cb4-dde93aca8fcf	FREESWAP less than 10MB	5
8f03fee4-35af-4f9b-968b-7426497ced05	FREESWAP less than 50MB	3
3af2e386-03c2-42b1-a7e8-4f7467f7c41f	FREECPU 15-100	2
17444ac7-1a72-435f-ac4e-ea8b768b84d1	FREEMEM less than 10MB	1

6.17.4. Performance Data

Only MPIs with enabled performance data collection deliver performance data to the *boom* server. Appropriate database tables will be created after registering a performance class and submitting first record. After that the *boom* server will be able to generate content of this page.

If there are no MPIs with enabled performance collection installed on the managed *boom* agents, this page remains empty.

Open the **Statistics** view in the *boom* User Interface.
Select **Performance Storage** to get the Performance Data.



mysql_stats

HOST_NAME (VARCHAR) *key	*
INSTANCENAME (VARCHAR) *key	*
Select field:	ABORTCONNRATE
Interval:	1 hour Submit

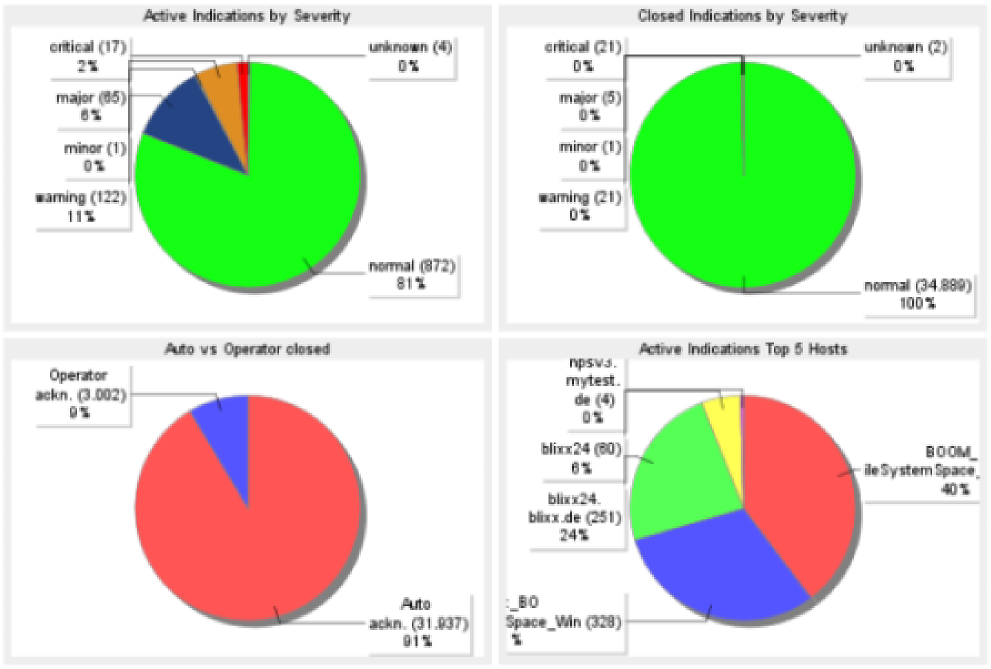
One click reports:

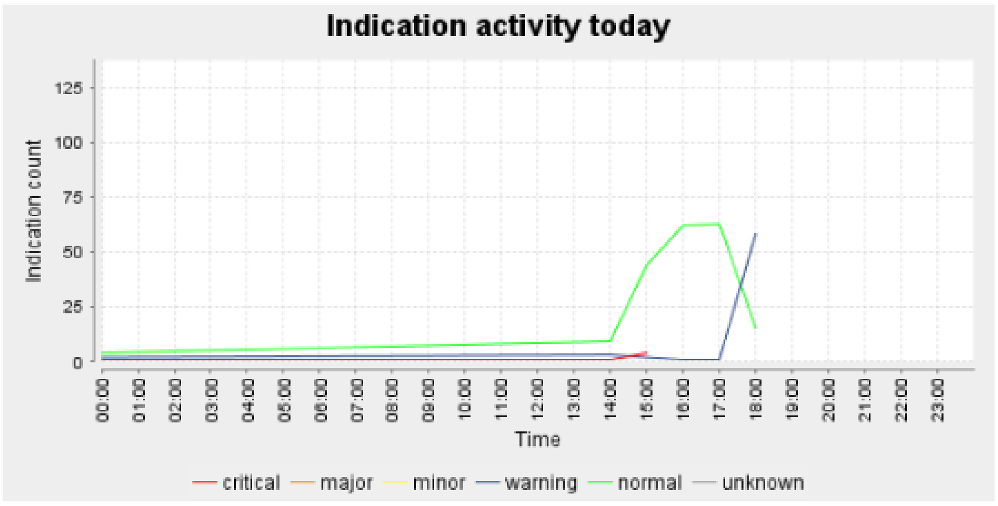
ABORTCONNRATE	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
ABORTEDCLIENTSDELTA	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGABORTEDCLIENTS	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGBYTESRECEIV	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGBYTESSENT	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGCPUUSED	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGHANDLERCOMMIT	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGOPENEDTABLES	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGQUERIES	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGSELECTFULLJOIN	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGSELECTFULLRANGEJO	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGSELECTRANGE	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGSELECTSCAN	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGSLOWLAUNCHTHREADS	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGSLOWQUERIES	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)
AVGSORTMERGEPASSES	30 days 7 days 3 days 1 day	8 hours 1 hour (DOUBLE)

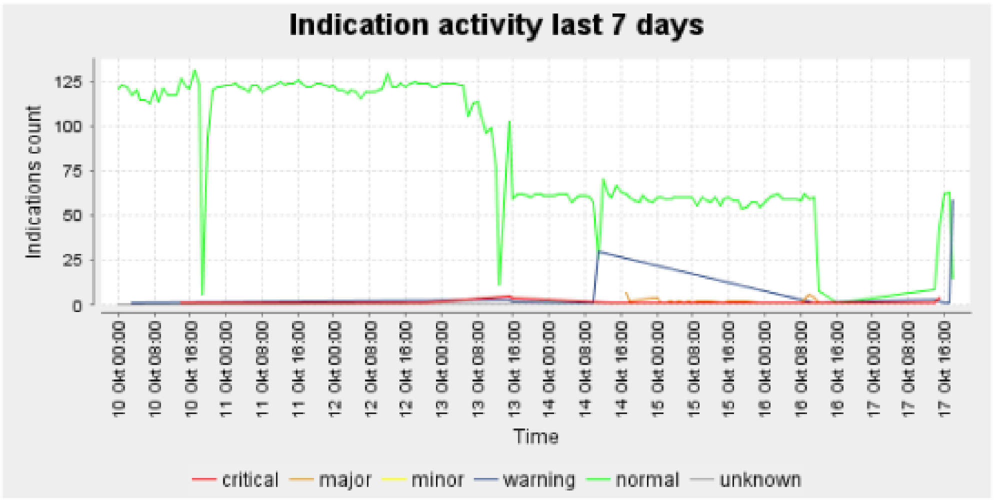
6.17.5. Indication Charts

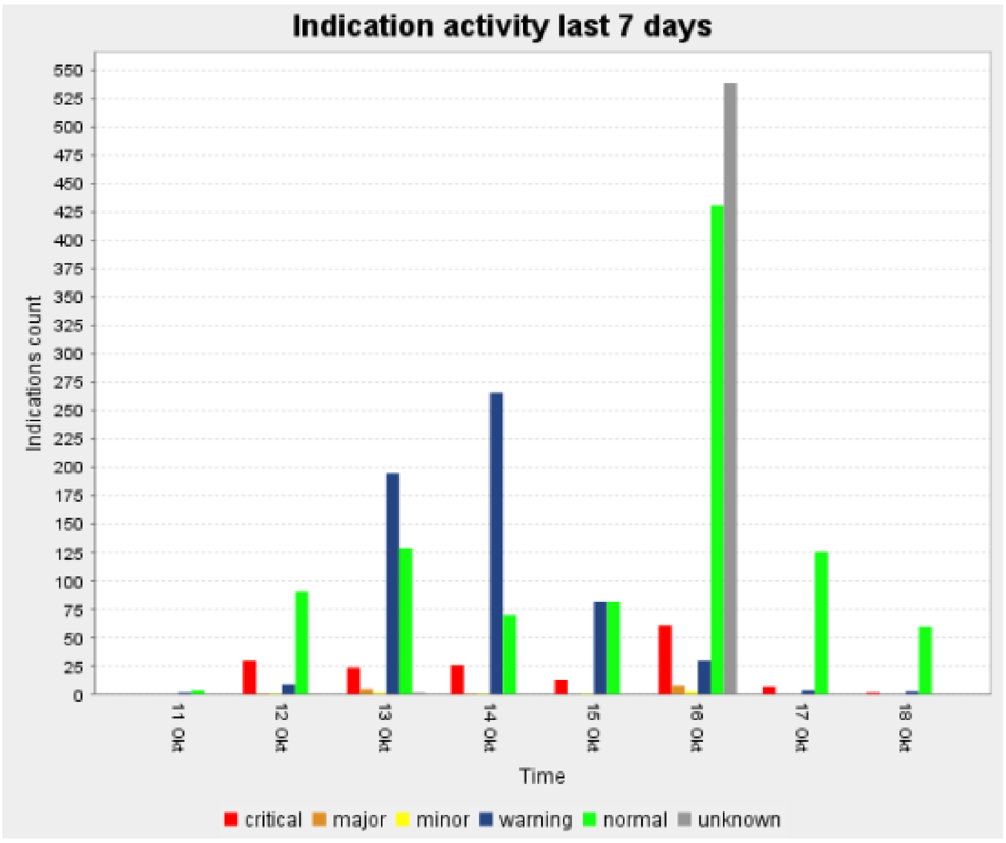
All charts below are reflecting current state (snapshot) of active indications on the *boom* server.

Open the **Statistics** view in the *boom* User Interface.
Scroll down to the indication charts.




Indication activity today
Indication count
125
100
75
50
25
0
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Time
critical
major
minor
warning
normal
unknown


Indication activity last 7 days
Indications count
125
100
75
50
25
0
10 Okt 00:00
10 Okt 08:00
10 Okt 16:00
11 Okt 00:00
11 Okt 08:00
11 Okt 16:00
12 Okt 00:00
12 Okt 08:00
12 Okt 16:00
13 Okt 00:00
13 Okt 08:00
13 Okt 16:00
14 Okt 00:00
14 Okt 08:00
14 Okt 16:00
15 Okt 00:00
15 Okt 08:00
15 Okt 16:00
16 Okt 00:00
16 Okt 08:00
16 Okt 16:00
17 Okt 00:00
17 Okt 08:00
17 Okt 16:00
Time
critical
major
minor
warning
normal
unknown

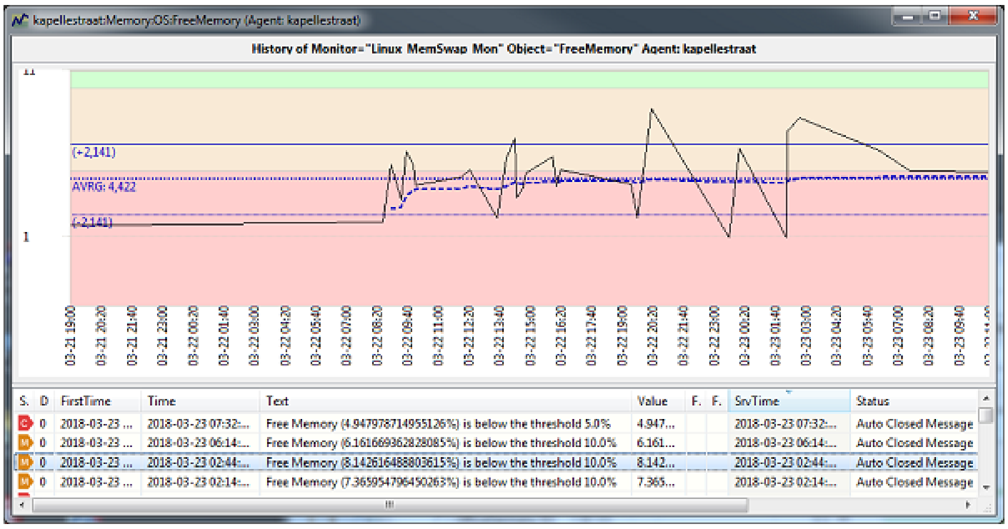

Indication activity last 7 days
Indications count
550
525
500
475
450
425
400
375
350
325
300
275
250
225
200
175
150
125
100
75
50
25
0
11 Okt
12 Okt
13 Okt
14 Okt
15 Okt
16 Okt
17 Okt
18 Okt
Time
critical
major
minor
warning
normal
unknown

6.18. History Chart

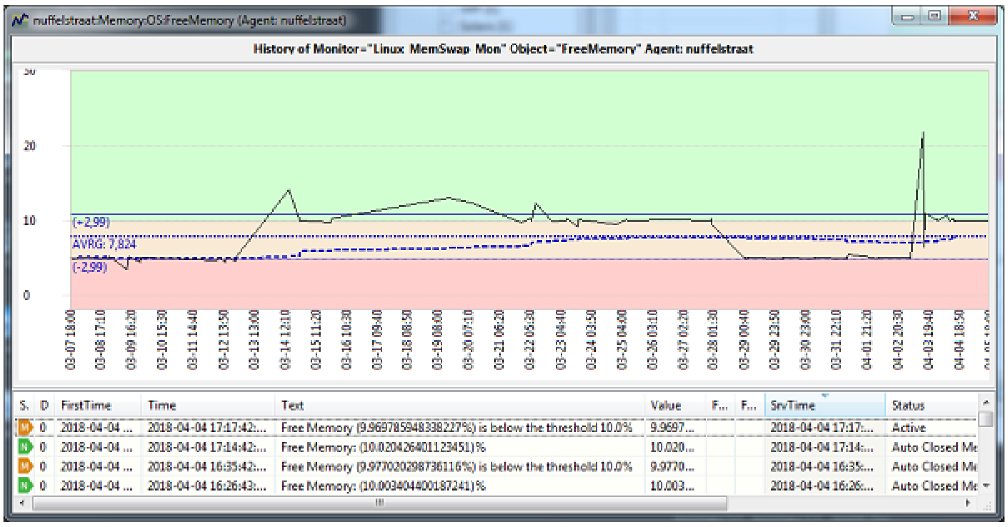
History Charts are available for any indication (right-click on an indication) that are submitted by a monitor policy. Such indications have a monitor value that can be processed by the chart engine which is integrated in the *boom* client. A generated chart displays the history of all values that are not archived and related to the selected indication. Related indications will be filtered based on the standard key:

```
AgentHost:MonitoredHost:Application:IndicationGroup:Object:Monitor
```

The background is colored according to the threshold levels and the severities defined in the policy.


kapellestraat:Memory:OS:FreeMemory (Agent: kapellestraat)
History of Monitor="Linux MemSwap Mon" Object="FreeMemory" Agent: kapellestraat
(+2,141)
AVRG: 4,422
(-2,141)
1
03-21 19:00
03-22 00:20
03-23 09:40

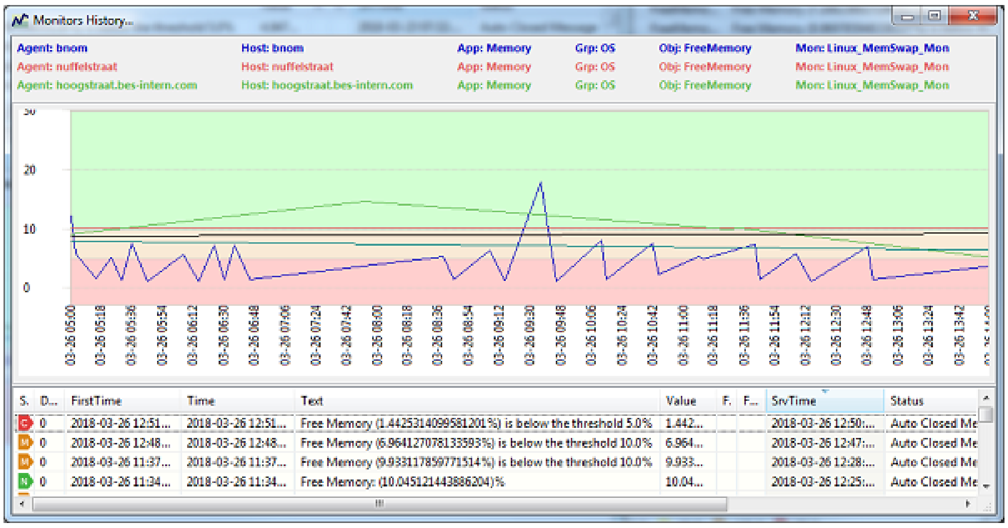
S.	D	FirstTime	Time	Text	Value	F.	F.	SrvTime	Status
C	0	2018-03-23 ...	2018-03-23 07:32:...	Free Memory (4.947978714955126%) is below the threshold 5.0%	4.947...			2018-03-23 07:32:...	Auto Closed Message
M	0	2018-03-23 ...	2018-03-23 06:14:...	Free Memory (6.161669362828085%) is below the threshold 10.0%	6.161...			2018-03-23 06:14:...	Auto Closed Message
M	0	2018-03-23 ...	2018-03-23 02:44:...	Free Memory (8.142616488803615%) is below the threshold 10.0%	8.142...			2018-03-23 02:44:...	Auto Closed Message
M	0	2018-03-23 ...	2018-03-23 02:14:...	Free Memory (7.365954796450263%) is below the threshold 10.0%	7.365...			2018-03-23 02:14:...	Auto Closed Message


nuffelstraat:Memory:OS:FreeMemory (Agent: nuffelstraat)
History of Monitor="Linux MemSwap Mon" Object="FreeMemory" Agent: nuffelstraat
20
10
0
(+2,99)
AVRG: 7,824
(-2,99)
03-07 18:00
04-04 18:50

S.	D	FirstTime	Time	Text	Value	F...	F...	SrvTime	Status
M	0	2018-04-04 ...	2018-04-04 17:17:42:...	Free Memory (9.969785948338227%) is below the threshold 10.0%	9.9697...			2018-04-04 17:17:...	Active
N	0	2018-04-04 ...	2018-04-04 17:14:42:...	Free Memory: (10.020426401123451)%	10.020...			2018-04-04 17:14:...	Auto Closed Me
M	0	2018-04-04 ...	2018-04-04 16:35:42:...	Free Memory (9.977020298736116%) is below the threshold 10.0%	9.9770...			2018-04-04 16:35:...	Auto Closed Me
N	0	2018-04-04 ...	2018-04-04 16:26:43:...	Free Memory: (10.003404400187241)%	10.003...			2018-04-04 16:26:...	Auto Closed Me

You can **zoom-in** by selecting necessary area on the chart (press left mouse button and select region).

It is possible to select up to 5 different objects and display as compare graph:



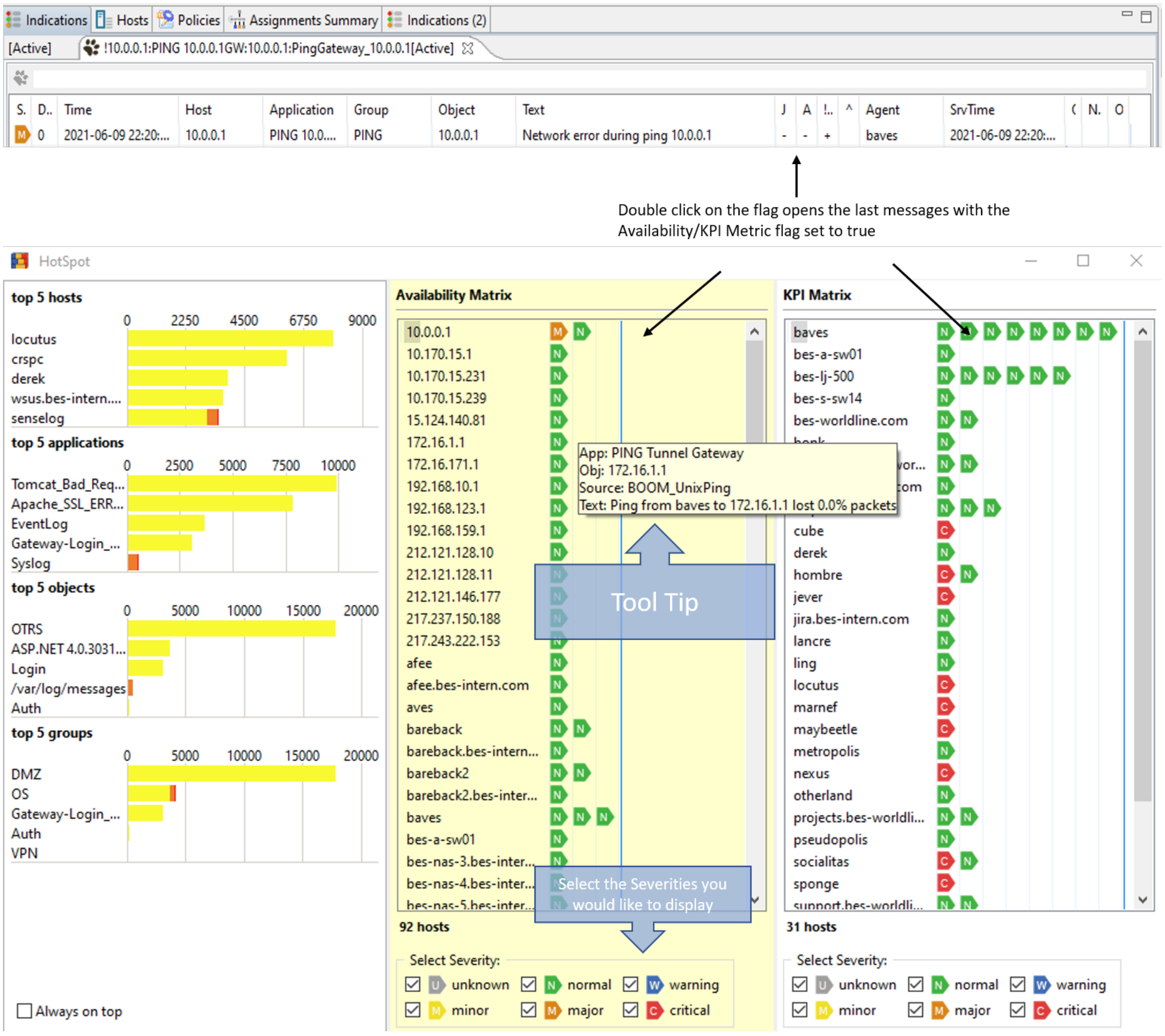
Agent: bnom | Host: bnom | App: Memory | Grp: OS | Obj: FreeMemory | Mon: Linux_MemSwap_Mon
Agent: nuffelstraat | Host: nuffelstraat | App: Memory | Grp: OS | Obj: FreeMemory | Mon: Linux_MemSwap_Mon
Agent: hoogstraat.bes-intern.com | Host: hoogstraat.bes-intern.com | App: Memory | Grp: OS | Obj: FreeMemory | Mon: Linux_MemSwap_Mon

S.	D...	FirstTime	Time	Text	Value	F.	F...	SrvTime	Status
C	0	2018-03-26 12:51...	2018-03-26 12:51...	Free Memory (1.4425314099581201%) is below the threshold 5.0%	1.442...			2018-03-26 12:50:...	Auto Closed Me
M	0	2018-03-26 12:48...	2018-03-26 12:48...	Free Memory (6.9641270781335393%) is below the threshold 10.0%	6.964...			2018-03-26 12:47:...	Auto Closed Me
M	0	2018-03-26 11:37...	2018-03-26 11:37...	Free Memory (9.933117859771514%) is below the threshold 10.0%	9.933...			2018-03-26 12:28:...	Auto Closed Me
N	0	2018-03-26 11:34...	2018-03-26 11:34...	Free Memory: (10.045121443886204)%	10.04...			2018-03-26 12:25:...	Auto Closed Me

6.19. HotSpot

The HotSpot is a small tool that gives you a quick status overview about:

- The top 5 most active hosts, applications, objects and groups that ordered by number of indications with severities higher that normal.
- The Availability Matrix shows the severity of last active indication that has the availability flag set. Every icon reflects a unique Host:Application:Object:Source key. The availability flag should be set in the condition of the policy to identify the indication to be related to an availability metric.
- The KPI (Key Performance Indicators) Matrix shows the severity of last active indication that has the KPI flag set. Every icon reflects a unique Host:Application:Object:Source key. The KPI flag should be set in the condition of the policy to identify the indication to be related to a service relevant KPI metric.



S.	D..	Time	Host	Application	Group	Object	Text	J	A	!..	^	Agent	SrvTime	(	N.	O
M	0	2021-06-09 22:20:...	10.0.0.1	PING 10.0....	PING	10.0.0.1	Network error during ping 10.0.0.1	-	-	+		baves	2021-06-09 22:20:...			

```
App: PING Tunnel Gateway
Obj: 172.16.1.1
Source: BOOM_UnixPing
Text: Ping from baves to 172.16.1.1 lost 0.0% packets
```

6.20. Pattern Validation

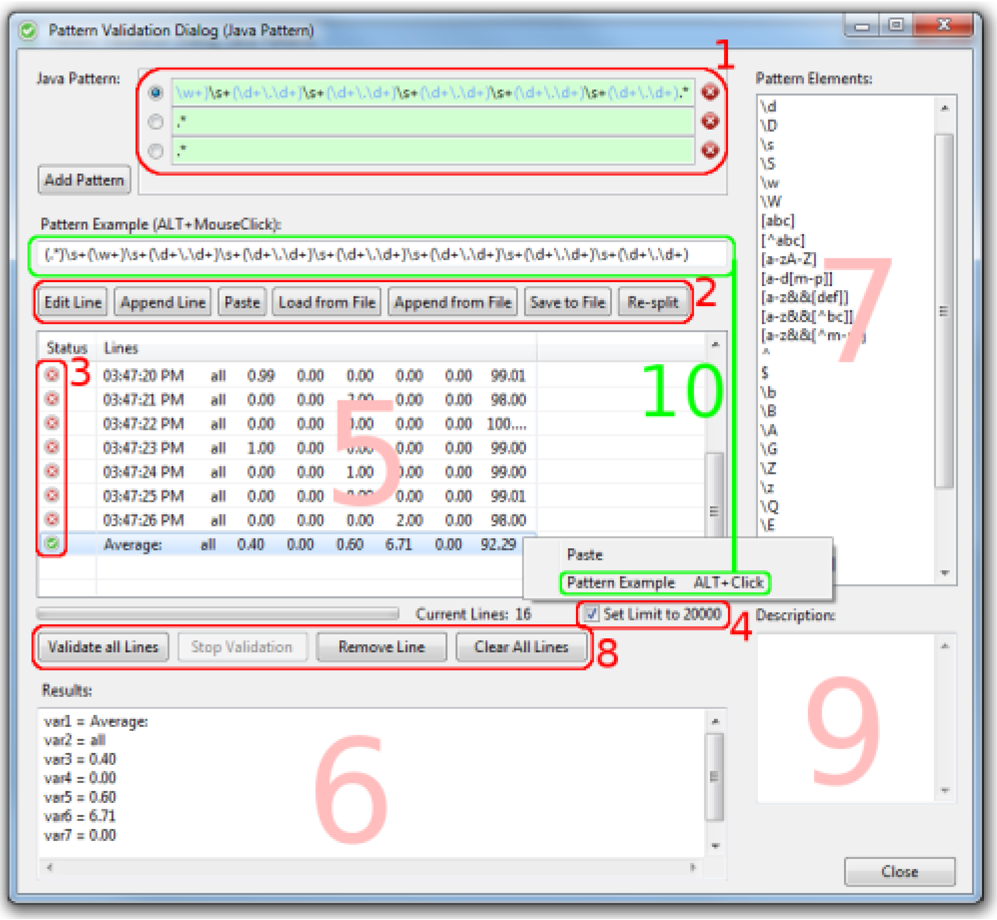
The "Pattern Validation Dialog" is designed to provide an easy and flexible way to create and test patterns against an user-defined text.

6.20.1. Pattern Validation Dialog (Java Pattern)

The most common way to use the Pattern Validation Dialog is the following:

1. Copy and paste the output of a command or any logfile entries into the "Pattern Validation Dialog"
2. Create a pattern example: select a line, right-click and select "Pattern Example"
3. Copy the example to the "Pattern Input Field" and modify/improve it according to your needs

Open the **Pattern Validation Dialog (Java Pattern)** view in the *boom* User Interface.



```
\w+)\s+(\d+\.\d+)\s+(\d+\.\d+)\s+(\d+\.\d+)\s+(\d+\.\d+)\s+(\d+\.\d+).*
.*
.*
```

```
(.*)\s+(\w+)\s+(\d+\.\d+)\s+(\d+\.\d+)\s+(\d+\.\d+)\s+(\d+\.\d+)\s+(\d+\.\d+)\s+(\d+\.\d+)
```

```
03:47:20 PM   all   0.99   0.00   0.00   0.00   0.00   99.01
03:47:21 PM   all   0.00   0.00   [illegible]   0.00   0.00   98.00
03:47:22 PM   all   0.00   0.00   [illegible]   0.00   0.00   100....
03:47:23 PM   all   1.00   0.00   [illegible]   0.00   0.00   99.00
03:47:24 PM   all   0.00   0.00   1.00   [illegible]   0.00   99.00
03:47:25 PM   all   0.00   0.00   [illegible]   0.00   0.00   99.01
03:47:26 PM   all   0.00   0.00   0.00   2.00   0.00   98.00
Average:     all   0.40   0.00   0.60   6.71   0.00   92.29
```

```
var1 = Average:
var2 = all
var3 = 0.40
var4 = 0.00
var5 = 0.60
var6 = 6.71
var7 = 0.00
```

```
\d
\D
\s
\S
\w
\W
[abc]
[^abc]
[a-zA-Z]
[a-d[m-p]]
[a-z&&[def]]
[a-z&&[^bc]]
[a-z&&[^m-[illegible]
^
$
\b
\B
\A
\G
\Z
\z
\Q
\E
```

1 Pattern Input field

Here you can add the Java patterns (with syntax highlighting support).
The user-defined text can be checked against multiple patterns.
There are three input boxes, only the selected one is used when you click validate, the purpose of multiple boxes is to offer the possibility to try out different patterns without losing the other approaches.

2 Table Operations

Like edit, append and remove a line or load data from file and save to file

Edit Line:

You can edit only one line at a time.

Append Line:

Appends a single line to the table.

Paste:

Paste (append) the data from the clipboard into the table.

Load from file:

Load a complete file to the table. Please note: All existing lines will be removed!

Append from file:

Load and append a complete file into the table. Please note: The existing lines will not be removed! The new data will be appended to the table.

Save to file:

Save the table content to a text file.

3 Validation Status

The validation status indicates if the line matches the pattern in the "Pattern Input" field.

- Red X means the line **did not** validate
- Green check means the line validated

Time

The Time field returns the required time for processing this line with this pattern in nanoseconds. This time should be used as indication if a pattern should be reworked because it requires too much processing time. Please note that there can be patterns that take more than a second or longer to be processed. The is time is in most cases caused by a very generic pattern, where the pattern matcher finds a wide variety of possibilities to match the incoming data. Such patterns, also called evil patterns, can stall the processing of the incoming data and severely impact the performance of the agent. Therefore such patterns should be changed, e.g. by specifying more specific chars or white-spaces in the pattern, which will in most cases solve the problem.

4 Limit number of lines

Here you can limit the number of lines for performance reasons

5 Input Box

Here you can add the text to be evaluated by the pattern in the pattern input field

6 Results area

Here are the variables extracted as a result of the pattern being applied over the selected line in the input box

7 Pattern elements

Here are the most common pattern elements for designing complex patterns. A description will appear below (9) when you select an element

8 Table Operations

Validate all Lines:

Check all lines in the input box (5) against the pattern (1) and output of the Validation Status (3)

Stop Validation:

Cancel the validation if it takes too long.

Remove line:

Remove selected line(s) from the input box (5).

Clear All Lines:

Clear all lines from the input box (5)

9 Pattern element description

A short description of the pattern element selected in (7)

10 Pattern Example:

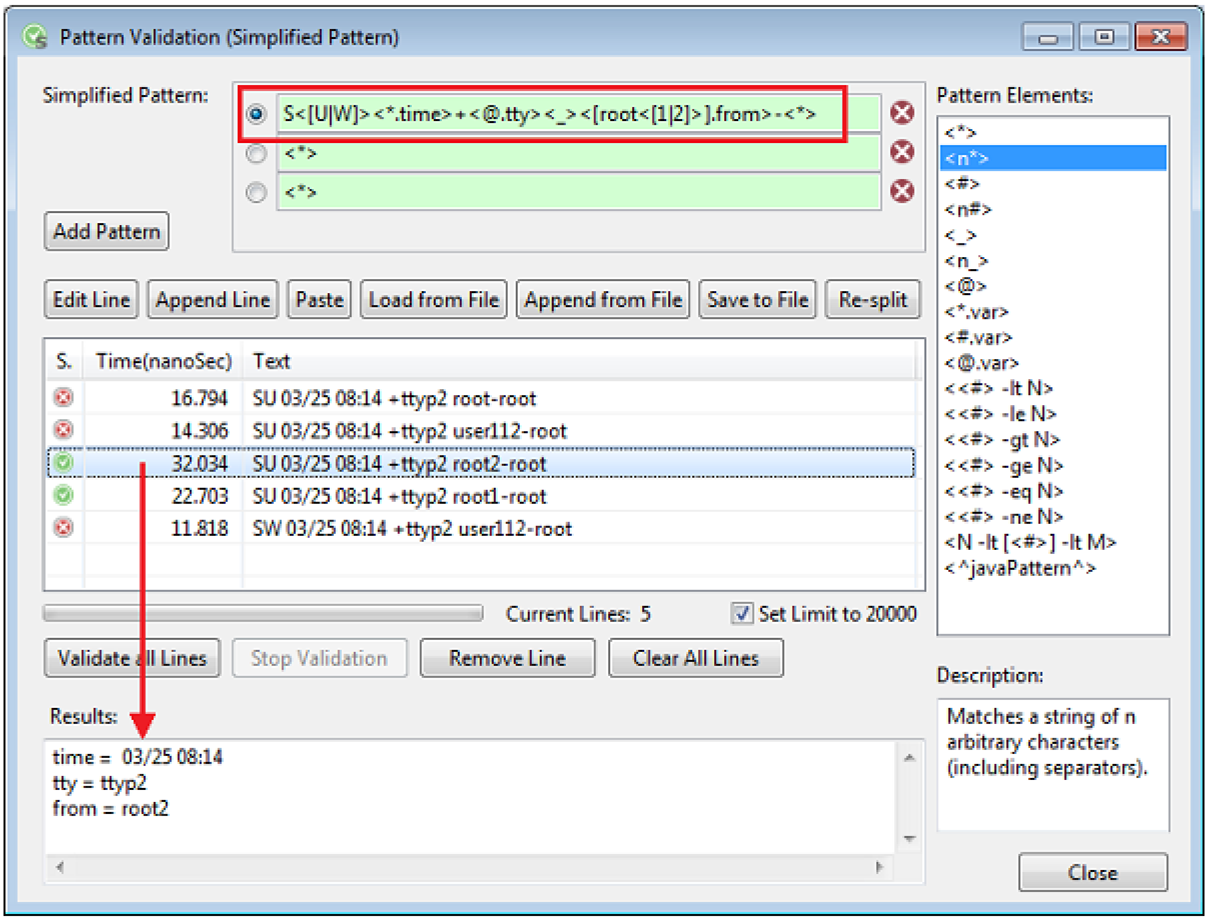
If you have added some text to the table and you would like to get an example pattern for a certain line, you can do the following:

- Select the "Pattern Example" function from the context menu (right-click on the selected line)
- or press ALT+Click on the appropriate table line at the same time.

6.20.2. Pattern Validation Dialog (Simplified Pattern)

The **Simplified Pattern Validation Dialog** has the same functions as the **Java Pattern Validation Dialog**. The only difference is that the used patterns are simplified patterns instead of Java patterns.

Open the **Pattern Validation Dialog (Simplified Pattern)** view in the *boom* User Interface.



Simplified Pattern:
- S<[U|W]><*.time>+<@.tty><_><[root<[1|2]>].from>-<*>
- <*>
- <*>

Pattern Elements:
<*>
<n*>
<#>
<n#>
<_>
<n_>
<@>
<*.var>
<#.var>
<@.var>
<<#> -lt N>
<<#> -le N>
<<#> -gt N>
<<#> -ge N>
<<#> -eq N>
<<#> -ne N>
<N -lt [<#>] -lt M>
<^javaPattern^>

S.	Time(nanoSec)	Text
✗	16.794	SU 03/25 08:14 +ttyp2 root-root
✗	14.306	SU 03/25 08:14 +ttyp2 user112-root
✓	32.034	SU 03/25 08:14 +ttyp2 root2-root
✓	22.703	SU 03/25 08:14 +ttyp2 root1-root
✗	11.818	SW 03/25 08:14 +ttyp2 user112-root

Current Lines: 5 [✓] Set Limit to 20000

Results:

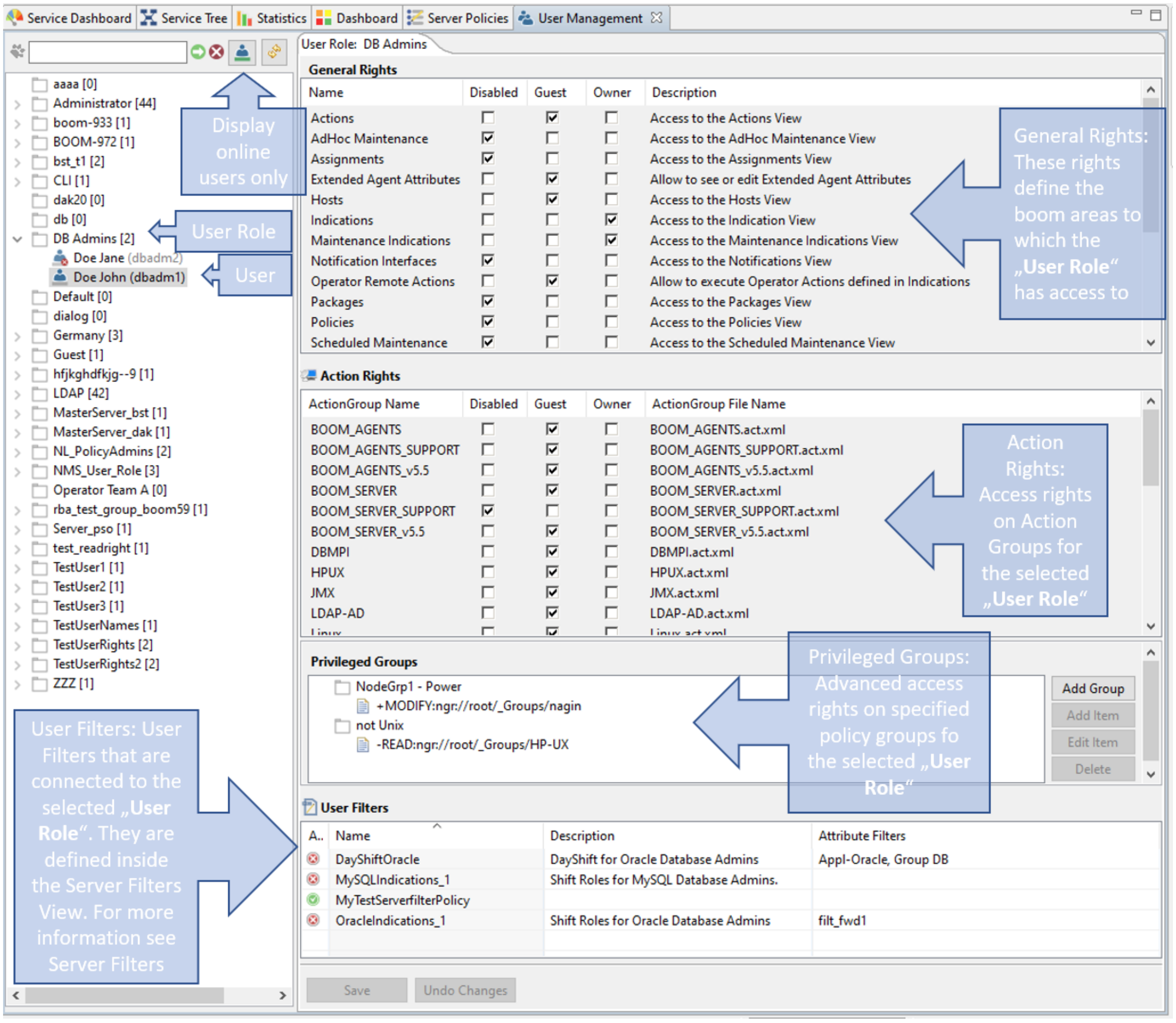
```
time =  03/25 08:14
tty = ttyp2
from = root2
```

Description: Matches a string of n arbitrary characters (including separators).

6.21. User Management

6.21.1. General Information

With the *boom* integrated user management you can manage your user accounts. User accounts let you control who can access the *boom* console and dashboards. You can create accounts that allow full access to *boom* or accounts that allow only access to certain areas. The *boom* user management comes with the default "Administrator" role and the default "admin" user. The "Administrator" role has full rights and cannot be changed or deleted! Also the "admin" user cannot be deleted or moved to another role. The *boom* user management is divided into two sections, the left section shows a list of all existing user roles with the user accounts assigned to them. The right area displays all the rights that belong to the selected user role.


Service Dashboard
Service Tree
Statistics
Dashboard
Server Policies
User Management
Display online users only
User Role
User
General Rights: These rights define the boom areas to which the „User Role“ has access to
Action Rights: Access rights on Action Groups for the selected „User Role“
Privileged Groups: Advanced access rights on specified policy groups fo the selected „User Role“
User Filters: User Filters that are connected to the selected „User Role“. They are defined inside the Server Filters View. For more information see Server Filters
Add Group
Add Item
Edit Item
Delete
Save
Undo Changes

aaaa [0]
Administrator [44]
boom-933 [1]
BOOM-972 [1]
bst_t1 [2]
CLI [1]
dak20 [0]
db [0]
DB Admins [2]
Doe Jane (dbadm2)
Doe John (dbadm1)
Default [0]
dialog [0]
Germany [3]
Guest [1]
hfjkghdfkjg--9 [1]
LDAP [42]
MasterServer_bst [1]
MasterServer_dak [1]
NL_PolicyAdmins [2]
NMS_User_Role [3]
Operator Team A [0]
rba_test_group_boom59 [1]
Server_pso [1]
test_readright [1]
TestUser1 [1]
TestUser2 [1]
TestUser3 [1]
TestUserNames [1]
TestUserRights [2]
TestUserRights2 [2]
ZZZ [1]

User Role: DB Admins

General Rights

Name	Disabled	Guest	Owner	Description
Actions	☐	☑	☐	Access to the Actions View
AdHoc Maintenance	☑	☐	☐	Access to the AdHoc Maintenance View
Assignments	☑	☐	☐	Access to the Assignments View
Extended Agent Attributes	☐	☑	☐	Allow to see or edit Extended Agent Attributes
Hosts	☐	☑	☐	Access to the Hosts View
Indications	☐	☐	☑	Access to the Indication View
Maintenance Indications	☐	☐	☑	Access to the Maintenance Indications View
Notification Interfaces	☑	☐	☐	Access to the Notifications View
Operator Remote Actions	☐	☑	☐	Allow to execute Operator Actions defined in Indications
Packages	☑	☐	☐	Access to the Packages View
Policies	☑	☐	☐	Access to the Policies View
Scheduled Maintenance	☑	☐	☐	Access to the Scheduled Maintenance View

Action Rights

ActionGroup Name	Disabled	Guest	Owner	ActionGroup File Name
BOOM_AGENTS	☐	☑	☐	BOOM_AGENTS.act.xml
BOOM_AGENTS_SUPPORT	☐	☑	☐	BOOM_AGENTS_SUPPORT.act.xml
BOOM_AGENTS_v5.5	☐	☑	☐	BOOM_AGENTS_v5.5.act.xml
BOOM_SERVER	☐	☑	☐	BOOM_SERVER.act.xml
BOOM_SERVER_SUPPORT	☑	☐	☐	BOOM_SERVER_SUPPORT.act.xml
BOOM_SERVER_v5.5	☐	☑	☐	BOOM_SERVER_v5.5.act.xml
DBMPI	☐	☑	☐	DBMPI.act.xml
HPUX	☐	☑	☐	HPUX.act.xml
JMX	☐	☑	☐	JMX.act.xml
LDAP-AD	☐	☑	☐	LDAP-AD.act.xml

Privileged Groups

NodeGrp1 - Power
+MODIFY:ngr://root/_Groups/nagin
not Unix
-READ:ngr://root/_Groups/HP-UX

User Filters

A..	Name	Description	Attribute Filters
●	DayShiftOracle	DayShift for Oracle Database Admins	Appl-Oracle, Group DB
●	MySQLIndications_1	Shift Roles for MySQL Database Admins.	
●	MyTestServerfilterPolicy		
●	OracleIndications_1	Shift Roles for Oracle Database Admins	filt_fwd1

6.21.2. User Roles and Rights

User accounts are assigned to user roles. User roles define all rights a user have.

If you select a user role in the user management view , you will get for groups of access rights that are assigned to a single user role.

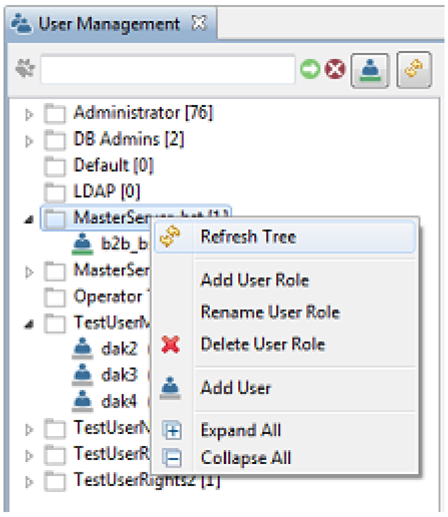
- "General Rights" specify a list of all *boom* workbench areas which can be restricted by user rights.
- "Action Rights" define the access rights to the *boom* actions groups.

There are three possible access rights:

Disabled	**no access** (this View Area or Action Group will not be visible to the user)
Guest	**read only access**
Owner	**full access**

ℹ	The access rights of the "Administrator" role cannot be changed!

Right-click on a User Role to open the role context menu:

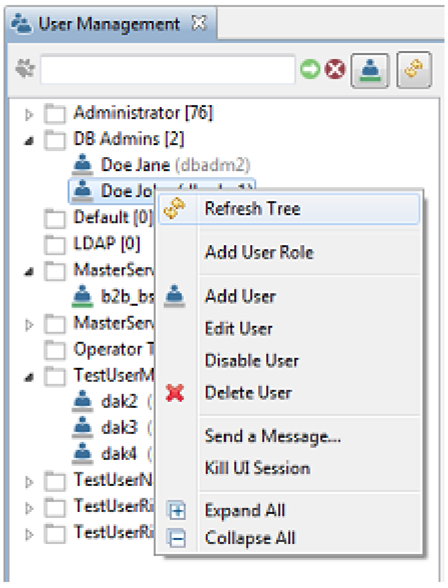


Add User Role:	Adds a new user role.
Rename User Role:	The rename action is disabled for the "Administrator" Role. All other user roles can be renamed. This "Administrator" role is not editable at all!
Delete User Role:	The delete action is disabled for the "Administrator" Role. All other user roles can be deleted. This "Administrator" role is not editable at all!
Add User:	A new user can be added to any user role. This action opens the User Detail Dialog where all user information can be specified.

6.21.3. User Accounts

A user account has to be created for every user who wants to login to the *boom* workbench. Once an user account has been created, you can move the user by drag & drop to a different user role.

Right-click on a single User to open the user context menu:



Add User Role:

Adds a new user role.

Add User:

Adds a new user account. This action opens the "User Detail Dialog" where all user information can be specified.

Edit User:

All user specific information can be specified in the "User Details Dialog".

Delete User:

Deleting a user removes him completely from the system.

Disable User:

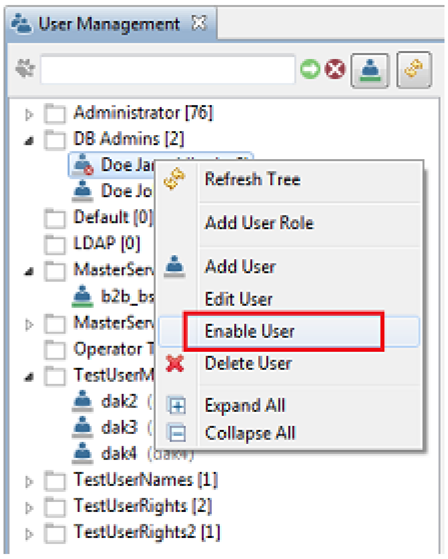
Disabling a user prevents the user from logging in to the *boom* workbench. You would typically do this when a user leaves your organization.

Send a message:

Allows you to send a message to a selected user.

Kill UI Session:

Allows you to log off any signed on user from the user interface.



Enable User:

This will enable an user account.

Color coding of users: online vs. offline, *boom* user vs. LDAP user

For details on LDAP users and how to set up LDAP authentication please refer to the LDAP Authentication section.

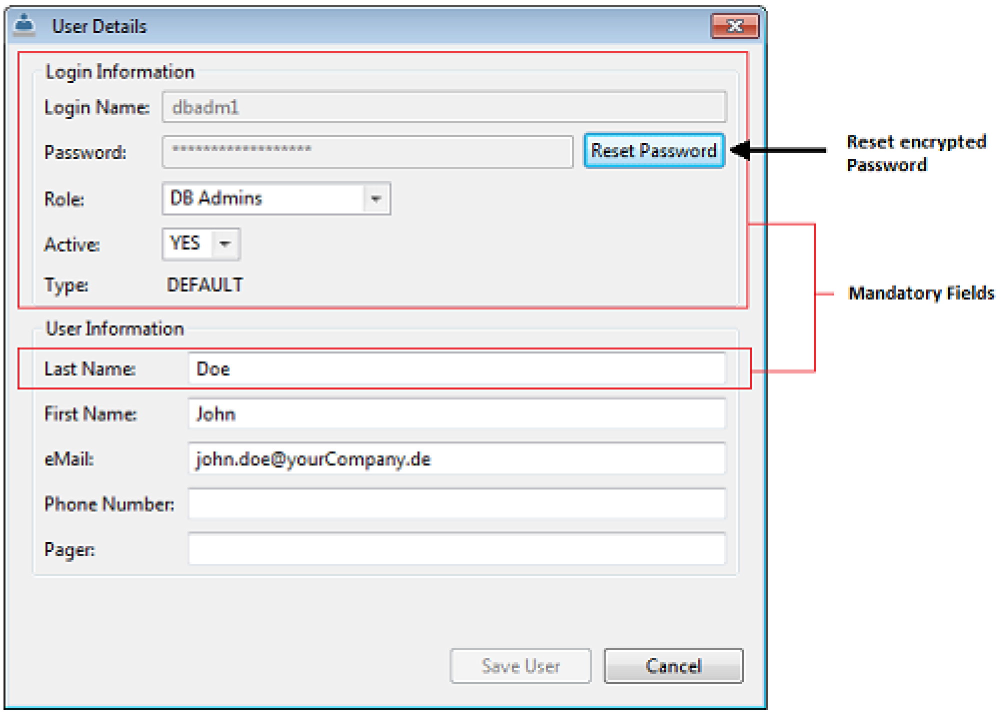
Logged on users are depicted with a **green bar** beneath the corresponding user icon, while the offline users have none. Refresh the corresponding top folder in the user management tab to see changes, who is logged on.

Native *boom* users (configured in the *boom* database) are depicted by a **blue user icon**, while LDAP users are presented by a **brown user icon**.

: *boom* user (online)
: LDAP user (online)
: *boom* user (offline)
: LDAP user (offline)

6.21.4. User Details Dialog

All user specific attributes are managed in the "User Details Dialog". When adding a new user, all mandatory fields have to be specified. Once the user account has been saved, the login name is the only information that cannot be changed any more.



User Details

Login Information

Login Name: dbadm1

Password: ******************

Role: DB Admins

Active: YES

Type: DEFAULT

User Information

Last Name: Doe

First Name: John

eMail: john.doe@yourCompany.de

Phone Number:

Pager:

Field Description:

Login Name

The login name is case insensitive and cannot be modified once the user has been created!

Password

Passwords are case sensitive! When creating a password you have to pay attention to:

- upper case and lower case letters
- passwords must not contain any blanks
- avoid really short passwords

Reset Password

It is not possible to readout a users password because the password is encrypted. Only an administrator has the right to reset a password. The administrator has to enter a new password which will overwrite the old one.

Active

Indicates if the user is active or if he has been disabled.

Role

Add the user to an existing user role.

Last Name

The last name of the user is mandatory, first name can be empty.

First Name

First name of the user. This field is not mandatory and can be empty.

eMail

Email of the user. This field is not mandatory and can be empty.

Phone Number

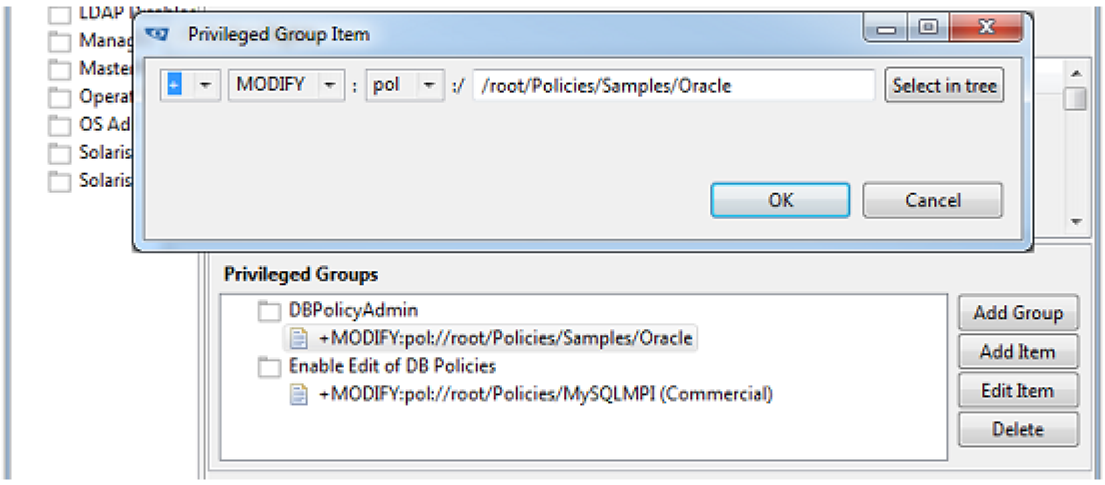
Phone number of the user. Since this field is a text field, no special format needs to be considered. This field is not mandatory and can be empty.

Pager

If the user has a pager, you can enter the number here. Since this field is a text field, no special format needs to be considered. This field is not mandatory and can be empty.

6.21.5. Privileged Groups

The general user rights are defining the overall access rights per user role on the different areas and views of boom. The Privileged Groups allow to define granular access rights for user roles on policy groups, assignments and host groups. Privilege Groups are defined with the user role and each one contains a list of access rights for specific objects that override the global rights.
The functionality for restricting the access rights for indications is available through the user filters that allow to limit the scope of visible indications based on combinations of filters for indication attributes like the application, indication group, host, hostgroups, etc. For details about the user filters please refer to the chapter Server Filters.
Please be aware that the visibility of a host group can be configured by the Priviledged Groups, while the user filters only influence, which indications are displayed to the user.


Privileged Group Item
+
MODIFY
pol
:/
/root/Policies/Samples/Oracle
Select in tree
OK
Cancel
Privileged Groups
DBPolicyAdmin
+MODIFY:pol://root/Policies/Samples/Oracle
Enable Edit of DB Policies
+MODIFY:pol://root/Policies/MySQLMPI (Commercial)
Add Group
Add Item
Edit Item
Delete

Each Item of a Privilege Group corresponds to an access control entry for a specific Policy Group, Assignment Group or Host Group. An Item consists of:

```
<Action Permission> <Action>:<ObjectType>:<URL|ID>
```

```
<ActionPermission>
    *: exclusive rights
    +: add right
    -: substract right
<Action>
    Modify
    Read (available only for Object Type Policy Group)
<ObjectType>
    pol:  Policy Group
    pga: Assignment Group
    ngr:  Node/Host Group
<URL|ID>
    Path or ID of a Group
```

Exclusive (*) rights: the "exclusive" right is a special right that affects other user roles rights. Giving a user role the ***Modify** right will override the global "full rights" that other user roles might have. This means if one or more user roles have exclusive rights on a policy group, no one else can anymore e.g. edit the policies within this group/subtree, unless they have also the exclusive right on this group or a related group in the subtree.

Add (+) rights: giving a user role that has global "read only" rights the +**Modify** right on a policy group will give the users of this role the right to edit all the policies in this policy group/subtree, as long as these are not restricted by an exclusive right.

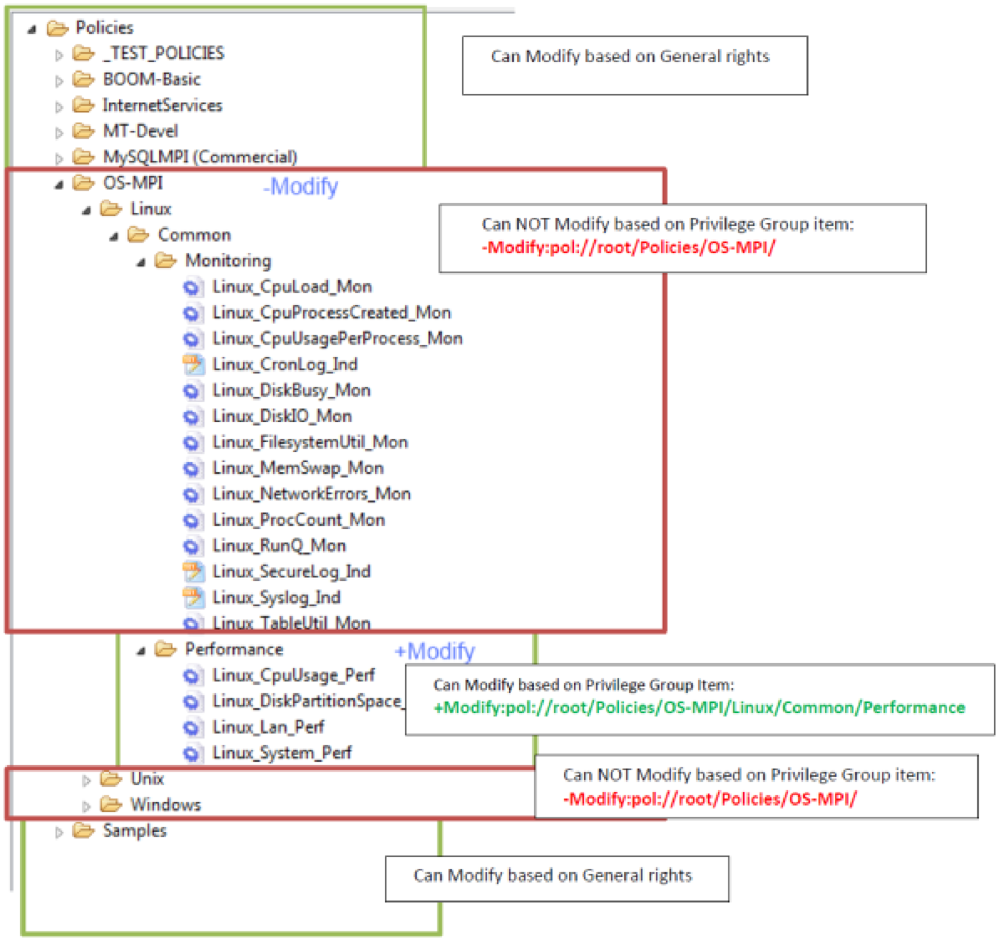
Substract(-) rights: giving a user role that has global "full rights" the –**Modify** right on a policy group will restrict the users of this role to have just read-only access to this policy group/subtree.

A Privilege Group Item will modify rights recursive, so if a Policy Group has the +**Modify** right, all sub-elements inherit the same right as well. Sub-elements can have own Item Rules that override the inherited access rights with other privileges. The result privilege scope will be a tree, where the particular element privileges will be calculated upwards.

To avoid confusion by applying privilege to a node group that contains an Agent which is also linked to other node groups - the following rules are valid:

- if an User has Guest access rights to the full tree and an enforcement (+**Modify**) of rights on one of the Node Group is added, then he becomes Owner right to all related Agents independent from other links.
  (for Guests: "Enforce right" has higher priority)
- if an User has Owner access rights to the full tree and a restriction (-**Modify**) of rights on one of the Node Groups is added, then all related Agents will be read-only, independent from other links.
  (for Owner "Restrict right" has higher priority)

Example: User Role with general "owner" rights for Policies and two Item rules:
- Modify:pol://root/Policies/OS-MPI/
+ **Modify:pol://root/Policies/OS-MPI/Linux/Common/Performance**



Can Modify based on General rights

Can NOT Modify based on Privilege Group item:
-Modify:pol://root/Policies/OS-MPI/

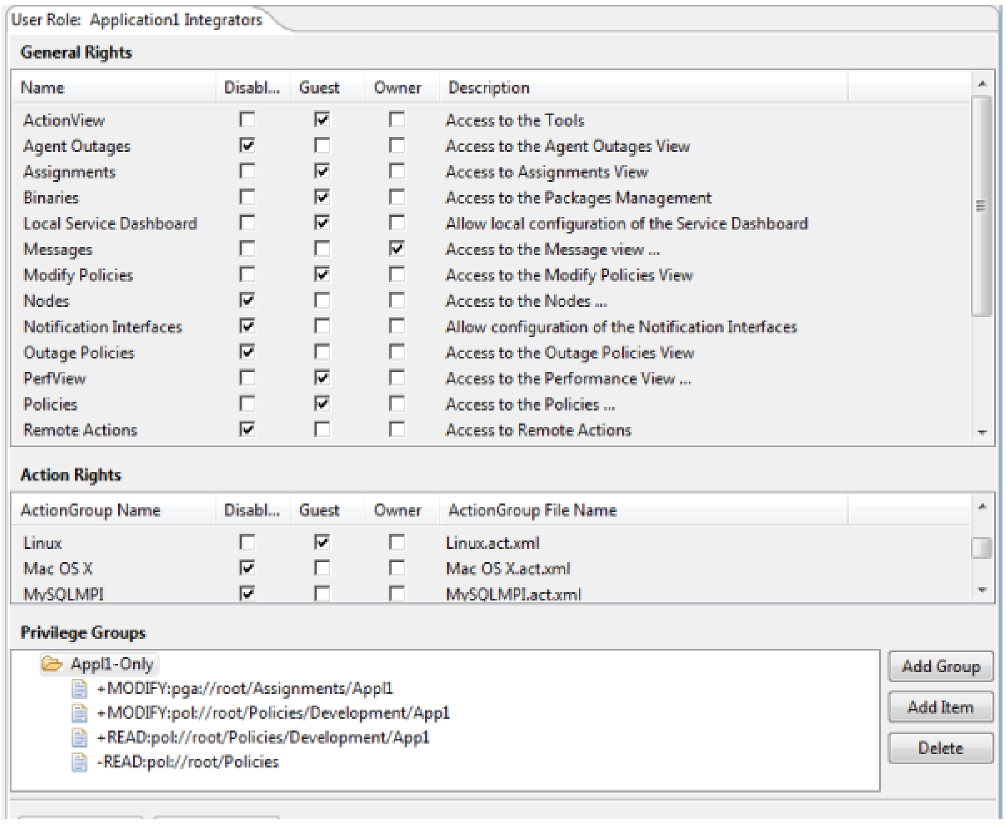
Can Modify based on Privilege Group Item:
+Modify:pol://root/Policies/OS-MPI/Linux/Common/Performance

Can NOT Modify based on Privilege Group item:
-Modify:pol://root/Policies/OS-MPI/

Can Modify based on General rights

Example:

To configure a user role for a group of admins that should be able to modify the policies for a specific application, but can not see other policies on the system:
In the General Rights section grant the "Guest" rights for the Policy and Assignment areas. As first Item Rule revoke the READ access to the Policy Root, override this with the READ and MODIFY rights for the specific Policy Group(s) that they should be able to modify and if necessary give them also MODIFY rights on the according Assignment Group.
Such kind of configuration can also be used to implement multi tenancy for the policies and assignment groups. In such a scenario the exclusive (*) MODIFY right can be used to make sure that no one else can modify these policies.


User Role: Application1 Integrators

General Rights

Name	Disabl...	Guest	Owner	Description
ActionView	☐	☑	☐	Access to the Tools
Agent Outages	☑	☐	☐	Access to the Agent Outages View
Assignments	☐	☑	☐	Access to Assignments View
Binaries	☐	☑	☐	Access to the Packages Management
Local Service Dashboard	☐	☑	☐	Allow local configuration of the Service Dashboard
Messages	☐	☐	☑	Access to the Message view ...
Modify Policies	☐	☑	☐	Access to the Modify Policies View
Nodes	☑	☐	☐	Access to the Nodes ...
Notification Interfaces	☑	☐	☐	Allow configuration of the Notification Interfaces
Outage Policies	☑	☐	☐	Access to the Outage Policies View
PerfView	☐	☑	☐	Access to the Performance View ...
Policies	☐	☑	☐	Access to the Policies ...
Remote Actions	☑	☐	☐	Access to Remote Actions

Action Rights

ActionGroup Name	Disabl...	Guest	Owner	ActionGroup File Name
Linux	☐	☑	☐	Linux.act.xml
Mac OS X	☑	☐	☐	Mac OS X.act.xml
MySQLMPI	☑	☐	☐	MySQLMPI.act.xml

Privilege Groups

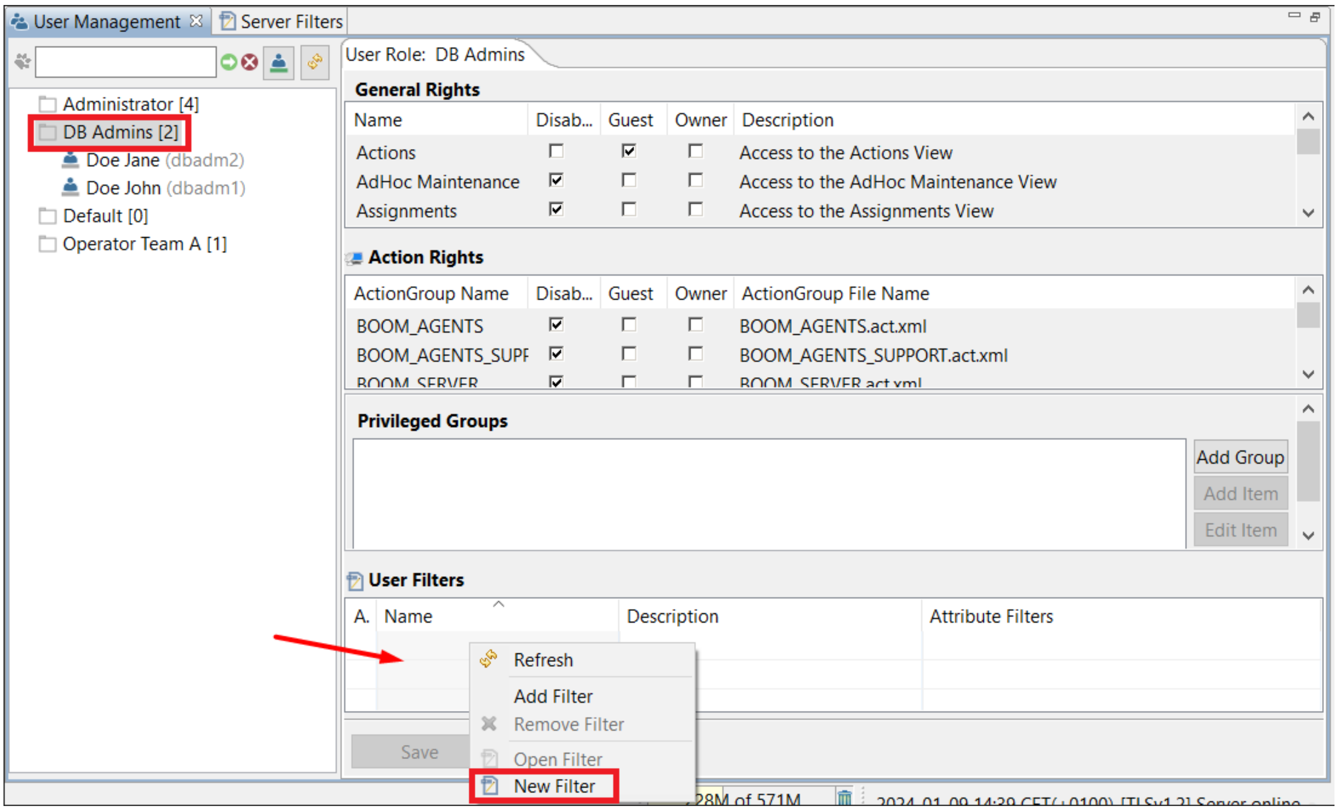
- Appl1-Only
  - +MODIFY:pga://root/Assignments/Appl1
  - +MODIFY:pol://root/Policies/Development/Appl1
  - +READ:pol://root/Policies/Development/Appl1
  - -READ:pol://root/Policies

6.21.6. User Filters

An user filter can be created/added for every user group who wants to login to the *boom* workbench. Once an user filter has been created or an already existing user filter has been assigned, UI access and operation possibilities handled according the specified user filter rules.

Create:

Select intended User Role and Right-click on User Filters window to open the content menu and choose "New Filter"



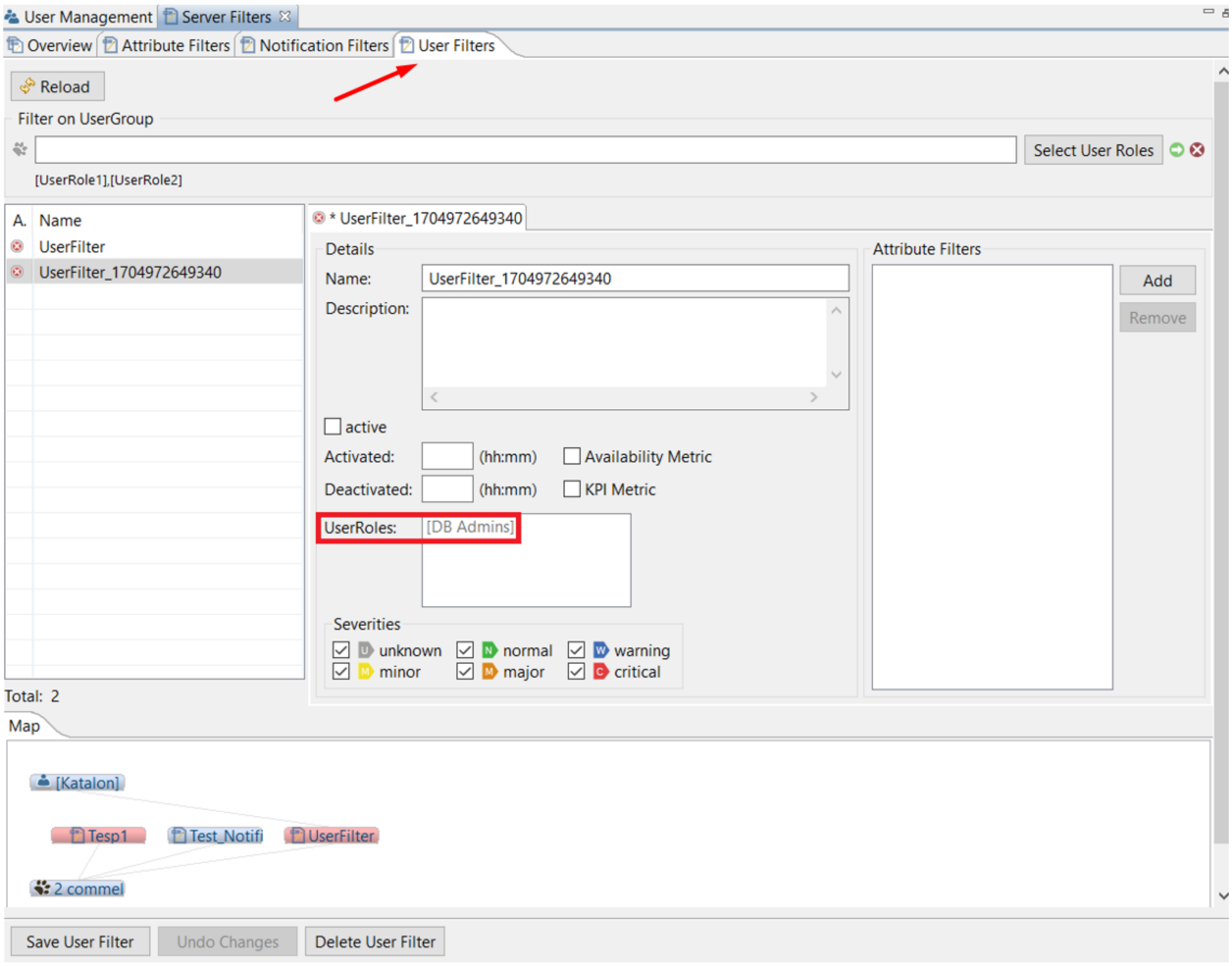
Name	Disab...	Guest	Owner	Description
Actions	☐	✓	☐	Access to the Actions View
AdHoc Maintenance	✓	☐	☐	Access to the AdHoc Maintenance View
Assignments	✓	☐	☐	Access to the Assignments View

ActionGroup Name	Disab...	Guest	Owner	ActionGroup File Name
BOOM_AGENTS	✓	☐	☐	BOOM_AGENTS.act.xml
BOOM_AGENTS_SUPF	✓	☐	☐	BOOM_AGENTS_SUPPORT.act.xml
BOOM_SERVER	✓	☐	☐	BOOM_SERVER.act.xml

A.	Name	Description	Attribute Filters

28M of 571M 2024-01-09 14:39 CET(+0100) [TLSv1.2] Server online

"New Filter" opens User Filters Tab of Server Filters window



User Filter Details:

Name

This is the unique name of the User Filter.

Description

A description of the User Filter.

active

This will set the User Filter status to active. A disabled User Filter will not change the visibility of indications.

Activated

This is the activation time.

Deactivated

This is the deactivation time.

Availability Metric

Only indications having the Availability Flag set will match this condition.

KPI Metric

Only indications having the KPI Metric Flag set will match this condition.

User Roles

That user role will be found here which was selected inside User Management window done before.

Severities

Only indications with selected severities match with the condition.

Attribute Filters:

Add

Opens a pop-up dialog that allows to select from the defined Attribute Filters which ones should be used by the User Filter. The Attribute Filters are connected with a logical "And", meaning all Attribute Filters must be matching.

Remove

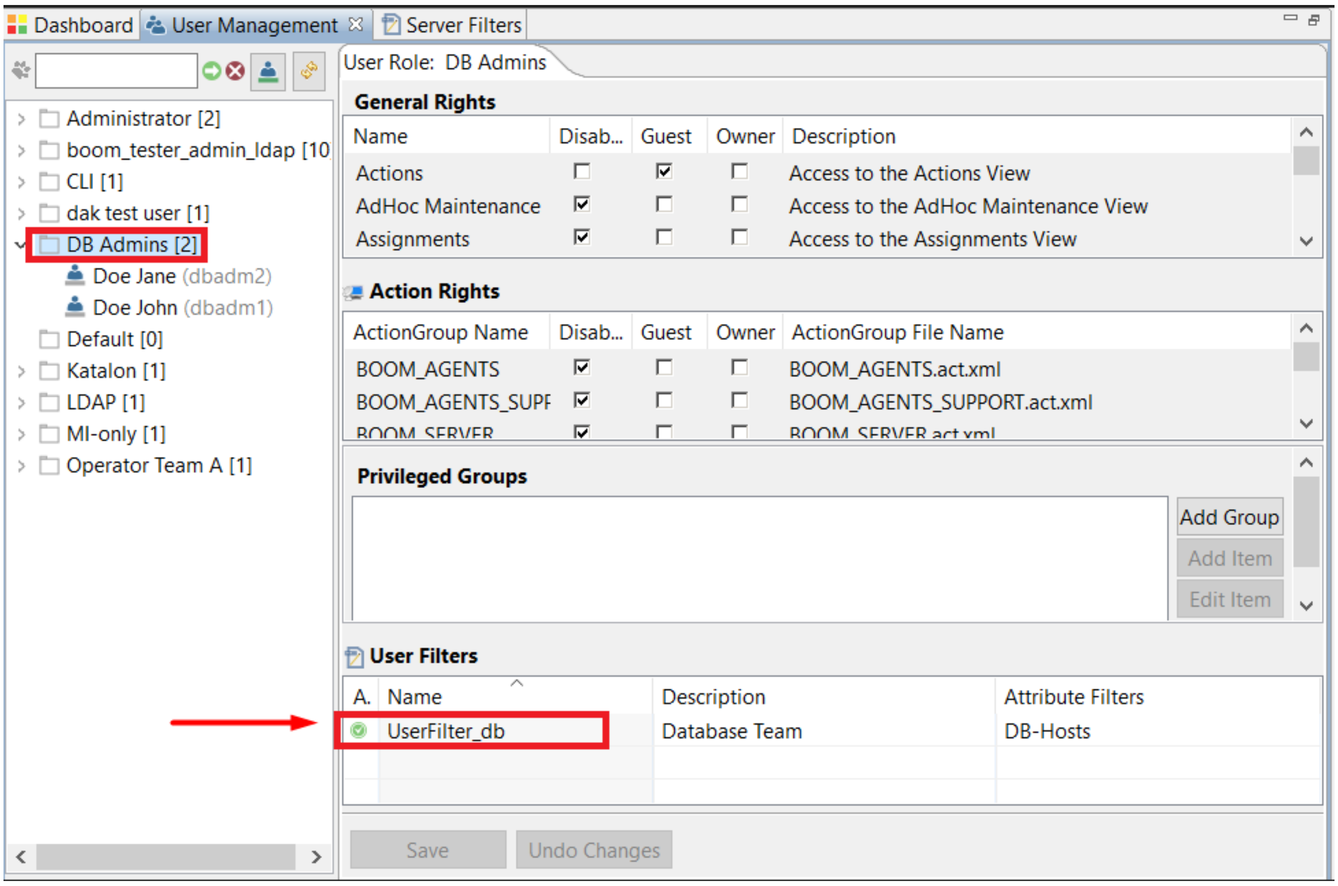
Removes selected Attribute Filter.

Map:

Tab

The Map view gives an overview about the relation between User Filters, Attribute Filters and affected user roles.

As soon as User Filter configuration is complete and saved it appears within the related User Filters window because of User Role to User Filter assignment done internally.



Name	Disab...	Guest	Owner	Description
Actions	☐	☑	☐	Access to the Actions View
AdHoc Maintenance	☑	☐	☐	Access to the AdHoc Maintenance View
Assignments	☑	☐	☐	Access to the Assignments View

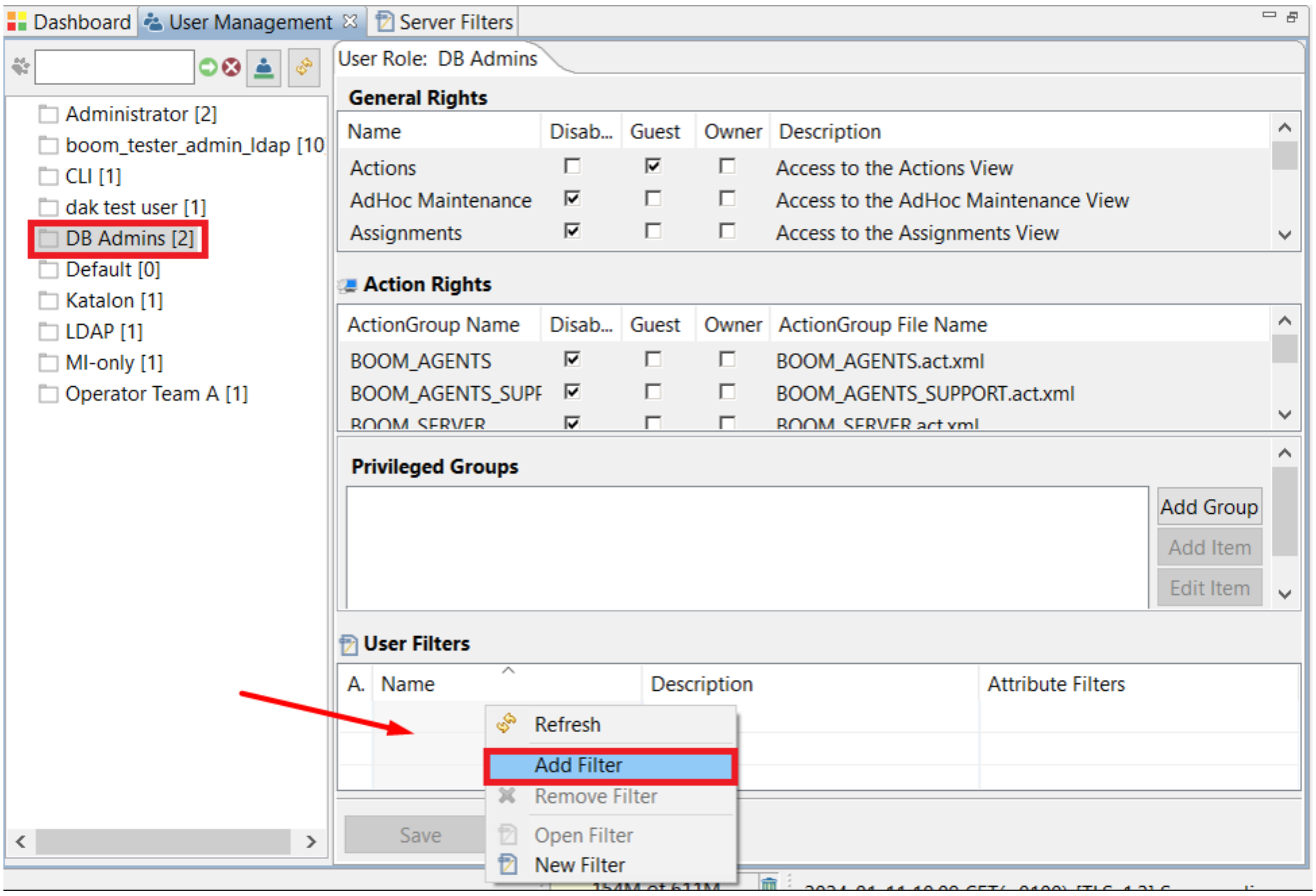
ActionGroup Name	Disab...	Guest	Owner	ActionGroup File Name
BOOM_AGENTS	☑	☐	☐	BOOM_AGENTS.act.xml
BOOM_AGENTS_SUPF	☑	☐	☐	BOOM_AGENTS_SUPPORT.act.xml
BOOM_SERVER	☑	☐	☐	BOOM_SERVER.act.xml

A.	Name	Description	Attribute Filters
●	UserFilter_db	Database Team	DB-Hosts

Add:

Requires existence of already existing User Filters.

Select intended User Role and Right-click on User Filters window to open the content menu and choose "Add Filter"



General Rights

Name	Disab...	Guest	Owner	Description
Actions	☐	☑	☐	Access to the Actions View
AdHoc Maintenance	☑	☐	☐	Access to the AdHoc Maintenance View
Assignments	☑	☐	☐	Access to the Assignments View

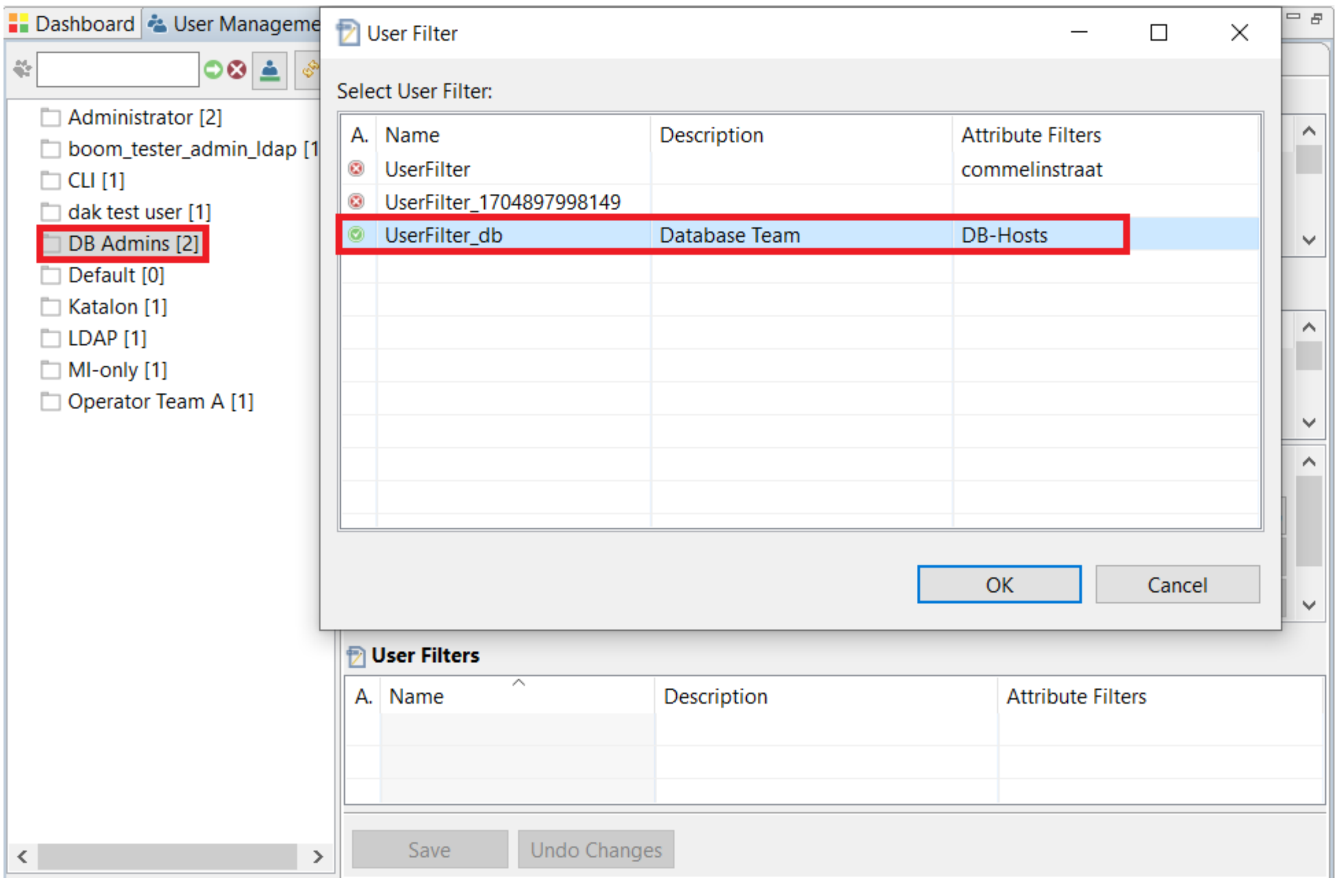
Action Rights

ActionGroup Name	Disab...	Guest	Owner	ActionGroup File Name
BOOM_AGENTS	☑	☐	☐	BOOM_AGENTS.act.xml
BOOM_AGENTS_SUPF	☑	☐	☐	BOOM_AGENTS_SUPPORT.act.xml
BOOM_SERVER	☑	☐	☐	BOOM_SERVER.act.xml

User Filters

A.	Name	Description	Attribute Filters

"Add Filter" opens User Filter window presenting already existing User Filters



Select User Filter:

A.	Name	Description	Attribute Filters
⊗	UserFilter		commelinstraat
⊗	UserFilter_1704897998149		
✓	UserFilter_db	Database Team	DB-Hosts

User Filters

A.	Name	Description	Attribute Filters

As soon as an User Filter has been chosen and confirmed by "OK" it appears within the user role related User Filters window of the User Management.

ℹ	More than one User Filter can be added to one User Role.

💡	One User Filter can also be linked to several User Roles.

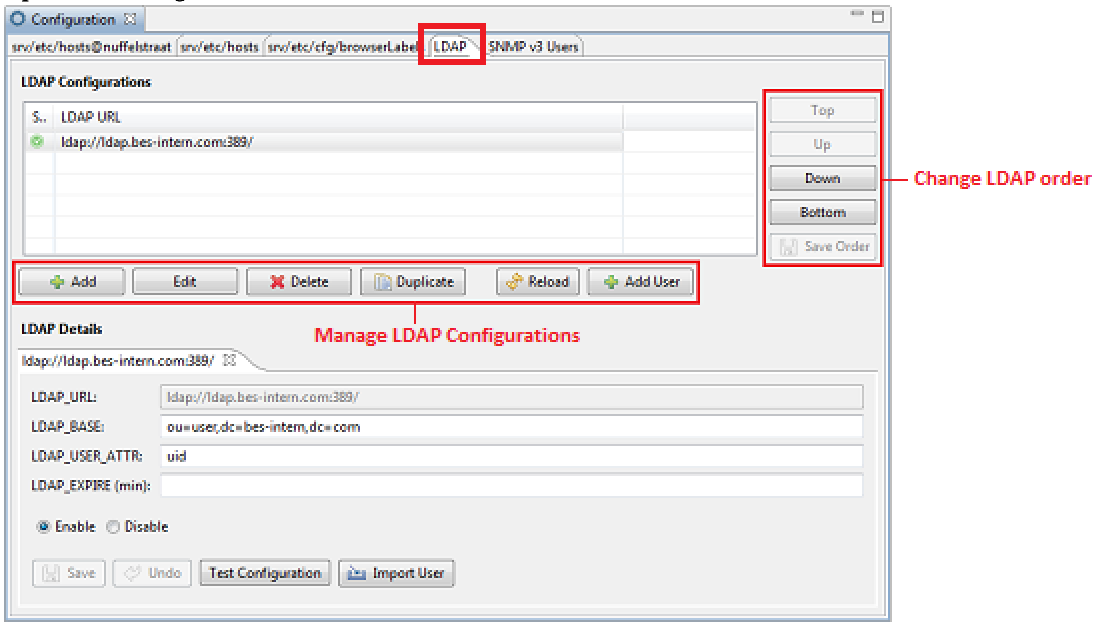
6.21.7. LDAP Authentication

With LDAP authentication configured, during a login *boom* will forward authentication requests to the configured LDAP server(s).
If the first LDAP server reports back that the user is unknown or the LDAP server is not reachable, the next configured LDAP server will be used. If all LDAP servers failed to know the user or are not reachable, *boom* will authenticate the user against the native *boom* users stored in the database.

ℹ	if an LDAP server responds that the password is incorrect *boom* will not try to authenticate against the next LDAP server, but will return a **Login failed** message to the user.

Setting up the LDAP Authentication

Open the Configuration View and choose the LDAP Tab.



S..	LDAP URL
●	ldap://ldap.bes-intern.com:389/

ldap://ldap.bes-intern.com:389/

LDAP_URL: ldap://ldap.bes-intern.com:389/
LDAP_BASE: ou=user,dc=bes-intern,dc=com
LDAP_USER_ATTR: uid
LDAP_EXPIRE (min):

[✓ Enable] Disable

The LDAP Configurations will show all currently configured LDAP servers.

- Press **Add** to enter the LDAP connection details or **Edit** to change the LDAP settings

```
    LDAP_URL=ldap://ldapserver:389

    LDAP_BASE=dc=company,dc=com

    LDAP_USER_ATTR=uid (attribute in LDAP that contains username, i.e. uid (for linux, OpenLDAP), empty
for Active Directory).

    LDAP_EXPIRE=60     (session expiration time in minutes, i.e. 60)
```

  Check the LDAP configuration and access by pressing Test Configuration. After this test was successful Enable and Save the LDAP configuration.
  To use LDAP over SLL (LDAPS) just specify the LDAP_URL and port as

```
LDAP_URL=ldaps://ldapserver:636
```

boom allows to specify the LDAP_BASE attribute as domain extension in email format if the connected LDAP Server is a Microsoft Active Directory. This allows the users to login with their short name instead of the full qualified main address e.g.

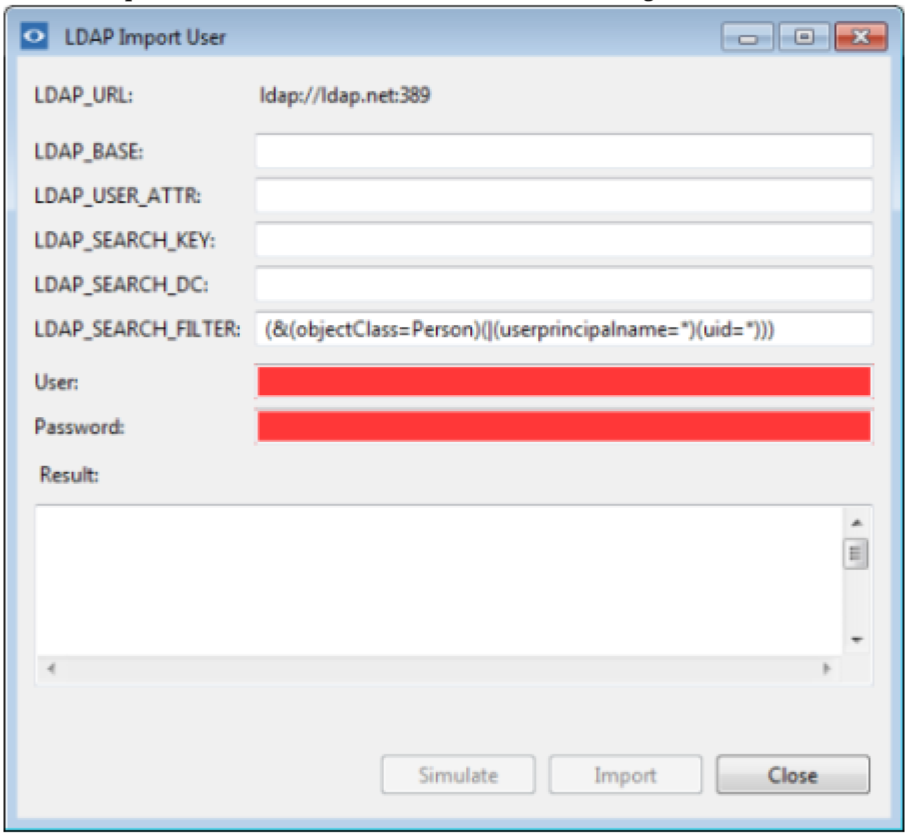
Listing 1. the user john has to login with john@mycompany.local

```
setting
  LDAP_BASE=@mycompany.local
allows to login simply with "john"
```

ℹ	If special LDAP parameters are necessary, please use the LDAP Actions instead of the UI.

Users can be imported from a LDAP (or several) server(s) into *boom*. Initially, such imported users are per default disabled and placed in the LDAP role folder.

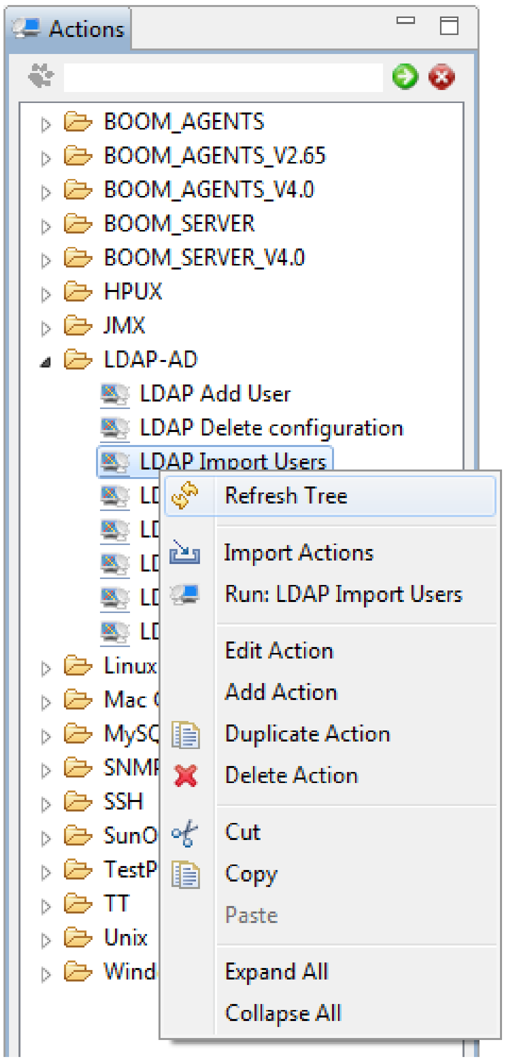
- Use the Import User button in the LDAP Tab of the Configuration view


LDAP Import User

LDAP_URL: ldap://ldap.net:389
LDAP_BASE:
LDAP_USER_ATTR:
LDAP_SEARCH_KEY:
LDAP_SEARCH_DC:
LDAP_SEARCH_FILTER: (&(objectClass=Person)(|(userprincipalname=*)(uid=*)))
User:
Password:
Result:

or

- use the server action "LDAP Import Users"



Adjust the call field to your LDAP settings and save the changes, e.g.:

```
LDAP IMPORT LDAP_URL=ldap://ldapserver:389/

LDAP_ADMIN_PASS=adminPassword LDAP_SEARCH_KEY=userprincipalname

LDAP_SEARCH_DC="dc=company,dc=com"
```

- Execute the previously adjusted "LDAP Import Users" server action to import the LDAP users in *boom*. Locate the imported LDAP users by opening the **User Management** view and expanding the LDAP user role folder. All new users are **disabled**.
- Configure the imported LDAP users by selectively moving them an appropriate user role and **enabling** the accounts. The LDAP user role should be handled just as a transition group for the LDAP import step and not get any access rights in production environments.

boom LDAP configuration file(s) location:

```
<boom_server_installdir>/srv/ldap/ldap.conf

<boom_server_installdir>/srv/ldap/ldap1.conf

<boom_server_installdir>/srv/ldap/ldap2.conf

<boom_server_installdir>/srv/ldap/...
```

LDAP configuration parameters:

Parameter	Default	Description
LDAP_URL	-	url of LDAP server, i.e. ldap://ldapserver:389
LDAP_BASE	-	LDAP base dc's, i.e. dc=company,dc=com or cn=USERS,dc=company,dc=com
ENABLED	-	Valid values are: true
false. Enable or disable LDAP authentication.	LDAP_USER_ATTR	-
Attribute in LDAP that contains username, i.e. uid (for linux).	LDAP_AUTHENTICATION	simple
Authentication type.	LDAP_FACTORY	com.sun.jndi.ldap.LdapCtxFactory
java LDAP Factory class.	LDAP_EXPIRE	-

6.21.8. Synchronize Users Job

Description

The server job "Synchronize Users" is responsible for regular user synchronization of Active Directory Users with *boom* users.
The Active Directory data is the "master" data, so the job will ensure that the list of *boom* users is same match-ing the Active Directory information.

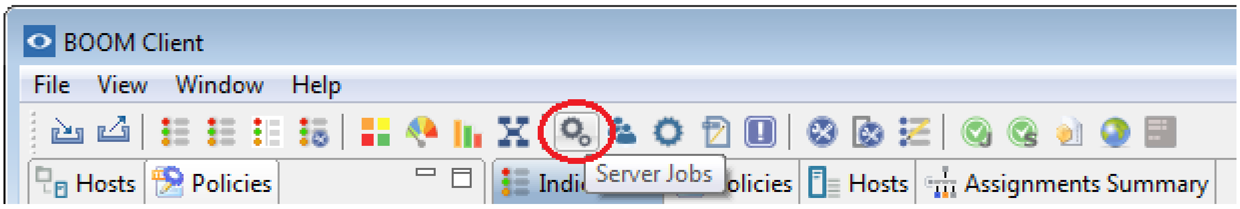
In the USER-GROUP-MAPPING configuration section of the job the mapping between Active Directory user groups and *boom* user roles is configured. Only Active Directory user groups configured in this section are synchronized to boom. The synchronization of multiple user groups is possible.

Important Notes:

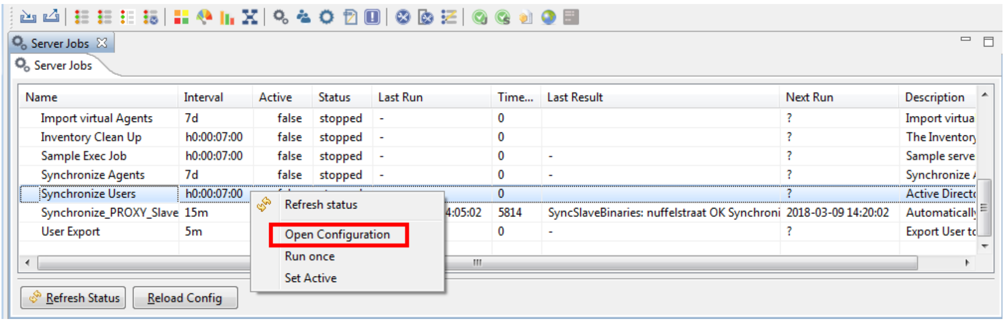
- This job will not change any local *boom* users that might be present in the target *boom* user role unless there exists a local user and Active Directory user with the same name.
- If the specified *boom* user group does not exist, the job will automatically create it.
- A user in *boom* can only be member of one user role. Therefore if an Active Directory user is member of multiple user groups that are configured to be synchronized only the first found occurrence will be taken onto account.
- Active Directory users will be automatically created or removed from *boom* or their *boom* user role membership changed according to the information retrieved from the Active Directory information.
- Do not use the *boom* Tools for LDAP (i.e. LDAP Import/Reload) if you want to use the Synchronization Job. If there are changes that need to be synchronized at once, use the "Run once" Feature of the job as described later.

Configuration

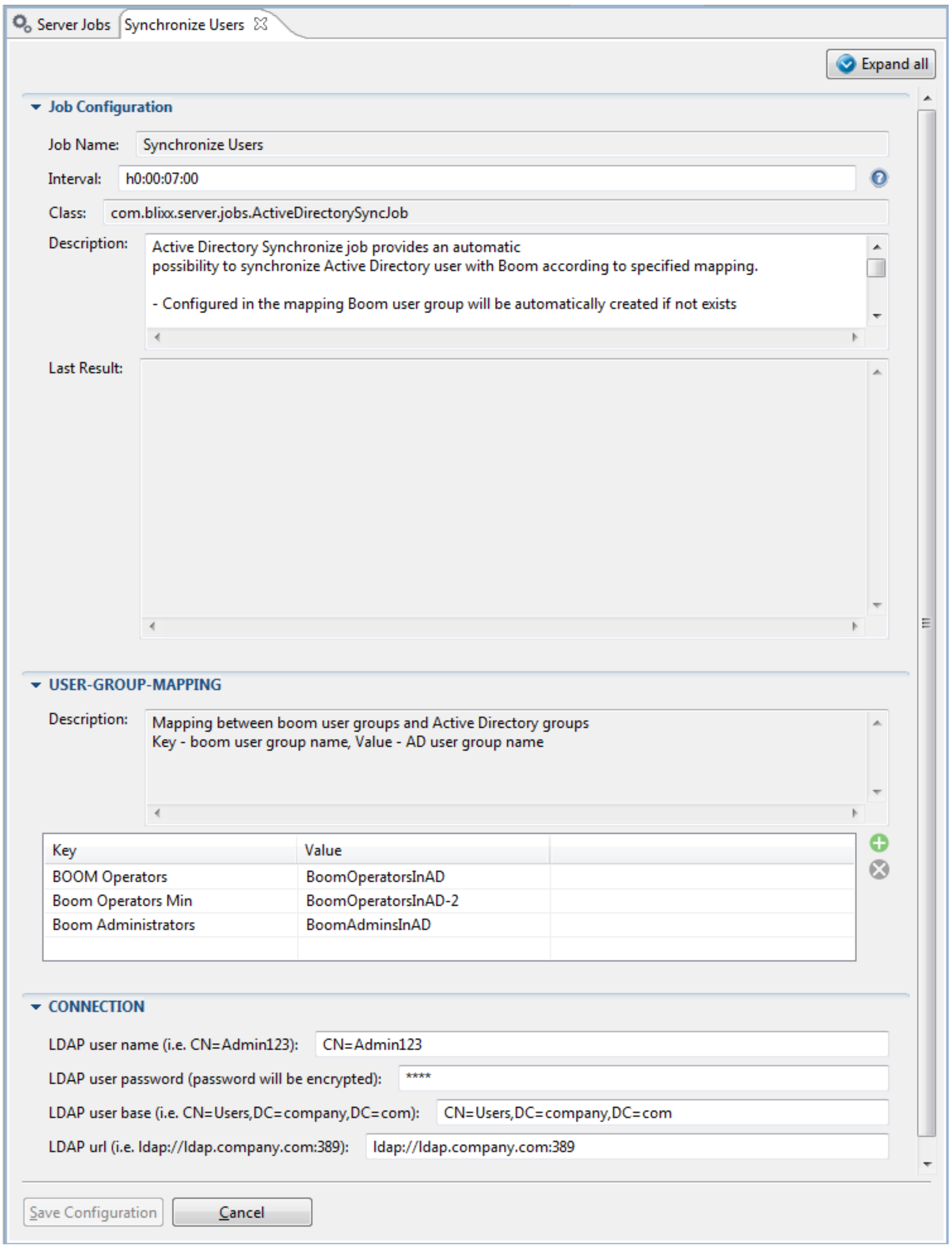
Open "Server Jobs" in the *boom* Client.



Then right click on the "Synchronize Users" server job and click on "Open Configuration"



Name	Interval	Active	Status	Last Run	Time...	Last Result	Next Run	Description
Import virtual Agents	7d	false	stopped	-	0		?	Import virtua
Inventory Clean Up	h0:00:07:00	false	stopped	-	0		?	The Inventory
Sample Exec Job	h0:00:07:00	false	stopped	-	0	-	?	Sample serve
Synchronize Agents	7d	false	stopped	-	0	-	?	Synchronize /
Synchronize Users	h0:00:07:00				0		?	Active Directo
Synchronize_PROXY_Slave	15m			4:05:02	5814	SyncSlaveBinaries: nuffelstraat OK Synchroni	2018-03-09 14:20:02	Automatically
User Export	5m				0	-	?	Export User to



Job Configuration

Job Name: Synchronize Users

Interval: h0:00:07:00

Class: com.blixx.server.jobs.ActiveDirectorySyncJob

Description: Active Directory Synchronize job provides an automatic
possibility to synchronize Active Directory user with Boom according to specified mapping.

- Configured in the mapping Boom user group will be automatically created if not exists

Last Result:

USER-GROUP-MAPPING

Description: Mapping between boom user groups and Active Directory groups
Key - boom user group name, Value - AD user group name

Key	Value
BOOM Operators	BoomOperatorsInAD
Boom Operators Min	BoomOperatorsInAD-2
Boom Administrators	BoomAdminsInAD

CONNECTION

LDAP user name (i.e. CN=Admin123): CN=Admin123

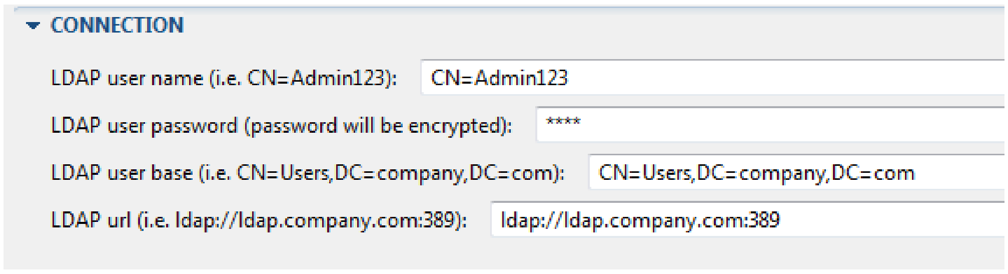
LDAP user password (password will be encrypted): ****

LDAP user base (i.e. CN=Users,DC=company,DC=com): CN=Users,DC=company,DC=com

LDAP url (i.e. ldap://ldap.company.com:389): ldap://ldap.company.com:389

Connection

Expand the Connection section and add the connection properties for the Active Directory Service:



LDAP user name (i.e. CN=Admin123):	CN=Admin123
LDAP user password (password will be encrypted):	****
LDAP user base (i.e. CN=Users,DC=company,DC=com):	CN=Users,DC=company,DC=com
LDAP url (i.e. ldap://ldap.company.com:389):	ldap://ldap.company.com:389

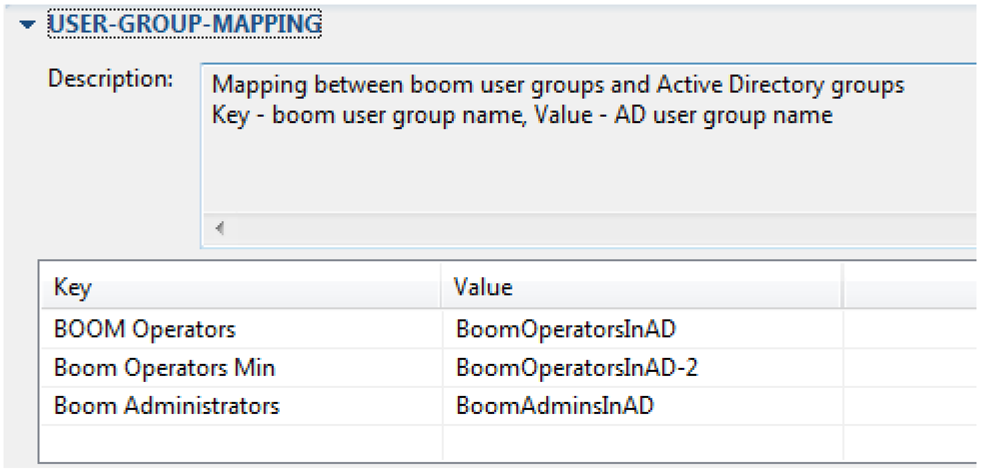
Notes:

- LDAP user must have according rights to list user groups and users.
- URL : for secure connection use: **ldaps://hostname:port**

User-Group-Mapping

You have to define the mapping between *boom* User Role name and Active Directory User Group name.

Key:	Name of the *boom* User Role
Value:	Name of the Active Directory User Group



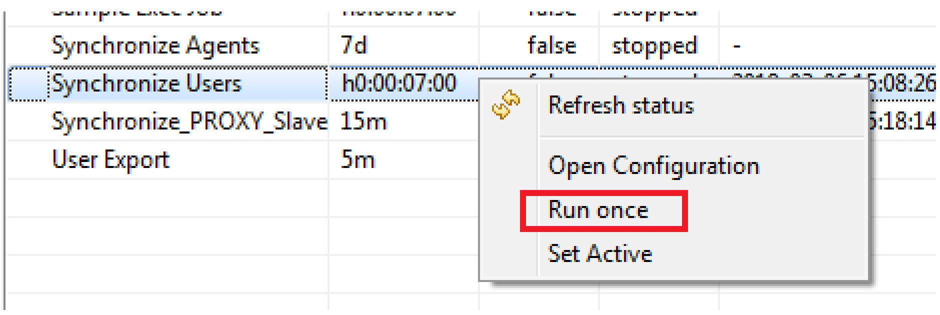
Description: Mapping between boom user groups and Active Directory groups
Key - boom user group name, Value - AD user group name

Key	Value
BOOM Operators	BoomOperatorsInAD
Boom Operators Min	BoomOperatorsInAD-2
Boom Administrators	BoomAdminsInAD

Usage Notes

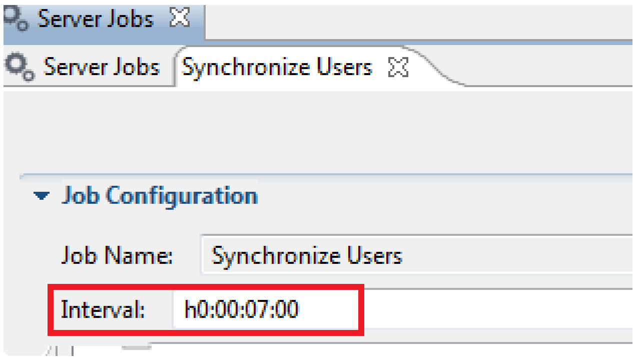
Run once

Then right click "Synchronize Users" server job and click on "Run Once"



Scheduling

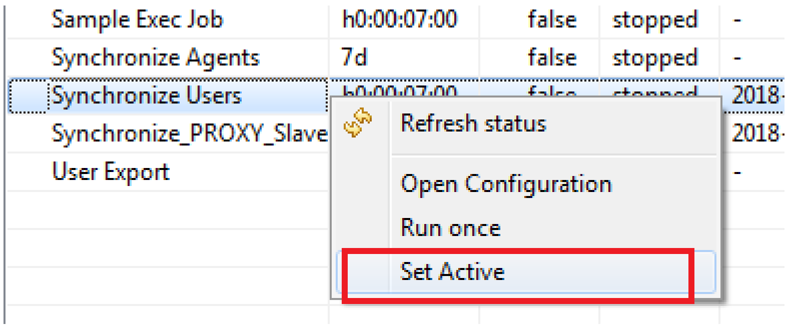
Double click on "Synchronize Users" server job it will open configuration tab, under "Job Configuration" section you will see Interval field:



Job Name: Synchronize Users

Interval: h0:00:07:00

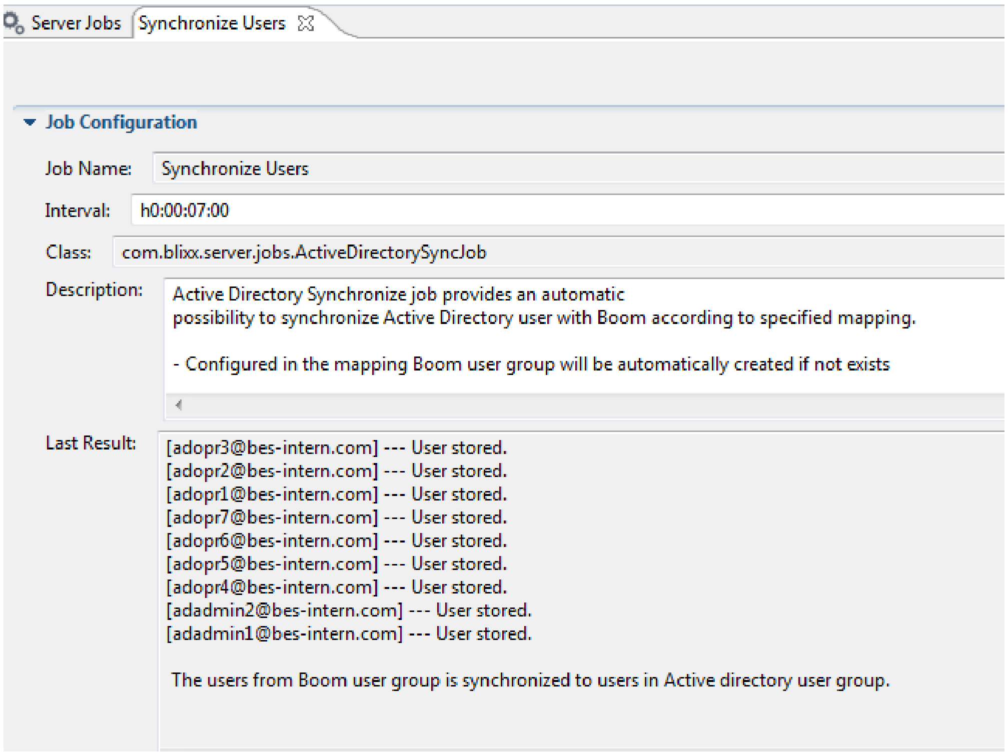
After setting the interval time, switch back to server job tab, right click on the name of the job and click on "Set Active".



Sample Exec Job	h0:00:07:00	false	stopped	-
Synchronize Agents	7d	false	stopped	-
Synchronize Users	h0:00:07:00	false	stopped	2018-
Synchronize_PROXY_Slave				2018-
User Export				-

Final Result Summary

The results of the synchronization job are displayed in the section of the Last Result.



Job Name: Synchronize Users

Interval: h0:00:07:00

Class: com.blixx.server.jobs.ActiveDirectorySyncJob

Description: Active Directory Synchronize job provides an automatic possibility to synchronize Active Directory user with Boom according to specified mapping.

- Configured in the mapping Boom user group will be automatically created if not exists

Last Result:

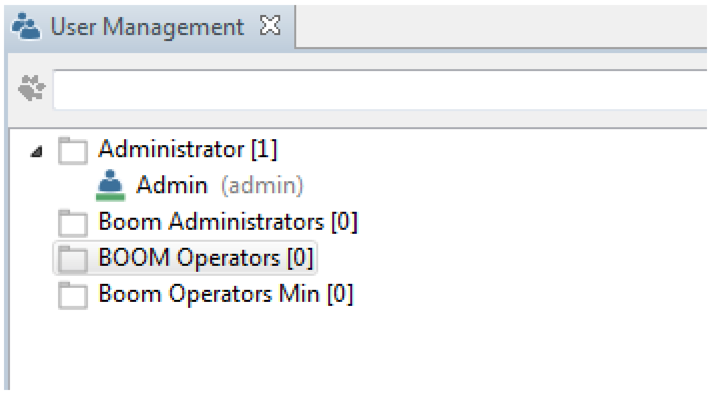
```
[adopr3@bes-intern.com] --- User stored.
[adopr2@bes-intern.com] --- User stored.
[adopr1@bes-intern.com] --- User stored.
[adopr7@bes-intern.com] --- User stored.
[adopr6@bes-intern.com] --- User stored.
[adopr5@bes-intern.com] --- User stored.
[adopr4@bes-intern.com] --- User stored.
[adadmin2@bes-intern.com] --- User stored.
[adadmin1@bes-intern.com] --- User stored.

 The users from Boom user group is synchronized to users in Active directory user group.
```

Scenarios:

Fresh Synchronization

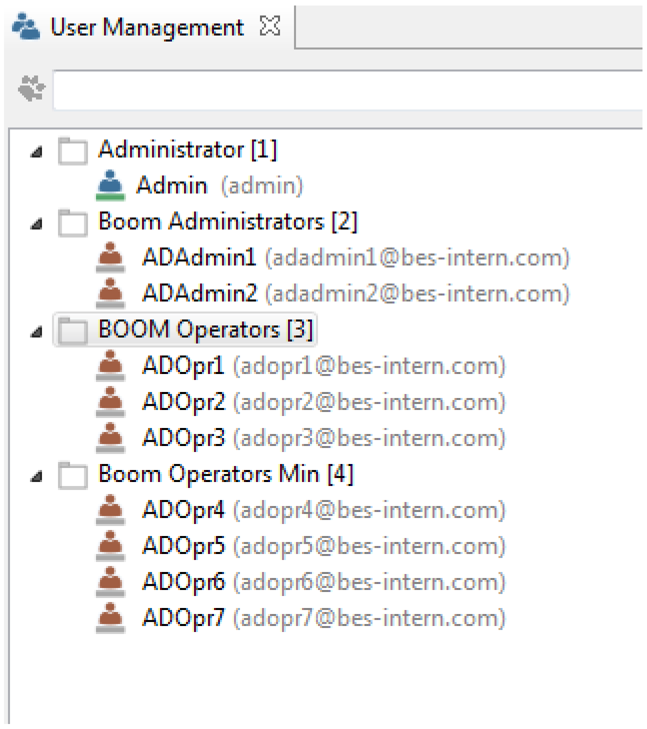
All the non-existing User groups will be created automatically.

Before running the job


User Management

Administrator [1]
 Admin (admin)
Boom Administrators [0]
BOOM Operators [0]
Boom Operators Min [0]

After running the job

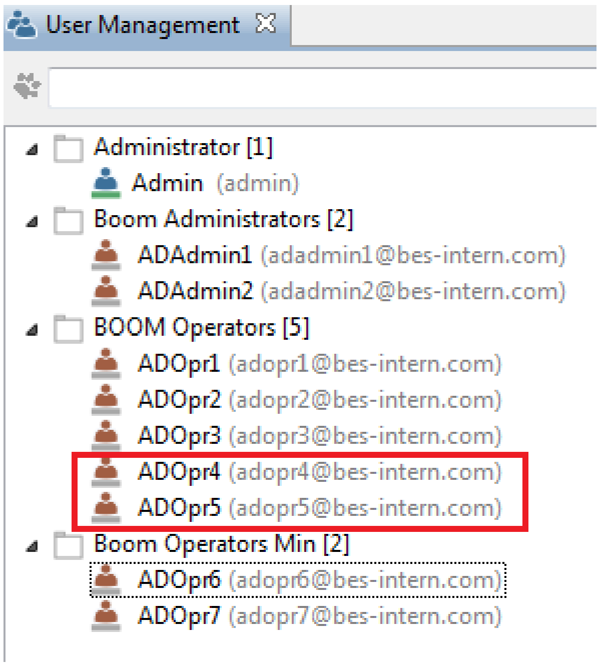

User Management

Administrator [1]
 Admin (admin)
Boom Administrators [2]
 ADAdmin1 (adadmin1@bes-intern.com)
 ADAdmin2 (adadmin2@bes-intern.com)
BOOM Operators [3]
 ADOpr1 (adopr1@bes-intern.com)
 ADOpr2 (adopr2@bes-intern.com)
 ADOpr3 (adopr3@bes-intern.com)
Boom Operators Min [4]
 ADOpr4 (adopr4@bes-intern.com)
 ADOpr5 (adopr5@bes-intern.com)
 ADOpr6 (adopr6@bes-intern.com)
 ADOpr7 (adopr7@bes-intern.com)

Users are moved in AD from one to another group

If **ADOpr4** and **ADOpr5** users were moved in ActiveDirectory from **BoomOperatorsInAD** to **BoomOperatorsInAD-2** group, they also will be moved to the according *boom* User Roles.

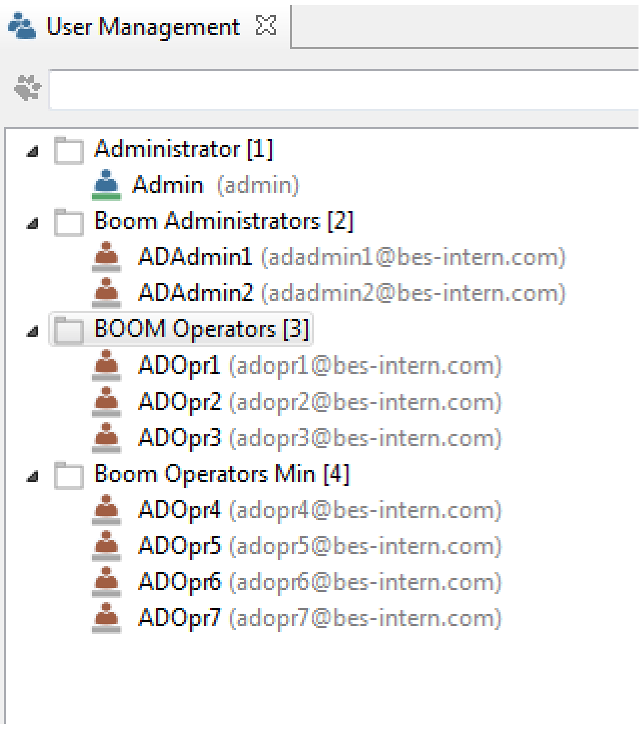
Before running the job



User Management

- Administrator [1]
  - Admin (admin)
- Boom Administrators [2]
  - ADAdmin1 (adadmin1@bes-intern.com)
  - ADAdmin2 (adadmin2@bes-intern.com)
- BOOM Operators [5]
  - ADOpr1 (adopr1@bes-intern.com)
  - ADOpr2 (adopr2@bes-intern.com)
  - ADOpr3 (adopr3@bes-intern.com)
  - ADOpr4 (adopr4@bes-intern.com)
  - ADOpr5 (adopr5@bes-intern.com)
- Boom Operators Min [2]
  - ADOpr6 (adopr6@bes-intern.com)
  - ADOpr7 (adopr7@bes-intern.com)

After running the job



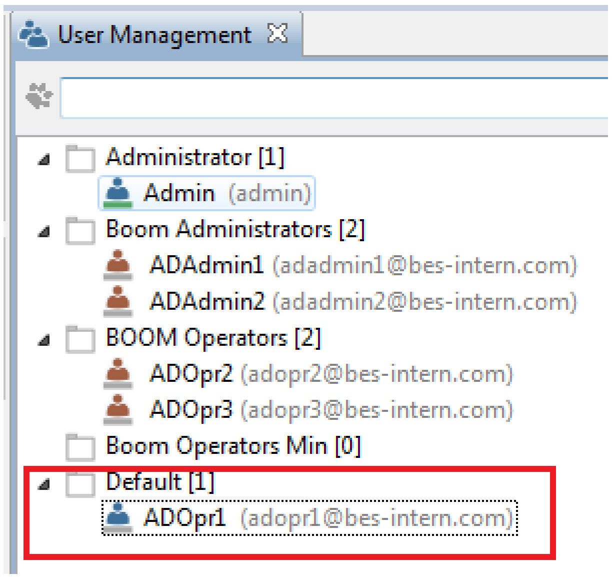
User Management

- Administrator [1]
  - Admin (admin)
- Boom Administrators [2]
  - ADAdmin1 (adadmin1@bes-intern.com)
  - ADAdmin2 (adadmin2@bes-intern.com)
- BOOM Operators [3]
  - ADOpr1 (adopr1@bes-intern.com)
  - ADOpr2 (adopr2@bes-intern.com)
  - ADOpr3 (adopr3@bes-intern.com)
- Boom Operators Min [4]
  - ADOpr4 (adopr4@bes-intern.com)
  - ADOpr5 (adopr5@bes-intern.com)
  - ADOpr6 (adopr6@bes-intern.com)
  - ADOpr7 (adopr7@bes-intern.com)

AD user conflicts with a local user

If conflicting user: **ADOpr1(adopr1@bes-intern.com)** will be detected during the synchronization - the priority will be given to the Active Directory user and the local user will be replaced by the incoming AD User.

Before running the job


User Management

Administrator [1]
- Admin (admin)

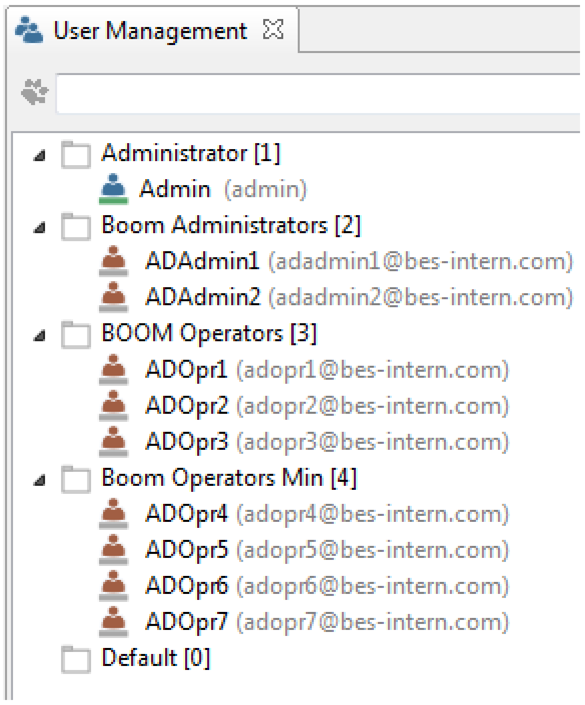
Boom Administrators [2]
- ADAdmin1 (adadmin1@bes-intern.com)
- ADAdmin2 (adadmin2@bes-intern.com)

BOOM Operators [2]
- ADOpr2 (adopr2@bes-intern.com)
- ADOpr3 (adopr3@bes-intern.com)

Boom Operators Min [0]

Default [1]
- ADOpr1 (adopr1@bes-intern.com)

Change In AD


User Management

Administrator [1]
- Admin (admin)

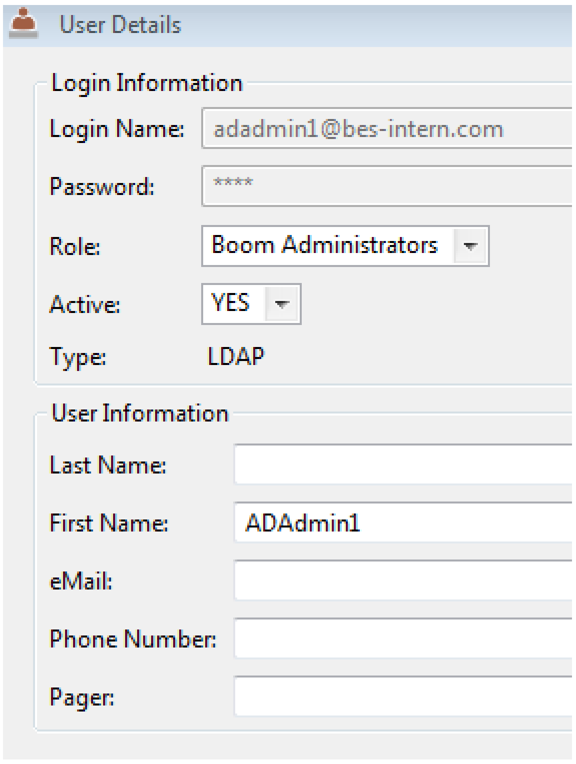
Boom Administrators [2]
- ADAdmin1 (adadmin1@bes-intern.com)
- ADAdmin2 (adadmin2@bes-intern.com)

BOOM Operators [3]
- ADOpr1 (adopr1@bes-intern.com)
- ADOpr2 (adopr2@bes-intern.com)
- ADOpr3 (adopr3@bes-intern.com)

Boom Operators Min [4]
- ADOpr4 (adopr4@bes-intern.com)
- ADOpr5 (adopr5@bes-intern.com)
- ADOpr6 (adopr6@bes-intern.com)
- ADOpr7 (adopr7@bes-intern.com)

Default [0]

After running the job


User Details

Login Information

Login Name:	adadmin1@bes-intern.com
Password:	****
Role:	Boom Administrators
Active:	YES
Type:	LDAP

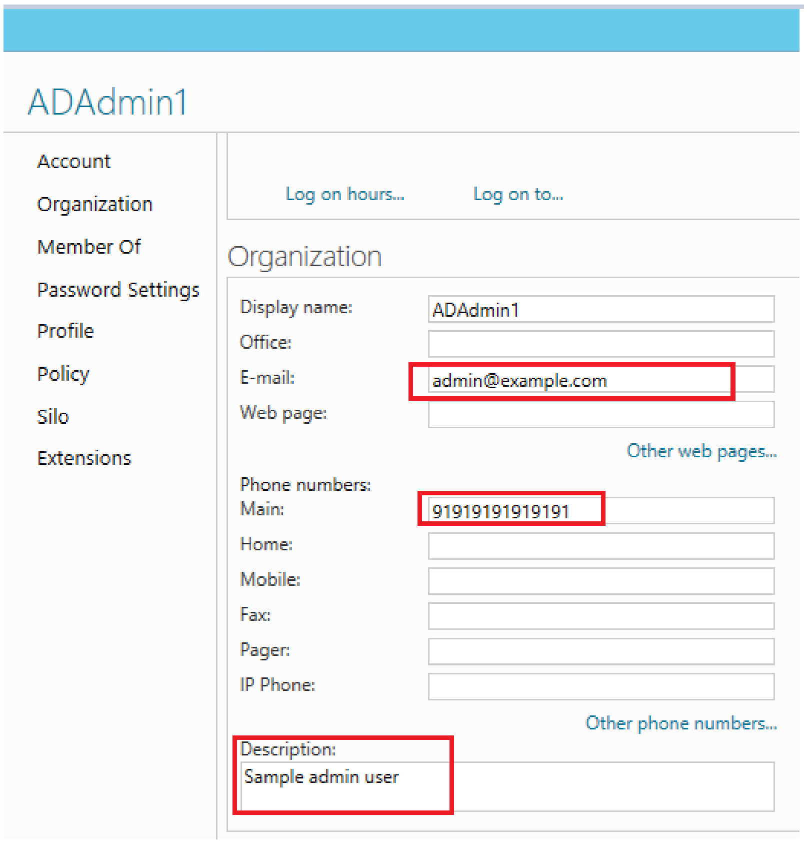
User Information

Last Name:	
First Name:	ADAdmin1
eMail:	
Phone Number:	
Pager:	

Attribute changes

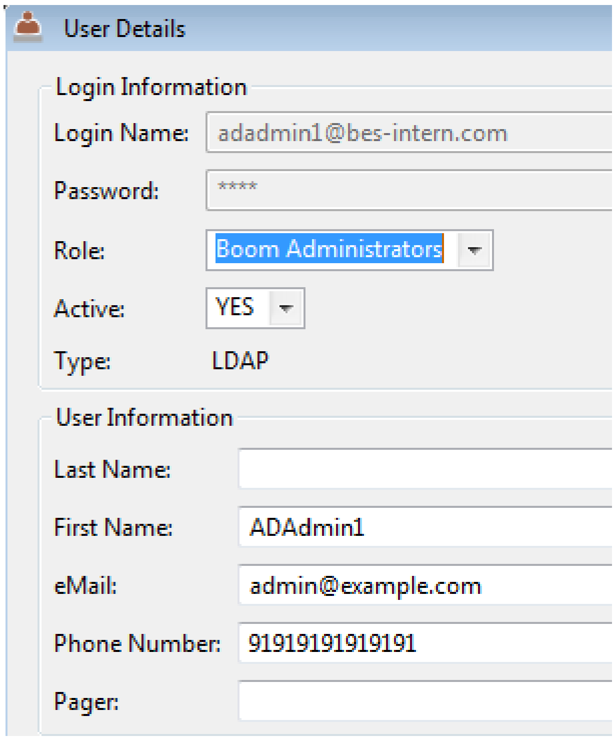
Any changes on attributes: first name, last name, phone, email in Active Directory will be replicated to the *boom* user.
For example, if the user **ADAdmin1** has changes in email, phone, and description then this data is updated for **ADAdmin1** in boom.

Before running the job



Display name:	ADAdmin1
Office:	
E-mail:	admin@example.com
Web page:	
Phone numbers: Main:	91919191919191
Home:	
Mobile:	
Fax:	
Pager:	
IP Phone:	
Description:	Sample admin user

After running the job



User Details

Login Information

Login Name: adadmin1@bes-intern.com

Password: ****

Role: Boom Administrators

Active: YES

Type: LDAP

User Information

Last Name:

First Name: ADAdmin1

eMail: admin@example.com

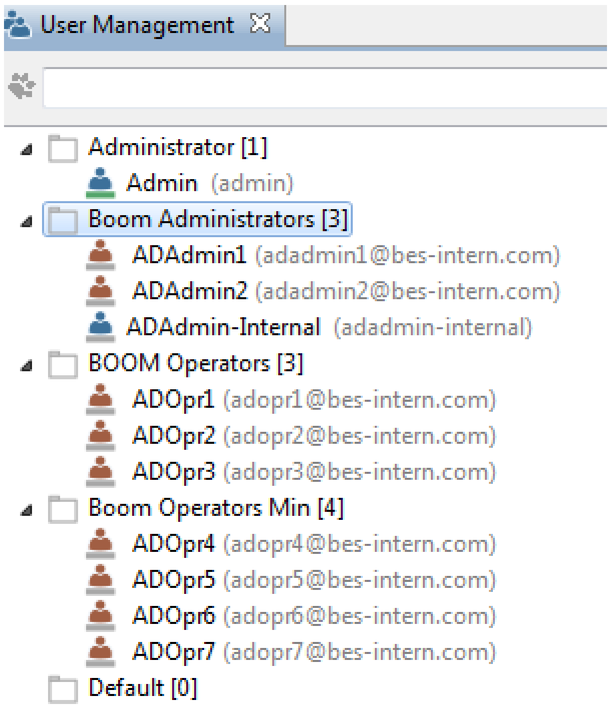
Phone Number: 91919191919191

Pager:

Deleted Active Directory User Group

If the Active Directory User Group was deleted or renamed in Active Directory - the job will clean up the according *boom* user role and remove all previously synced Active Directory users. The *boom* User Role will be not deleted and local *boom* users are not affected.
For example the group **BoomAdminsInAD** Active Directory User Group was deleted:

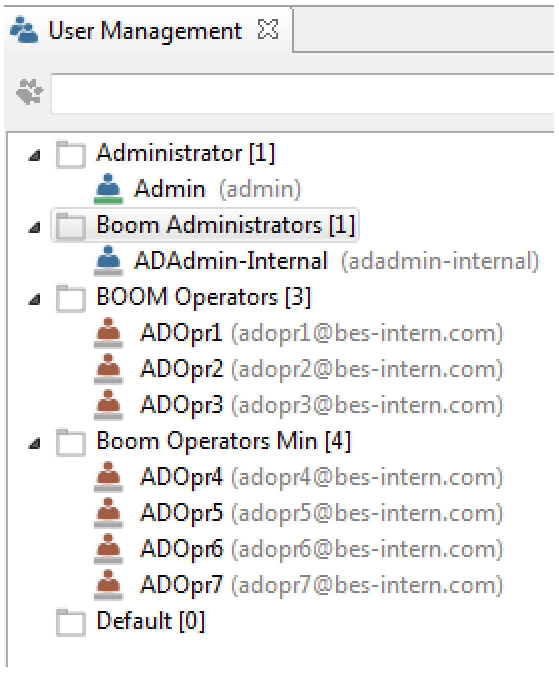
Before running the job



User Management

- Administrator [1]
  - Admin (admin)
- Boom Administrators [3]
  - ADAdmin1 (adadmin1@bes-intern.com)
  - ADAdmin2 (adadmin2@bes-intern.com)
  - ADAdmin-Internal (adadmin-internal)
- BOOM Operators [3]
  - ADOpr1 (adopr1@bes-intern.com)
  - ADOpr2 (adopr2@bes-intern.com)
  - ADOpr3 (adopr3@bes-intern.com)
- Boom Operators Min [4]
  - ADOpr4 (adopr4@bes-intern.com)
  - ADOpr5 (adopr5@bes-intern.com)
  - ADOpr6 (adopr6@bes-intern.com)
  - ADOpr7 (adopr7@bes-intern.com)
- Default [0]

After running the job



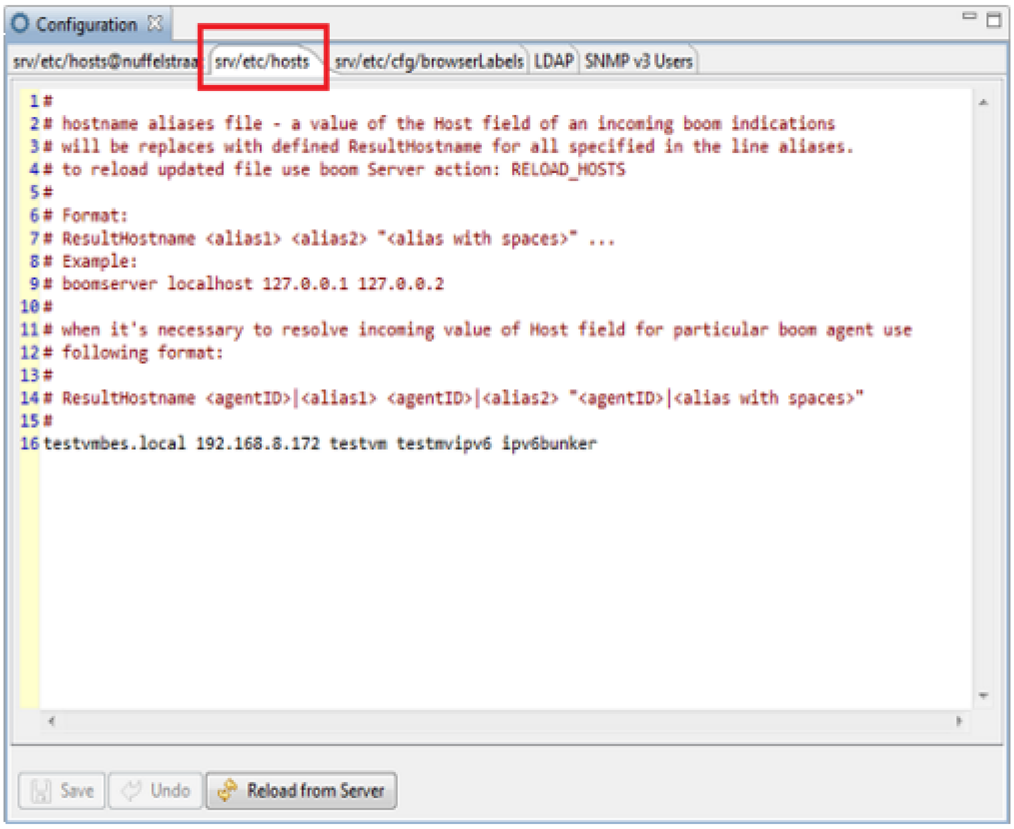
```
Administrator [1]
    Admin  (admin)
Boom Administrators [1]
    ADAdmin-Internal  (adadmin-internal)
BOOM Operators [3]
    ADOpr1 (adopr1@bes-intern.com)
    ADOpr2 (adopr2@bes-intern.com)
    ADOpr3 (adopr3@bes-intern.com)
Boom Operators Min [4]
    ADOpr4 (adopr4@bes-intern.com)
    ADOpr5 (adopr5@bes-intern.com)
    ADOpr6 (adopr6@bes-intern.com)
    ADOpr7 (adopr7@bes-intern.com)
Default [0]
```

6.22. Configuration View

The configuration view allows to configure different aspects of the server behavior

- Alias mapping configuration
- relabelling of the indication browser columns
- LDAP Server integrations
- SNMP v3 Authentication Information

Hosts Tab



```
#
# hostname aliases file - a value of the Host field of an incoming boom indications
# will be replaces with defined ResultHostname for all specified in the line aliases.
# to reload updated file use boom Server action: RELOAD_HOSTS
#
# Format:
# ResultHostname <alias1> <alias2> "<alias with spaces>" ...
# Example:
# boomserver localhost 127.0.0.1 127.0.0.2
#
# when it's necessary to resolve incoming value of Host field for particular boom agent use
# following format:
#
# ResultHostname <agentID>|<alias1> <agentID>|<alias2> "<agentID>|<alias with spaces>"
#
testvmbes.local 192.168.8.172 testvm testmvipv6 ipv6bunker
```

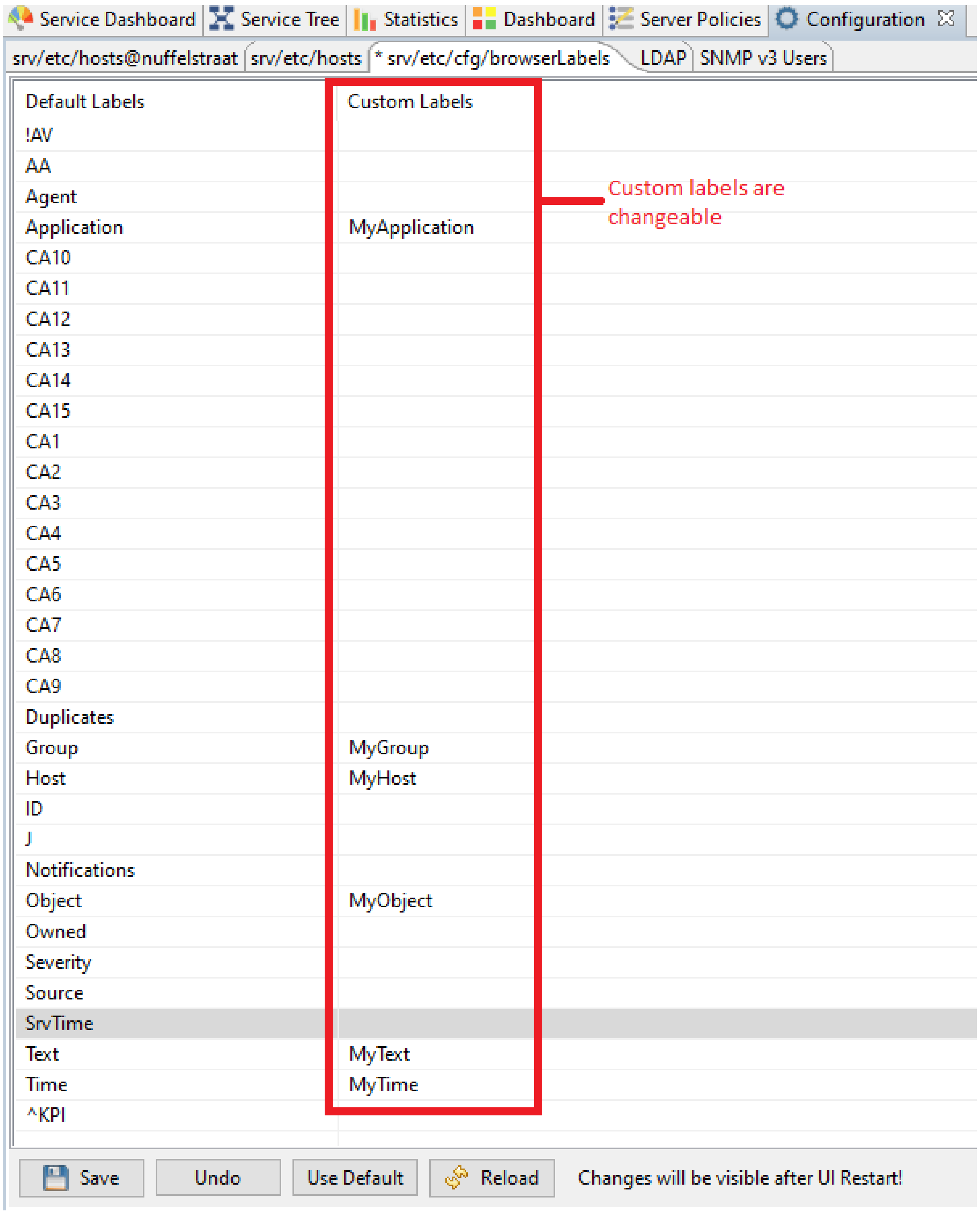
This configuration view gives the possibility to edit/store the host name alias file **<boom_server_dir>/srv/etc/hosts** for the master server and all connected slave servers. The hostname alias files are stored on the servers at the location **<installation directory>/srv/etc/hosts**. In some environments a single host might be known under multiple names, ip addresses, mixtures of short/long as well as capital and lower case letter names. This happens especially if the host has specific addresses or names e.g. for it’s web service or web applications, multiple network cards or is monitored from local and remote monitors. To make sure that all indications are mapped to the correct host, the hosts file can be used to configure alias mappings that could not be automatically resolved.

The *boom* server will check the host field of incoming indications. If a value matches an alias that is configured in the hosts file, the hostname will be replaced with the defined "Result Hostname".

If multiple entries for one alias are found, the Result Hostname of the last entry will be used.

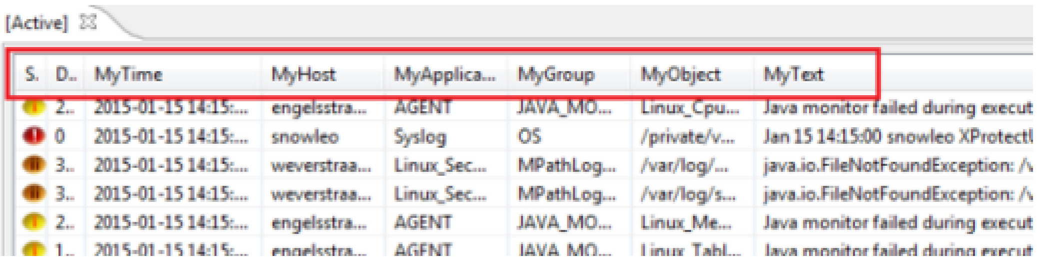
Browser Labels Tab

An user with administrative rights can change the column labels that are displayed in the indication browser. This is especially interesting if e.g. the Custom Attributes are used and the column name should reflect the meaning of the CA.



Default Labels	Custom Labels
!AV	
AA	
Agent	
Application	MyApplication
CA10	
CA11	
CA12	
CA13	
CA14	
CA15	
CA1	
CA2	
CA3	
CA4	
CA5	
CA6	
CA7	
CA8	
CA9	
Duplicates	
Group	MyGroup
Host	MyHost
ID	
J	
Notifications	
Object	MyObject
Owned	
Severity	
Source	
SrvTime	
Text	MyText
Time	MyTime
^KPI	

Changes will be visible after UI Restart!



S.	D..	MyTime	MyHost	MyApplica...	MyGroup	MyObject	MyText
●	2..	2015-01-15 14:15:...	engelsstra...	AGENT	JAVA_MO...	Linux_Cpu...	Java monitor failed during execut
●	0	2015-01-15 14:15:...	snowleo	Syslog	OS	/private/v...	Jan 15 14:15:00 snowleo XProtectl
●	3..	2015-01-15 14:15:...	weverstraa...	Linux_Sec...	MPathLog...	/var/log/...	java.io.FileNotFoundException: /\
●	3..	2015-01-15 14:15:...	weverstraa...	Linux_Sec...	MPathLog...	/var/log/s...	java.io.FileNotFoundException: /\
●	2..	2015-01-15 14:15:...	engelsstra...	AGENT	JAVA_MO...	Linux_Me...	Java monitor failed during execut
●	1..	2015-01-15 14:15:...	engelsstra...	AGENT	JAVA_MO...	Linux_Tabl...	Java monitor failed during execut

LDAP Tab

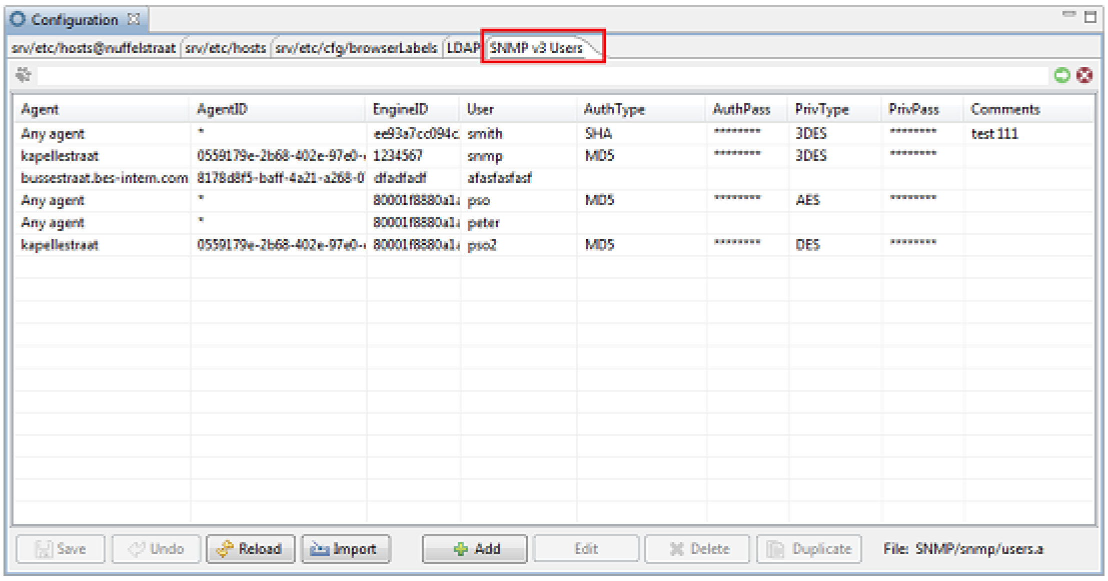
Please refer to the chapter LDAP Authentication for detailed information.

SNMP v3 Users

Please refer to the chapter SNMP v3 User Authentication Management for detailed information.

6.22.1. SNMP v3 User Authentication Management

The tab SNMP v3 Users in the configuration view allows to configure centrally all information that is necessary for the trap handling with SNMPv3 authentication and encryption.



Agent	AgentID	EngineID	User	AuthType	AuthPass	PrivType	PrivPass	Comments
Any agent	*	ee93a7cc094c	smith	SHA	********	3DES	********	test 111
kapellestraat	0559179e-2b68-402e-97e0-	1234567	snmp	MD5	********	3DES	********	
bussestraat.bes-intern.com	8178d8f5-baff-4a21-a268-0	dfadfadf	afasfasfasf					
Any agent	*	80001f8880a1	pso	MD5	********	AES	********	
Any agent	*	80001f8880a1	peter					
kapellestraat	0559179e-2b68-402e-97e0-	80001f8880a1	pso2	MD5	********	DES	********	

File: SNMP/snmp/users.a

Each configuration entry consists of

AgentID and Agent (Label)

The affected Agent that will accept the specified credentials and encryption for receiving traps/informs from the SNMP Device referenced by the EngineID. The Agent label is fetched automatically by using the AgentID.

EngineID

The SNMP Engine ID used by the SNMP Device. This field is always required.

User

User name used for authentication credentials. It is always required, even if the AuthType is set to "none".

AuthType

One of the predefinded SNMPv3 authentication types (none, MD5, SHA,HMAC192SHA256, HMAC384SHA512). None means that the authentication pass-phrase is not used.

AuthPass

Pass-phrase for the authentication which is used to sign messages. It must be at least 8 chars or longer.

PrivType

Specifies the privacy type (none, DES, 3DES, AESxxx) that is used to encrypt messages

PrivPass

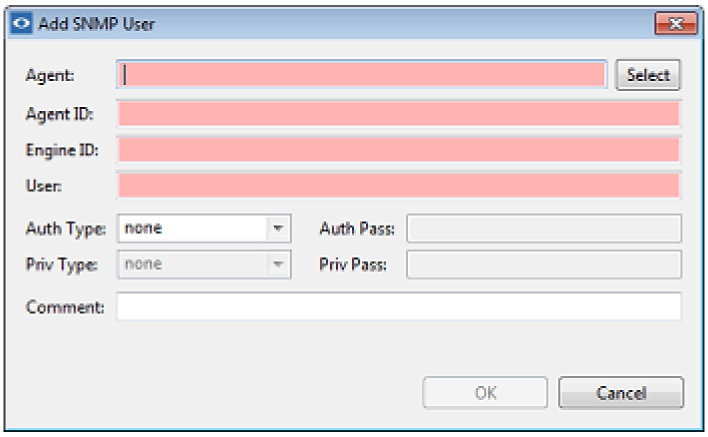
Key that is used for the message encryption. It must be at least 8 chars or longer.

Comments

Comment

Please be aware that a Privacy Type of anything other than "none" requires to also use an Authorization Type other than "none". This means it is not possible to use the AuthType=none with e.g. PrivType=AES256. It is possible to use AuthType=none and PrivType=none which basically switches off the checking except the clear text user name, or any combination of AuthType whith PrivType="none" which switches the encryption off.

Use the **Add** button to add a new entry to the list.



Select the Agent using the **Select** button or "Any Agent" by placing a "*" in the Agent ID Field. The Agent ID is automatically filled with the correct ID if an Agent is selected. Fill the EngineID with the information provided by SNMP and all the Authorization and Privacy settings. If all fields are entered correctly, the **OK** button is active.

You can also use the **Import** button to start the import of the data from a file. The dialog will allow to select the separator character (e.g. ",") and how the fields in the file should be mapped to the configuration fields. After pressing **OK** the new entries will show up in the list.

In order to save the work, please do not forget to press the **Save** button.

Activate the SNMPv3 User configuration

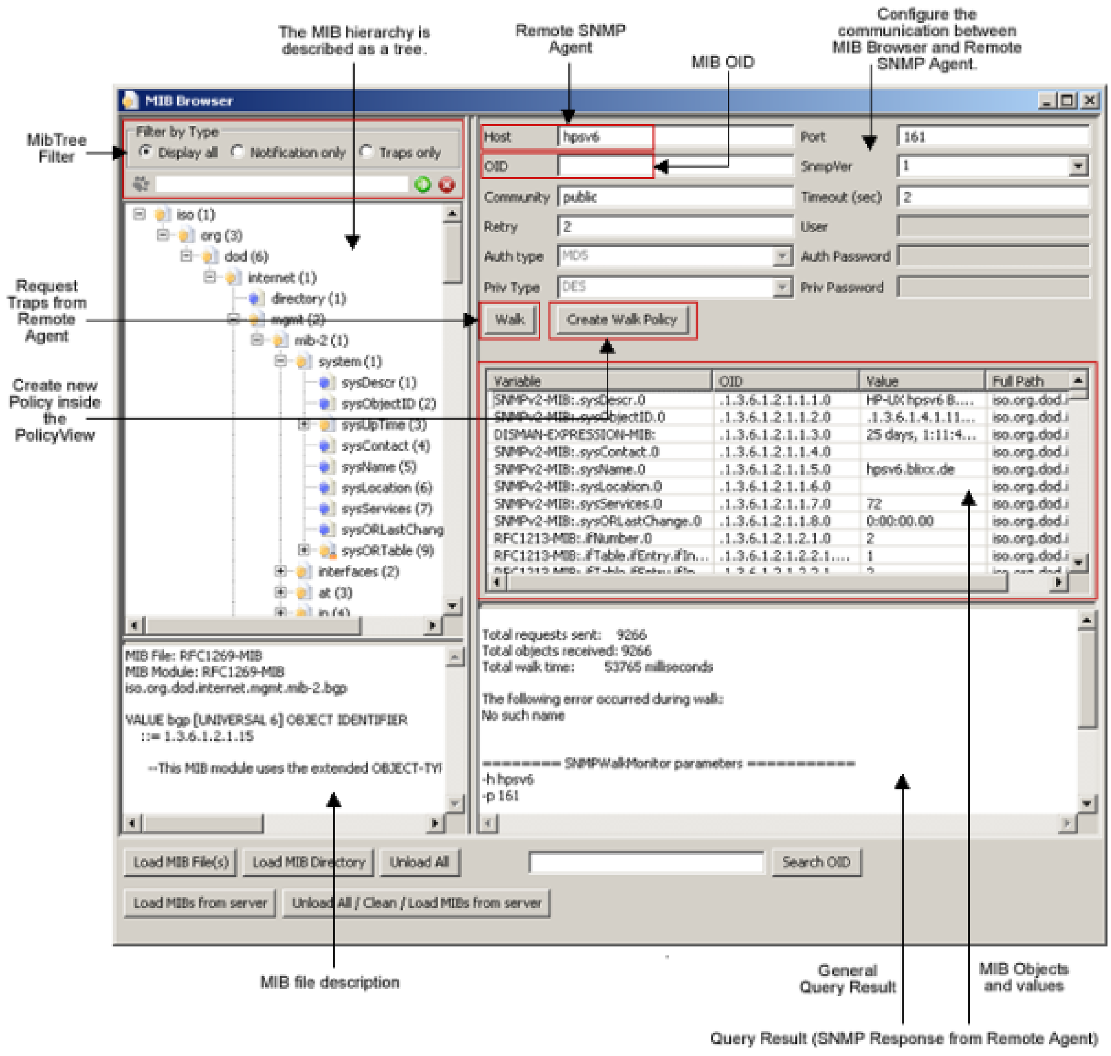
The global SNMPv3 user configuration is kept on the server in the file srv/packages/SNMP/snmp/users.asf. This file is referenced in the SNMP binary package. When this binary package gets deployed to an agent, the server will recognize the ".asf" extension (agent specific file) and process it, so that the resulting file that is deployed to the agent only contains the entries that are either valid for all agents (Agent=*) or valid for this specific Agent (i.e. identified by the AgentID). In *boom* Master/Slave Environments all changes need to be done centrally on the Master-Server.
In order to activate any change in the SNMPv3 User configuration it is sufficient to redeploy the SNMP binary package to the agents which need this update. Typically a "Redeploy to all" will be the easiest way to achieve this since there will be only a limited amount of trap receivers in the environment that get this update. *

6.23. MIB Browser

6.23.1. General Information

A management information base (MIB) is a collection of managed objects residing in a virtual information store. Collections of related managed objects are defined in specific MIB modules.

SNMP uses an extensible design, where the available information is defined by MIBs. The MIB hierarchy can be depicted as a tree with a nameless root. The tree entries are addressed through so called OID’s (object identifiers).



Host	hpsv6	Port	161
OID		SnmpVer	1
Community	public	Timeout (sec)	2
Retry	2	User	
Auth type	MD5	Auth Password	
Priv Type	DES	Priv Password	

Variable	OID	Value	Full Path
SNMPv2-MIB::sysDescr.0	.1.3.6.1.2.1.1.1.0	HP-UX hpsv6 B....	iso.org.dod.i
SNMPv2-MIB::sysObjectID.0	.1.3.6.1.2.1.1.2.0	.1.3.6.1.4.1.11...	iso.org.dod.i
DISMAN-EXPRESSION-MIB:	.1.3.6.1.2.1.1.3.0	25 days, 1:11:4...	iso.org.dod.i
SNMPv2-MIB::sysContact.0	.1.3.6.1.2.1.1.4.0		iso.org.dod.i
SNMPv2-MIB::sysName.0	.1.3.6.1.2.1.1.5.0	hpsv6.blixx.de	iso.org.dod.i
SNMPv2-MIB::sysLocation.0	.1.3.6.1.2.1.1.6.0		iso.org.dod.i
SNMPv2-MIB::sysServices.0	.1.3.6.1.2.1.1.7.0	72	iso.org.dod.i
SNMPv2-MIB::sysORLastChange.0	.1.3.6.1.2.1.1.8.0	0:00:00.00	iso.org.dod.i
RFC1213-MIB::ifNumber.0	.1.3.6.1.2.1.2.1.0	2	iso.org.dod.i
RFC1213-MIB::ifTable.ifEntry.ifIn...	.1.3.6.1.2.1.2.2.1....	1	iso.org.dod.i

```
MIB File: RFC1269-MIB
MIB Module: RFC1269-MIB
iso.org.dod.internet.mgmt.mib-2.bgp

VALUE bgp [UNIVERSAL 6] OBJECT IDENTIFIER
    ::= 1.3.6.1.2.1.15

    --This MIB module uses the extended OBJECT-TY
```

```
Total requests sent:    9266
Total objects received: 9266
Total walk time:        53765 milliseconds

The following error occurred during walk:
No such name


========= SNMPWalkMonitor parameters ============
-h hpsv6
-p 161
```

MIB Types

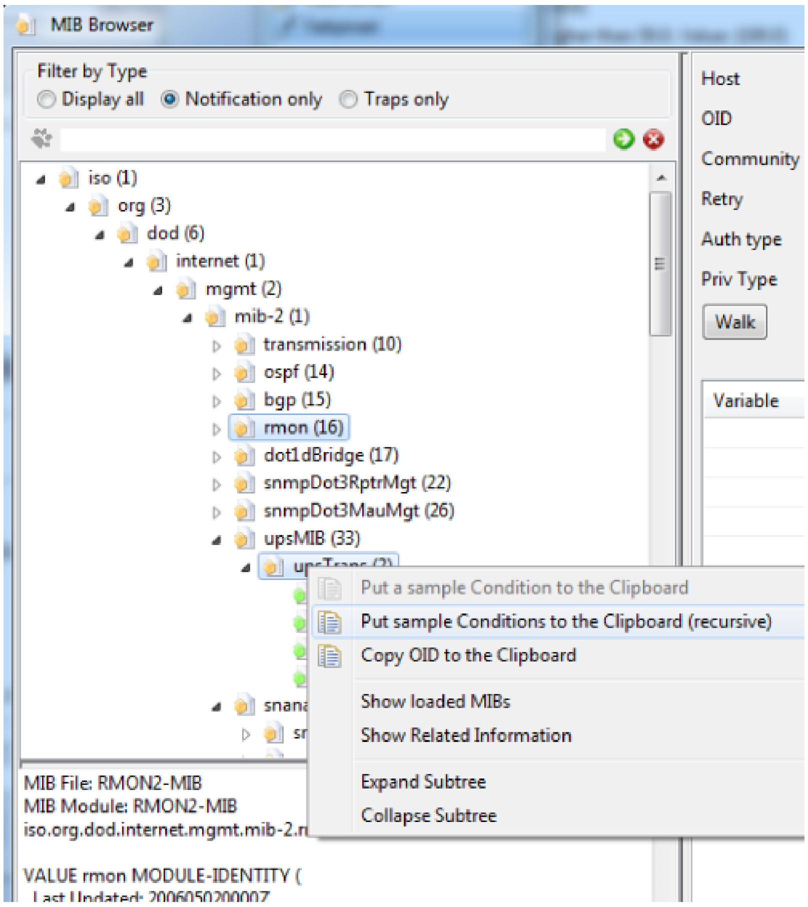
- Folder
- Folder not accessible
- Notification
- Trap
- Primitive
- Primitive not accessible
- Primitive write only

6.23.2. MIB Browser Operations

The MIB browser allows you to

- filter the display: Display all / Notification only (SNMPv2)/ Traps only (SNMPv1)
- load/unload and compile standard MIB files
- query values of SNMP-variables from network devices
- create a Walk Policy
- search OID in MIB Tree

Right-click on a tree element to open the context menu:

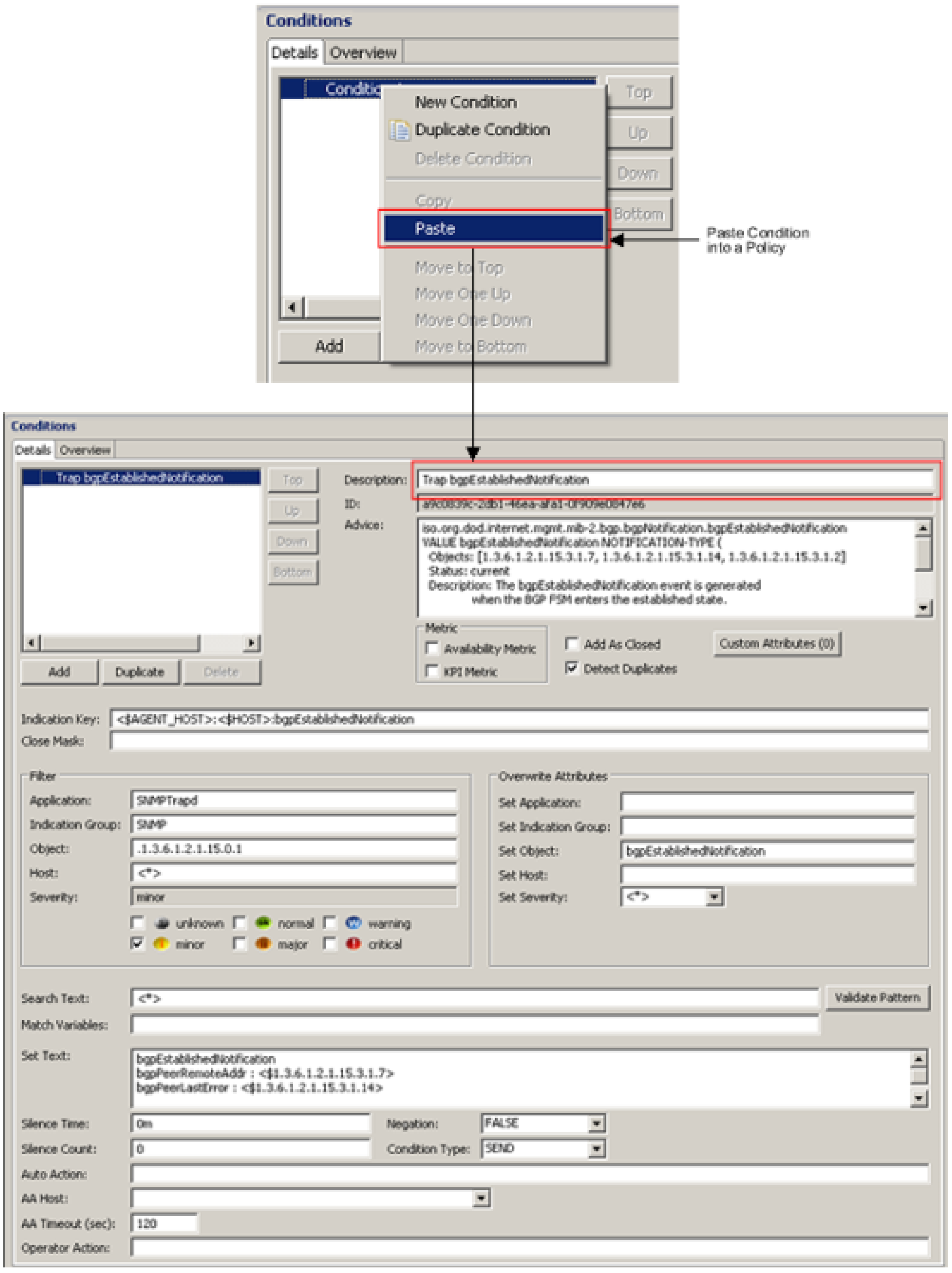


MIB File: RMON2-MIB
MIB Module: RMON2-MIB
iso.org.dod.internet.mgmt.mib-2.r

VALUE rmon MODULE-IDENTITY (
Last Updated: 200605020000Z

Put a sample condition(s) to the clipboard:

This will create a sample trap policy condition and put it into the clipboard. Now you can paste the condition(s) from the clipboard into a policy:



Conditions

Details | Overview

Trap bgpEstablishedNotification

Description: Trap bgpEstablishedNotification

ID: a9c0839c-2db1-46ea-afa1-0f909e0847e6

Advice:

```
iso.org.dod.internet.mgmt.mib-2.bgp.bgpNotification.bgpEstablishedNotification
VALUE bgpEstablishedNotification NOTIFICATION-TYPE (
  Objects: [1.3.6.1.2.1.15.3.1.7, 1.3.6.1.2.1.15.3.1.14, 1.3.6.1.2.1.15.3.1.2]
  Status: current
  Description: The bgpEstablishedNotification event is generated
            when the BGP FSM enters the established state.
```

Metric: ☐ Availability Metric ☐ KPI Metric

☐ Add As Closed

☑ Detect Duplicates

Custom Attributes (0)

Add | Duplicate | Delete

Indication Key: <$AGENT_HOST>:<$HOST>:bgpEstablishedNotification

Close Mask:

Filter

Application:	SNMPTrapd
Indication Group:	SNMP
Object:	.1.3.6.1.2.1.15.0.1
Host:	<*>
Severity:	minor

☐ unknown ☐ normal ☐ warning
☑ minor ☐ major ☐ critical

Overwrite Attributes

Set Application:	
Set Indication Group:	
Set Object:	bgpEstablishedNotification
Set Host:	
Set Severity:	<*>

Search Text: <*>

Validate Pattern

Match Variables:

Set Text:

```
bgpEstablishedNotification
bgpPeerRemoteAddr : <$1.3.6.1.2.1.15.3.1.7>
bgpPeerLastError : <$1.3.6.1.2.1.15.3.1.14>
```

Silence Time:	0m	Negation:	FALSE
Silence Count:	0	Condition Type:	SEND

Auto Action:

AA Host:

AA Timeout (sec): 120

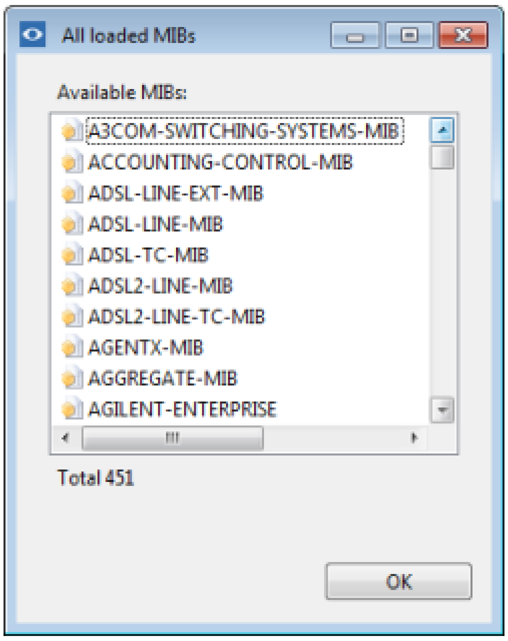
Operator Action:

Copy OID to the Clipboard:

Copies the OID of the selected MIB object into the clipboard.

Show loaded MIBs:

This will list all loaded modules in a separate dialog:

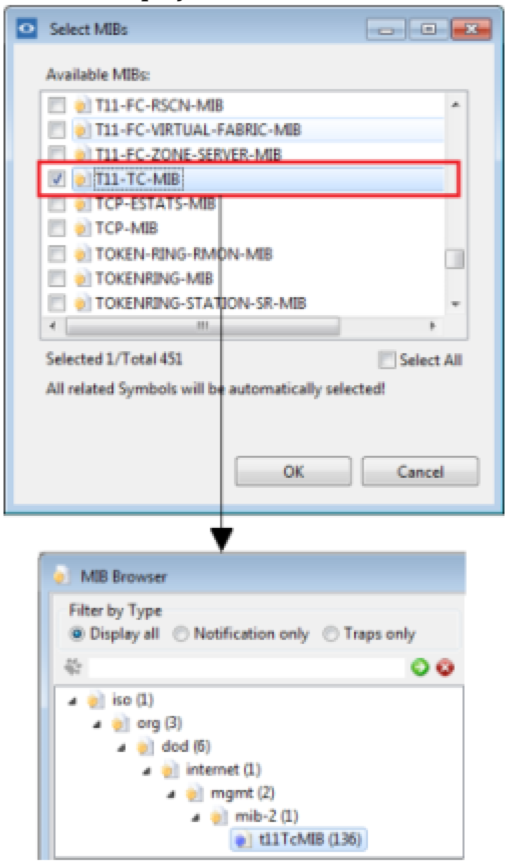


A3COM-SWITCHING-SYSTEMS-MIB
ACCOUNTING-CONTROL-MIB
ADSL-LINE-EXT-MIB
ADSL-LINE-MIB
ADSL-TC-MIB
ADSL2-LINE-MIB
ADSL2-LINE-TC-MIB
AGENTX-MIB
AGGREGATE-MIB
AGILENT-ENTERPRISE

Total 451

Select MIBs:

This will display the MIB tree with the selected MIB as root element



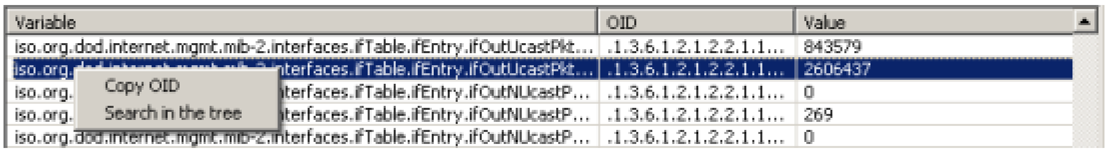
☐ T11-FC-RSCN-MIB
☐ T11-FC-VIRTUAL-FABRIC-MIB
☐ T11-FC-ZONE-SERVER-MIB
☑ T11-TC-MIB
☐ TCP-ESTATS-MIB
☐ TCP-MIB
☐ TOKEN-RING-RMON-MIB
☐ TOKENRING-MIB
☐ TOKENRING-STATION-SR-MIB

Selected 1/Total 451

All related Symbols will be automatically selected!

```
iso (1)
  org (3)
    dod (6)
      internet (1)
        mgmt (2)
          mib-2 (1)
            t11TcMIB (136)
```

Right-click on a table element to open the context menu:



Variable	OID	Value
iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOutUcastPkt...	.1.3.6.1.2.1.2.2.1.1...	843579
iso.org.nterfaces.ifTable.ifEntry.ifOutUcastPkt...	.1.3.6.1.2.1.2.2.1.1...	2606437
iso.org.terfaces.ifTable.ifEntry.ifOutNUcastP...	.1.3.6.1.2.1.2.2.1.1...	0
iso.org.terfaces.ifTable.ifEntry.ifOutNUcastP...	.1.3.6.1.2.1.2.2.1.1...	269
iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifOutNUcastP...	.1.3.6.1.2.1.2.2.1.1...	0

Copy OID:

Copies the OID of the select MIB object into the clipboard.

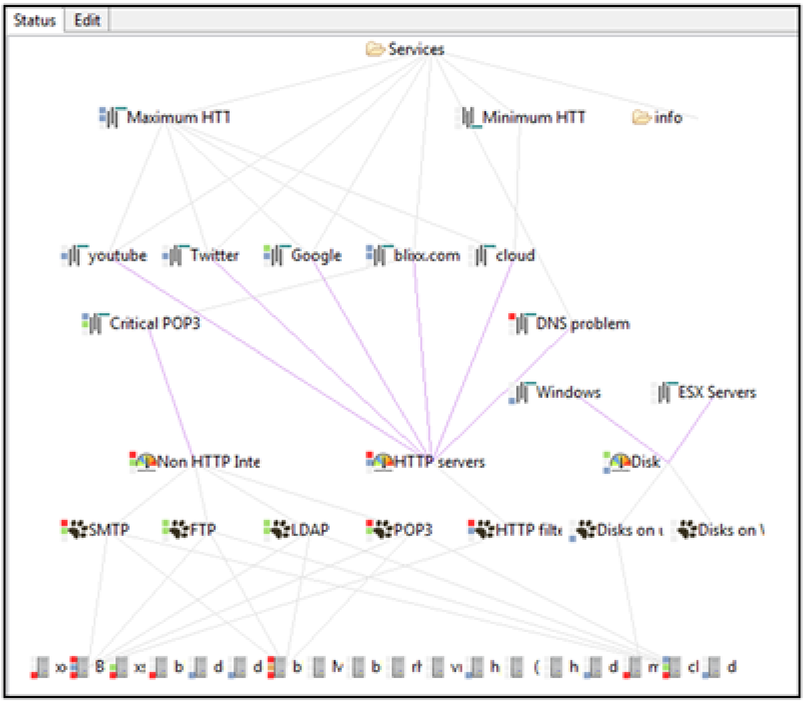
Search in MIB Tree:

This highlights the appropriate MIB in the MIB tree.

6.24. Service Tree

6.24.1. General Information

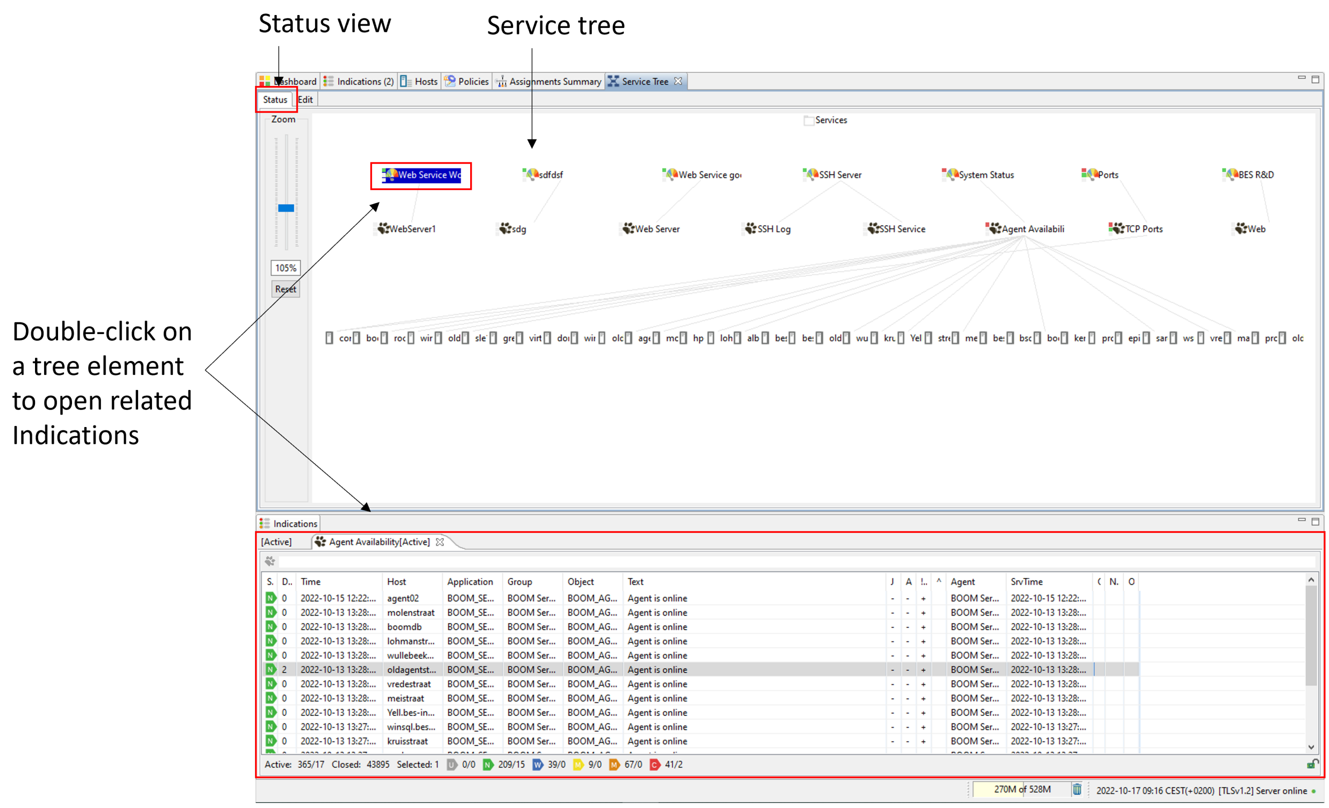
The service tree is a module that allows to build dependency tree on top of configured simple service dashboard elements. All links below the service dashboard elements are creating automatically based in active indications. Every element on top must be configured manually.



Every manually created element can be one of Maximum or Minimum type and can group multiple sub-elements via connections. Every connection in the tree can have defined filter. This allows to split indications grouped by lower level element to multiple elements on the upper levels. A double-click on an element in the tree will open an indication browser with related indications filtered.

6.24.2. Service Tree

Open the **Show Service Tree view** in the *boom* GUI.

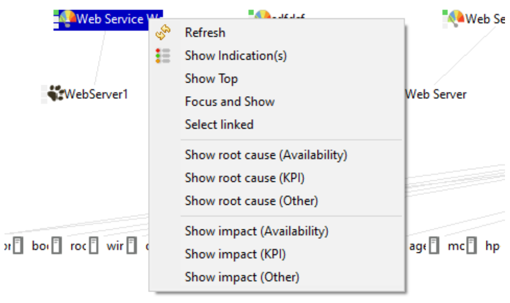


S.	D..	Time	Host	Application	Group	Object	Text	J	A	!..	^	Agent	SrvTime	(	N.	O
N	0	2022-10-15 12:22:...	agent02	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-15 12:22:...			
N	0	2022-10-13 13:28:...	molenstraat	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:28:...			
N	0	2022-10-13 13:28:...	boomdb	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:28:...			
N	0	2022-10-13 13:28:...	lohmanstr...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:28:...			
N	0	2022-10-13 13:28:...	wullebeek...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:28:...			
N	2	2022-10-13 13:28:...	oldagentst...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:28:...			
N	0	2022-10-13 13:28:...	vredestraat	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:28:...			
N	0	2022-10-13 13:28:...	meistraat	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:28:...			
N	0	2022-10-13 13:28:...	Yell.bes-in...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:28:...			
N	0	2022-10-13 13:27:...	winsql.bes...	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:27:...			
N	0	2022-10-13 13:27:...	kruisstraat	BOOM_SE...	BOOM Ser...	BOOM_AG...	Agent is online	-	-	+		BOOM Ser...	2022-10-13 13:27:...			

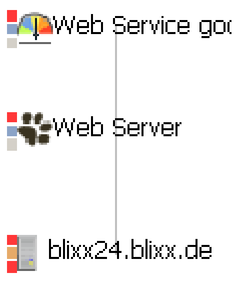
Active: 365/17 Closed: 43895 Selected: 1 U 0/0 N 209/15 W 39/0 M 9/0 M 67/0 C 41/2

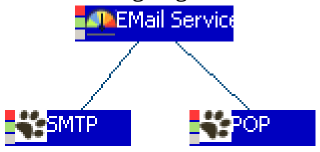
270M of 528M 2022-10-17 09:16 CEST(+0200) [TLSv1.2] Server online

Right-click on an element to open the context menu:



Zoom *+/-:*	Normal zoom in and zoom out of the "Service Tree".
Zoom Layout *+/-:*	Zoom in/out without downsizing the elements.
Refresh:	Refresh the Service Tree.
Focus and Show:	This displays the selected element with all linked item only. All other elements are not shown in this view.

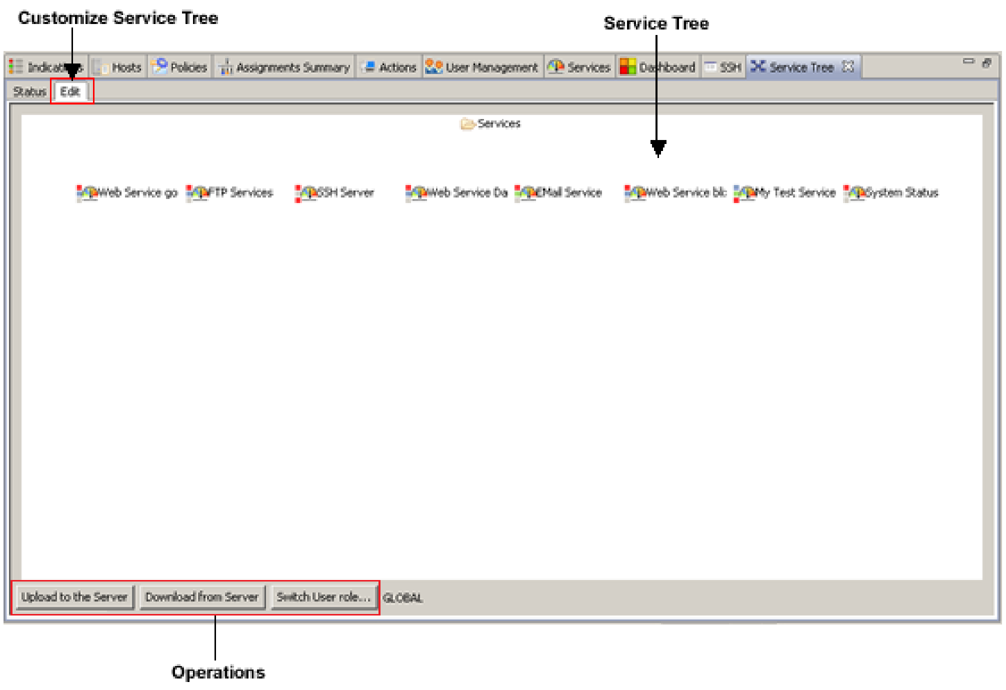


Select Linked:	This will highlight all linked elements. 
Show Root/Show Impact:	This functions will highlight the related elements for "Availability", "KPI" and "Other".

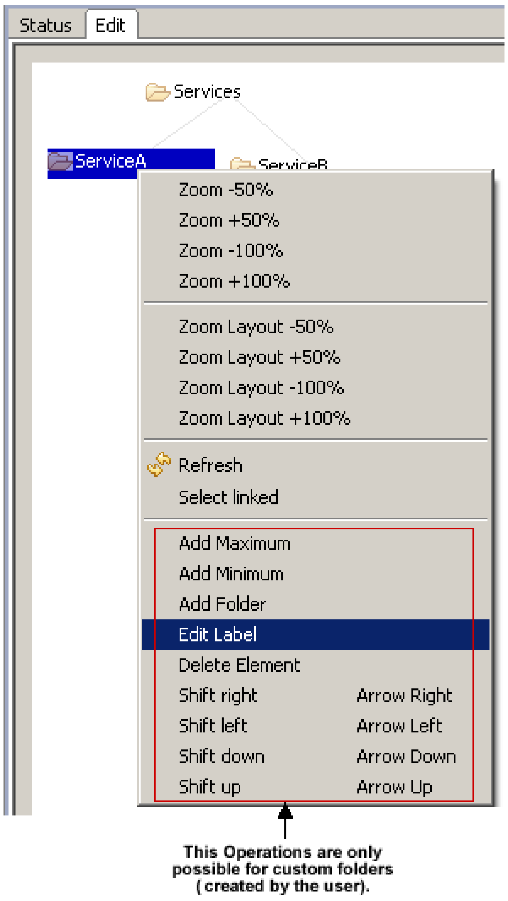
6.24.3. Customize Service Tree

Open the **Show Service Tree view** in the *boom* GUI.
Select **Edit** to customize the Service Tree.

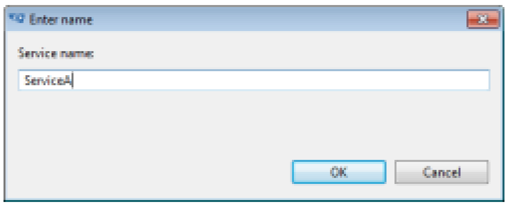
> Only users with "Owner Rights" on the service tree have permissions to change the global service tree definitions!



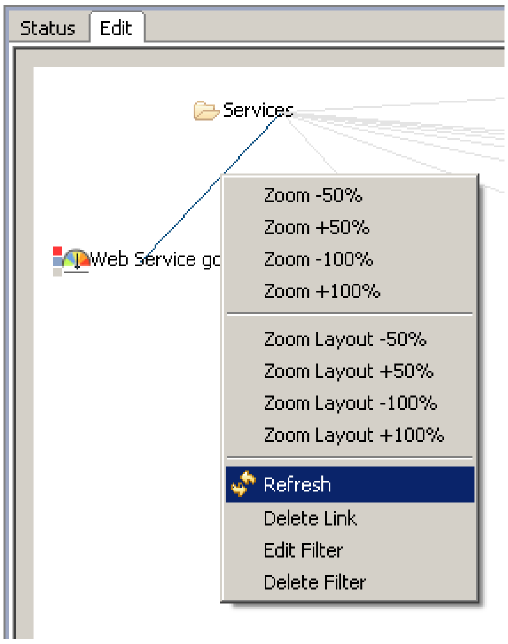
Right-click top/custom folder to open the context menu:



The following operations are only activated for custom elements. These are elements, that are created by the user!

Add Maximum:	
Add Minimum:	
Add Folder:	Add a custom folder to the "Service Tree": 
Edit Label:	Only labels of folders that are created by the user can be changed.
Delete Element:	Only custom folders that are created by the user can be deleted.
Shift right/Shiftleft:	Move custom folders that are created by the user one place to the right/left.
Shift up/Shiftdown:	Move custom folders that are created by the user one level up/down.

Right-click on a link to open the context menu:

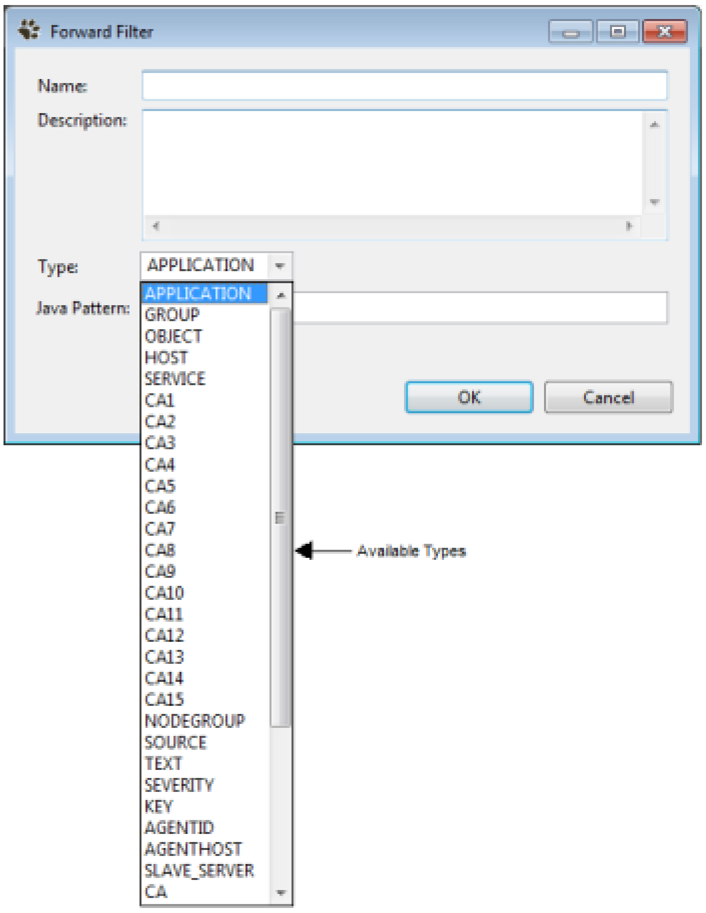


Add Link: To add a link you first have to highlight the 2 elements you would like to connect. Then press the 'Space Key' on your keyboard to add the link.

> You cannot add a link between more than 2 elements!

Delete Link: This deletes the selected link. You can also press the 'Del' key on your keyboard.

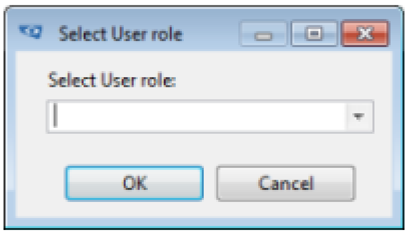
Edit Filter:

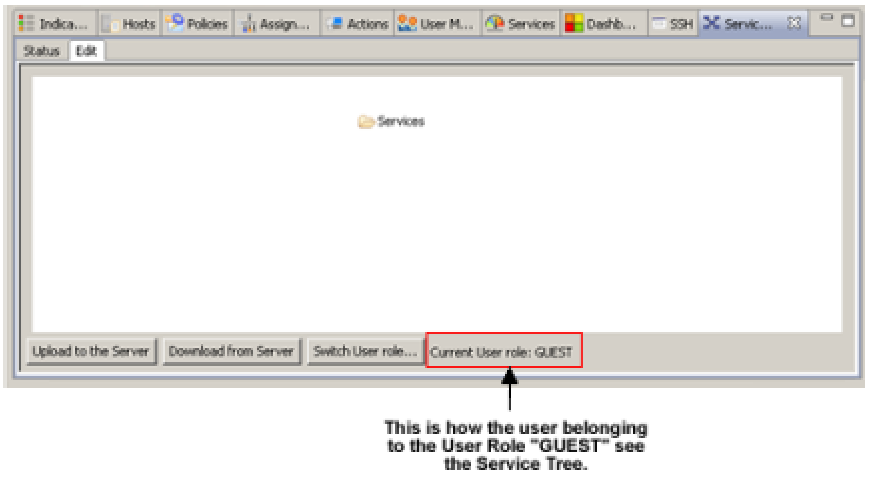


Delete Filter: This will delete the "Forward Filter".

The following general buttons are available:

Upload to the Server:	All changes that are made to the "Service Tree" will be saved locally. To make the tree a "Global Service Tree" you have to upload the tree to the *boom* server.
Download from the Server:	This will download the "Global Service Tree" definition to the local system.
Switch User Role:	You can switch the user role and see how the users belonging to this role see the service tree.





6.25. Maintenance Window Management

6.25.1. General Information

Maintenance Windows (formerly named Maintenances) are processed on the *boom* server. "Scheduled Maintenances" will be processed after "Modifiy Policies" and "AdHoc Maintenances".

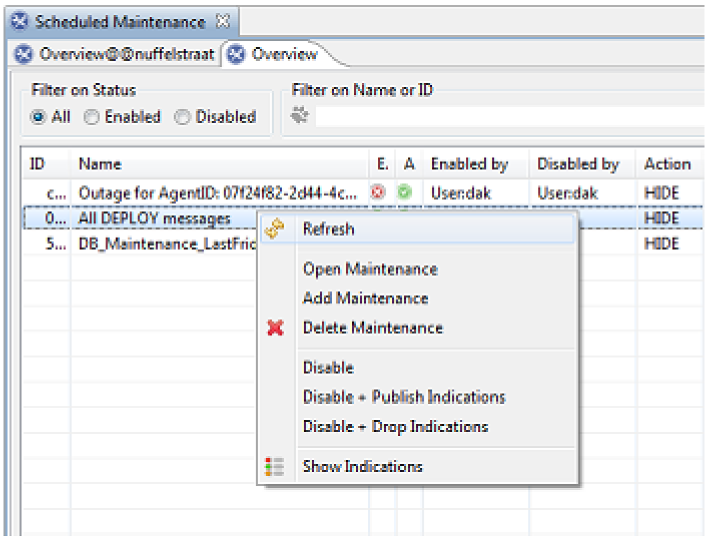
- The following Maintenance Actions are implemented:
  - DROP: Indications will be discarded
  - HIDE: Indications will come to a separate "Maintenance Indication Browser"
    - Post Action NONE the indications will stay in the Maintenance browser after the maintenance is ended
    - Post Action DROP the indications will be dropped from the Maintenance browser at the end of the maintenance
    - Post Action PUBLISH the indications will be moved to the active browser at the end of the maintenance
- Maintenance Processing:
  - Will be performed after indication enrichment/modification
  - Will be performed before indication De-duplication
  - Affects only new active/closed indications

- Doesn’t affect indication updates (Duplicates)
- Automatic actions are not performed during Maintenance time. A corresponding annotation is added ("Remote Action can’t be processed during Maintenance") to the indication.
- Forward filters with Exec-Target are not performed for indications that match an active Maintenance.

- Maintenance in a Master/Slave environment:
  - Maintenance can be configured for the **Master** and/or each Slave individually.
  - Maintenance configuration is only available for the *boom* Server Mode1 (Proxy = Slave).
  - Maintenance configuration is NOT available for Server Mode2 (Forward & Control) and Mode3 (Mirror).
  - **Recommendation:** Each Maintenance should be setup on the FIRST *boom* server receiving the corresponding indications (Slave).
    - For Primary-Backup (mode 8) the Primary (Master) server is the first *boom* server to receive the indications.
    - Automatic Actions can only be suppressed on the FIRST *boom* server (Slave)
    - Duplicates are created on the FIRST *boom* server:
      - If a Maintenance is activated on this FIRST server an indication update (Duplicate) is prevented the Maintenance because it is processed before De-duplication.
      - If the Maintenance isn’t activated on this FIRST server but on the Master server then an indication update (Duplicate) isn’t prevented by the Maintenance and therefore the indication is still visible and the Duplicate counter is increased.

6.25.2. Scheduled Maintenance Details and Operations

1. Use the **"Scheduled Maintenance"** in the GUI to view and configure the scheduled Maintenance Windows.
2. Right-click to open the context menu:

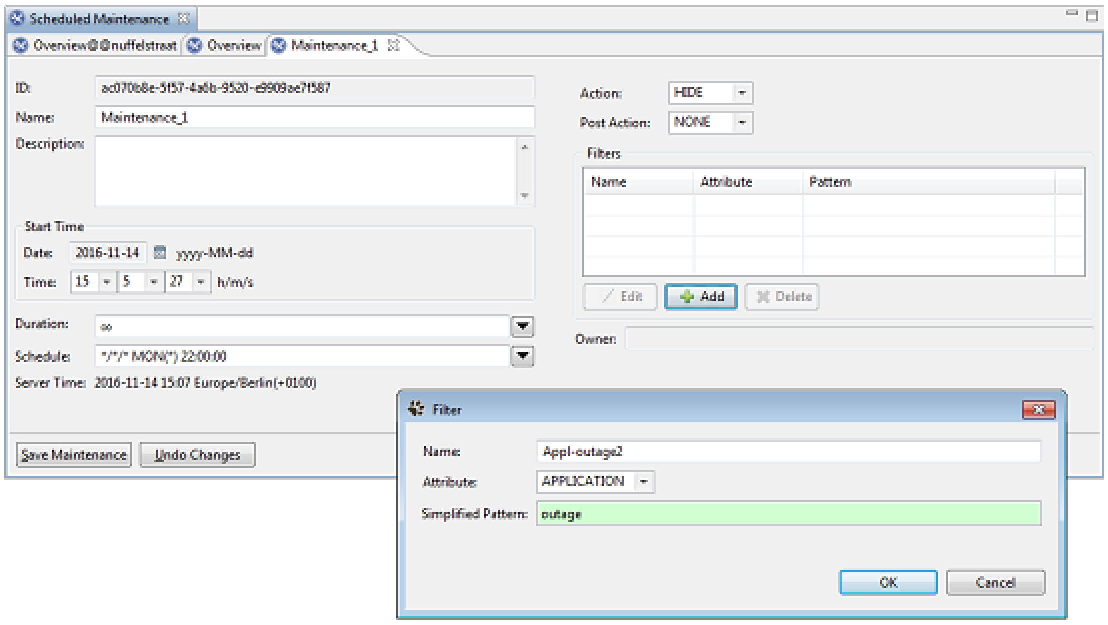


ID	Name	E.	A	Enabled by	Disabled by	Action
c...	Outage for AgentID: 07f24f82-2d44-4c...	✗	✓	User:dak	User:dak	HIDE
0...	All DEPLOY messages					HIDE
5...	DB_Maintenance_LastFric					HIDE

Add Maintenance:	Create a new entry for a Scheduled Maintenance Window
Enable:	Enable the Scheduled Maintenance
Show Indications:	Opens the Maintenance indication browser and shows all indications belonging to the selected maintenance.
Disable:	Disables the selected scheduled Maintenance

Disable + Publish Indications:	Disables the selected scheduled Maintenance and moves all indications from the Maintenance browser to the main indication browser
Disable + Drop Indications:	Disables the scheduled Maintenance and drops all indications from the Maintenance browser
Delete Maintenance:	Deletes the scheduled Maintenance

3. Select "Add Maintenance" to create a new Maintenance:



ID: ac070b8e-5f57-4a6b-9520-e9909ae7f587
Name: Maintenance_1
Description:
Start Time
Date: 2016-11-14 yyyy-MM-dd
Time: 15 5 27 h/m/s
Duration: ∞
Schedule: */*/* MON(*) 22:00:00
Server Time: 2016-11-14 15:07 Europe/Berlin(+0100)
Action: HIDE
Post Action: NONE
Owner:

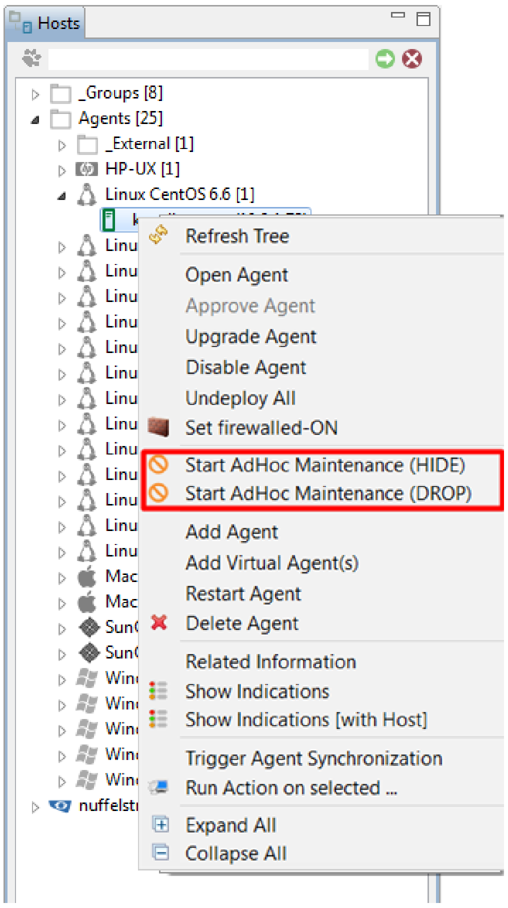
Filter
Name: Appl-outage2
Attribute: APPLICATION
Simplified Pattern: outage

The following parameters can be set:

Name:	Name of the scheduled Maintenance
Start Time:	Date and a time the Maintenance will become active e.g. a Maintenance can be created now but the start time can delayed to the next month.
Duration:	Time interval for the Maintenance (can also be unlimited). Format of intervals: 1w2d3h4m5s - One week, 2 days, 3 hours, 4 minutes, 5 seconds. Minimal interval is one minute(1m) Examples: 2h30m - two and half hour. 48h - 48 hours, 2d1s = 2 days and one second.
Schedule:	cron like scheduling for periodical maintenances Examples: + */*/* MON(*) 22:00: # every Monday at 22:00:00, */*/* 1-5(*) 08:00:00 # Monday-Friday at 08:00:00, */*/* *(*) 10,20:*:* # daily at 10 and 20 o'clock, */*/* *(*) *:05:00 # hourly at 5th minutes, */*/1L *(*) 23:00:00 # Last day of every month at 23:00:00, */*/* SAT(1L) 00:00:00 # last Saturday of each month at midnight, */*/15,1L *(*) 23:05:00 # 15th and last day of every month at 23:05:00 + # Last day of June, July, August at 23:00:00 */jun,jul,aug/1L *(*) 23:00:00 # First and Last day of 00 June, July, August at 23:00: `<br>`*/5-7/1,1L *(*) 23:00:00

Action:	[DROP|HIDE] DROP:all incoming indications during maintenance time will be dropped. HIDE: all incoming indications during maintenance time will be moved to the Maintenance browser [active]
PostAction:	[NONE|DROP|PUBLISH| NONE: all indications will stay in the Maintenance browser after the Maintenance is ended DROP: all indications will be dropped from the Maintenance browser at the end of the Maintenance PUBLISH: all indications will be moved to the main indication browser at the end of the Maintenance
Filters:	Supported indication attributes to filter which indications should be set to Maintenance. This set of filter must match with the incoming indication to process the indication.
Owner:	After the Maintenance Policy has been created – server sets "user" as Owner of this Maintenance.

4. **Second possibility to create a new Maintenance:**
   Open the Hosts view -→ select Agent -→ right-click -→ **Start AdHoc Maintenance(HIDE) | (DROP)**

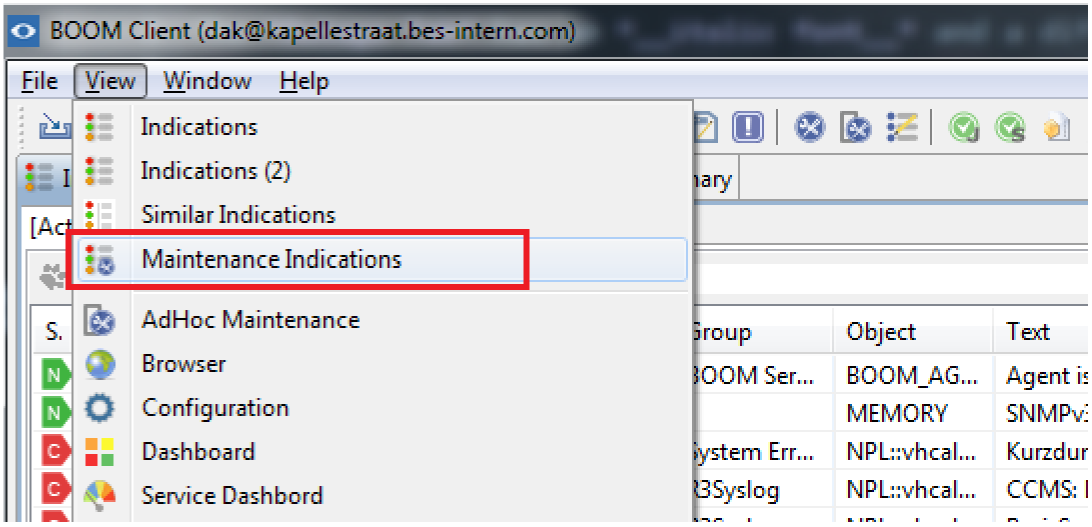


If a host has an active AdHoc Maintenance, it will be marked with the symbol 🚫 in the Hosts Tree view and the Agent Overview table.

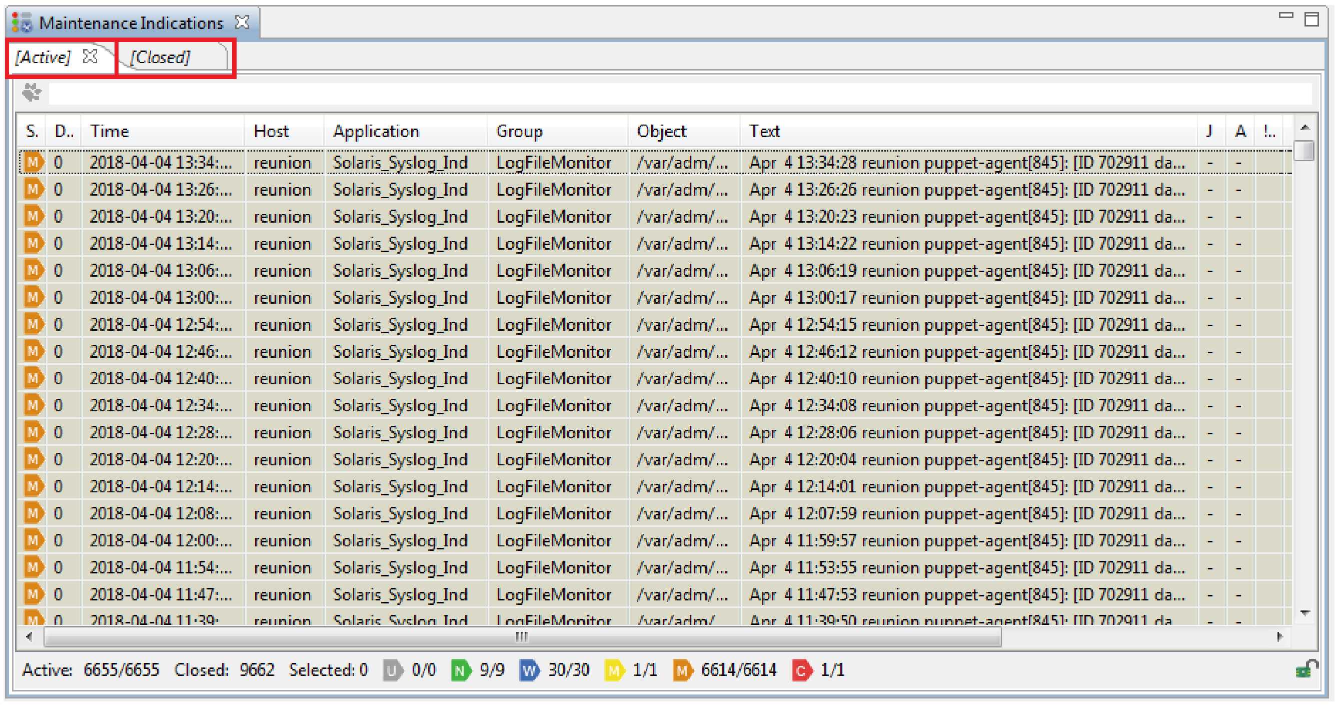
6.25.3. Indication Maintenance

An **Indication Maintenance view** exists in addition to the active indication browser. It holds all HIDDEN indications (action=HIDE) that are suppressed by an Maintenance. The Maintenance Repository has no archive, but it has de-duplication, correlation and closed indications.

The Maintenance Indications View can be opened in any UI clients via the View Menu by choosing **Maintenance Indications**.



The **Maintenance Indications View** has an ***italic font*** and a different background color.



S.	D..	Time	Host	Application	Group	Object	Text	J	A	!..
M	0	2018-04-04 13:34:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 13:34:28 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 13:26:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 13:26:26 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 13:20:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 13:20:23 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 13:14:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 13:14:22 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 13:06:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 13:06:19 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 13:00:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 13:00:17 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 12:54:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 12:54:15 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 12:46:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 12:46:12 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 12:40:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 12:40:10 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 12:34:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 12:34:08 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 12:28:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 12:28:06 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 12:20:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 12:20:04 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 12:14:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 12:14:01 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 12:08:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 12:07:59 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 12:00:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 11:59:57 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 11:54:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 11:53:55 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 11:47:...	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/...	Apr 4 11:47:53 reunion puppet-agent[845]: [ID 702911 da...	-	-	
M	0	2018-04-04 11:39:	reunion	Solaris_Syslog_Ind	LogFileMonitor	/var/adm/	Apr 4 11:39:50 reunion puppet-agent[845]: [ID 702911 da	-	-	

Active: 6655/6655 Closed: 9662 Selected: 0 U 0/0 N 9/9 W 30/30 M 1/1 M 6614/6614 C 1/1

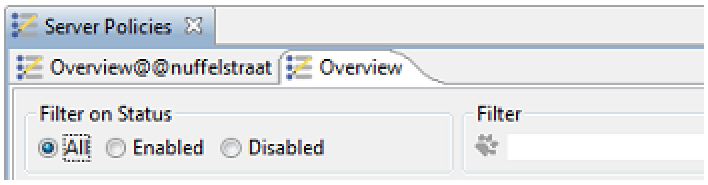
6.25.4. Scheduled Maintenance in a Master/Slave environment

A Scheduled Maintenance can be configured for the **Master** and/or each Slave individually. The Scheduled Maintenance configuration is only available for the *boom* Server Mode1 (Proxy = Slave).

The Scheduled Maintenance setup for the Master and/or each Slave can be configured on the Master

- Every Slave (Proxy) has to be configured individually via a predefined GUI Maintenance tab for each Slave **Overview@@<Slave>**
- The Scheduled Maintenance setup for the Master itself is done via a predefined GUI tab **Overview**
- Please note that for Primary-Backup (mode 8) maintenance windows are replicated from Primary to Backup server.

1. Open the Scheduled Maintenance view in the GUI to configure the Master and/or Slave server Maintenances.



6.26. AdHoc Maintenance

6.26.1. General Information

Any Agent has the possibility to trigger a Maintenance for itself. This AdHoc Maintenance has an expected duration and can have a fixed time period (see "–du" options). A typical example is a "backup time" e.g. when the agent must inform the server that for some hours all performance related indications must be suppressed or ignored. After the backup is finished the agent must inform the server to disable the maintenance. Multiple overlapping Maintenances are supported. AdHoc Maintenances will be processed before the Scheduled Maintenances!

AdHoc Maintenances are submitted only from the Agent and can't be modified. They can only be disabled or deleted. AdHoc Maintenances are stored in the database.
If an AdHoc Maintenance with an expected duration is not finished in that time, the server will create an according indication informing about the unexpected length of the maintenance.

Two Situations when to use an AdHoc Maintenances:

Example 1:

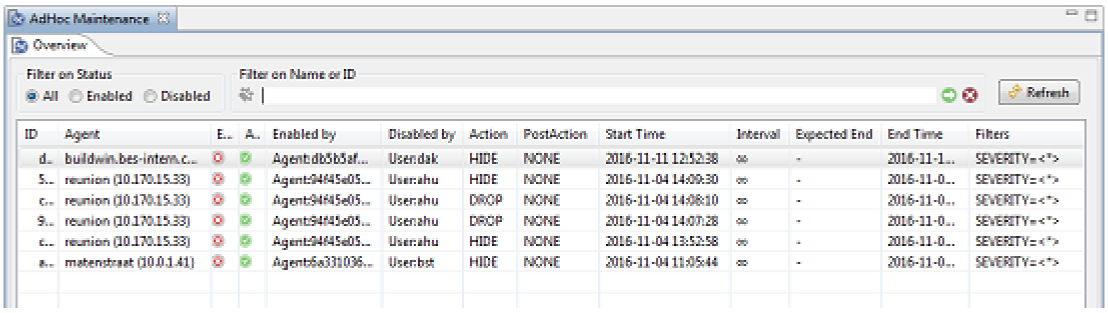
An Agent runs a backup script every friday. During the runtime of the backup script this Agent should be in maintenance mode. The script that is starting the backup can have a "Start Maintenance" at the begin and a "Stop Maintenance" at the end.

Example 2:

The administrator of the system can put the Agent manually into a maintenance mode via "**boom_agent.cli.jar**" (*boom* package which has to be deployed to the corresponding agent). At the end the AdHoc Maintenance must be deactivated by submitting "Disable Maintenance" or if auto-activated the server will disable the AdHoc Maintenance automatically after the specified interval even if the agent doesn't send a "stop".

6.26.2. AdHoc Maintenance Details and Operations

All AdHoc Maintenances are listed in the "AdHoc Maintenance View". An AdHoc Maintenances can only be disabled or deleted, they cannot be modified inside the GUI.



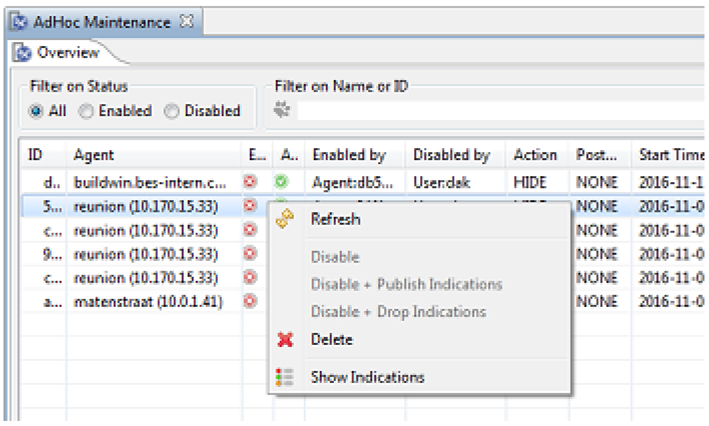
ID	Agent	E..	A..	Enabled by	Disabled by	Action	PostAction	Start Time	Interval	Expected End	End Time	Filters
d..	buildwin.bes-intern.c...	⊗	✓	Agent:db5b5af...	User:dak	HIDE	NONE	2016-11-11 12:52:38	∞	-	2016-11-1...	SEVERITY=<*>
5...	reunion (10.170.15.33)	⊗	✓	Agent:94f45e05...	User:ahu	HIDE	NONE	2016-11-04 14:09:30	∞	-	2016-11-0...	SEVERITY=<*>
c...	reunion (10.170.15.33)	⊗	✓	Agent:94f45e05...	User:ahu	DROP	NONE	2016-11-04 14:08:10	∞	-	2016-11-0...	SEVERITY=<*>
9...	reunion (10.170.15.33)	⊗	✓	Agent:94f45e05...	User:ahu	DROP	NONE	2016-11-04 14:07:28	∞	-	2016-11-0...	SEVERITY=<*>
c...	reunion (10.170.15.33)	⊗	✓	Agent:94f45e05...	User:ahu	HIDE	NONE	2016-11-04 13:52:58	∞	-	2016-11-0...	SEVERITY=<*>
a...	matenstraat (10.0.1.41)	⊗	✓	Agent:6a331036...	User:bst	HIDE	NONE	2016-11-04 11:05:44	∞	-	2016-11-0...	SEVERITY=<*>

+

Name	Description
Id	unique identification of the maintenance
Agent	Agent to which the maintenance belongs

Name	Description
Enabled	[enabled | disabled] Main state of the maintenance. Will be enabled with the *boom* command line interface (boom_agent.cli.jar).
Active	[active | inactive] Additional state which shows if the maintenance is effective e.g. the maintenance can be active every day from 22:00 to 23:00
Enabled by	ID of the Agent which set the main state of the maintenance to enabled.
Disabled by	[Auto | User] ID of the Agent which set the main state of the maintenance to disabled. Auto means the maintenance has beeen disabled automatically after the scheduled time.
Action	[DROP | HIDE] DROP: all incoming indications during maintenance time will be dropped. HIDE: all incoming indications during maintenance time will be moved to the Maintenance browser [active]
PostAction	[NONE | DROP | PUBLISH | NONE: all indications will stay in the Maintenance browser after the Maintenance is ended DROP: all indications will be dropped from the Maintenance browser at the end of the Maintenance PUBLISH: all indications will be moved to the main indication browser at the end of the Maintenance
Start Time	Start time of the Maintenance
Interval	Expected duration of the Maintenance
Expected End	Expected finish time of the Maintenance
End Time	End time of the Maintenance
Filters	Supported indication attributes to filter which indications should be set to maintenance. This set of filter must match with the incoming indication to process the indication.

1. Right-click on an AdHoc Maintenance to open the context menu:



ID	Agent	E...	A...	Enabled by	Disabled by	Action	Post...	Start Time
d...	buildwin.bes-intern.c...			Agent:db5...	User:dak	HIDE	NONE	2016-11-1
5...	reunion (10.170.15.33)						NONE	2016-11-0
c...	reunion (10.170.15.33)						NONE	2016-11-0
9...	reunion (10.170.15.33)						NONE	2016-11-0
c...	reunion (10.170.15.33)						NONE	2016-11-0
a...	matenstraat (10.0.1.41)						NONE	2016-11-0

Show Indications:

Opens the Maintenance indication browser and shows all indications belonging to the selected Maintenance.

Disable:

Disables the selected Maintenance

Disable + Publish Indications:

Disables the selected Maintenance and moves all indications from the Maintenance browser to the main indication browser

Disable + Drop Indications:

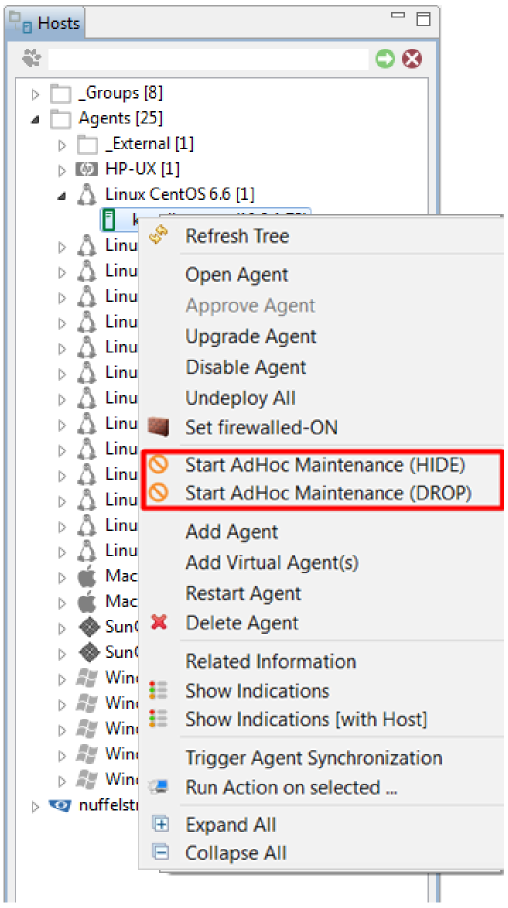
Disables the selected Maintenance and drops all indications from the Maintenance browser

Delete:

Deletes the Maintenance from the database

6.26.3. Create a Manual AdHoc Maintenance

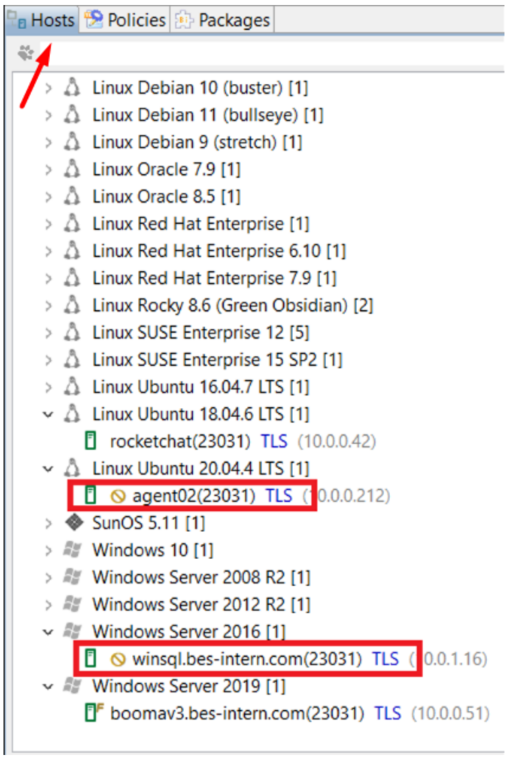
Open the Hosts view -→ select Agent -→ right-click -→ **Start AdHoc Maintenance (HIDE)** | **Start AdHoc Maintenance (DROP)**



The resulting AdHoc Maintenance will be created with immediate start time, infinite duration and will be activated immediately.

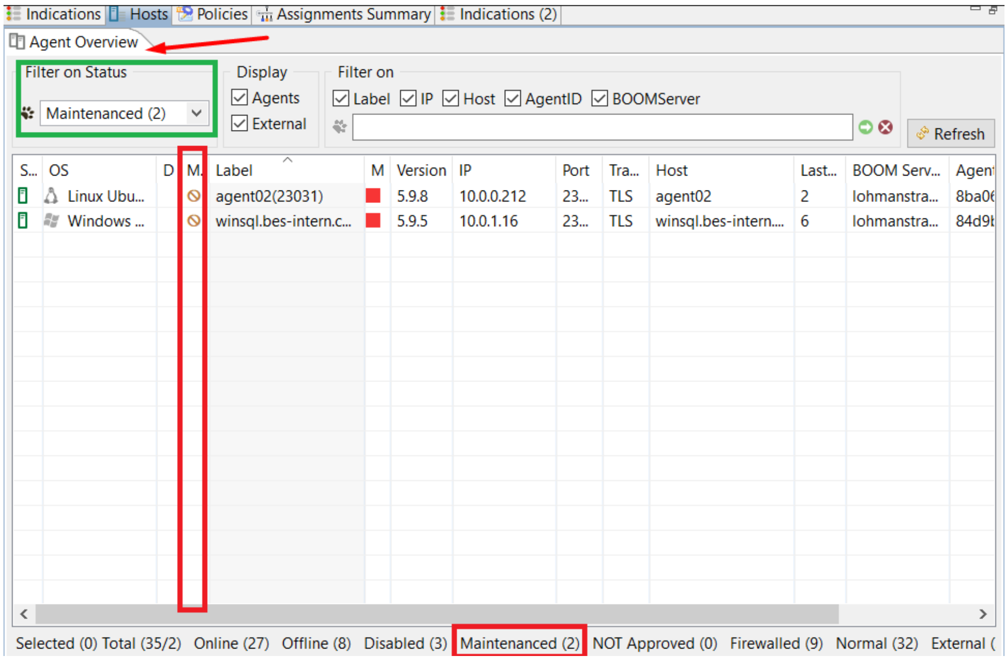
ℹ	If a host has an active AdHoc Maintenance, it will be marked with the symbol ⃠ in the Hosts Tree view and the Agent Overview table.

Hosts Tree view



- Linux Debian 10 (buster) [1]
- Linux Debian 11 (bullseye) [1]
- Linux Debian 9 (stretch) [1]
- Linux Oracle 7.9 [1]
- Linux Oracle 8.5 [1]
- Linux Red Hat Enterprise [1]
- Linux Red Hat Enterprise 6.10 [1]
- Linux Red Hat Enterprise 7.9 [1]
- Linux Rocky 8.6 (Green Obsidian) [2]
- Linux SUSE Enterprise 12 [5]
- Linux SUSE Enterprise 15 SP2 [1]
- Linux Ubuntu 16.04.7 LTS [1]
- Linux Ubuntu 18.04.6 LTS [1]
  - rocketchat(23031) TLS (10.0.0.42)
- Linux Ubuntu 20.04.4 LTS [1]
  - agent02(23031) TLS (10.0.0.212)
- SunOS 5.11 [1]
- Windows 10 [1]
- Windows Server 2008 R2 [1]
- Windows Server 2012 R2 [1]
- Windows Server 2016 [1]
  - winsql.bes-intern.com(23031) TLS (10.0.1.16)
- Windows Server 2019 [1]
  - boomav3.bes-intern.com(23031) TLS (10.0.0.51)

Agent Overview table



Filter on Status: Maintenanced (2)

Display: ☑ Agents ☑ External

Filter on: ☑ Label ☑ IP ☑ Host ☑ AgentID ☑ BOOMServer

S...	OS	D	M...	Label	M	Version	IP	Port	Tra...	Host	Last...	BOOM Serv...	Agen
	Linux Ubu...			agent02(23031)	🟥	5.9.8	10.0.0.212	23...	TLS	agent02	2	lohmanstra...	8ba0
	Windows ...			winsql.bes-intern.c...	🟥	5.9.5	10.0.1.16	23...	TLS	winsql.bes-intern....	6	lohmanstra...	84d9

Selected (0) Total (35/2) Online (27) Offline (8) Disabled (3) Maintenanced (2) NOT Approved (0) Firewalled (9) Normal (32) External (

6.27. Server Policies

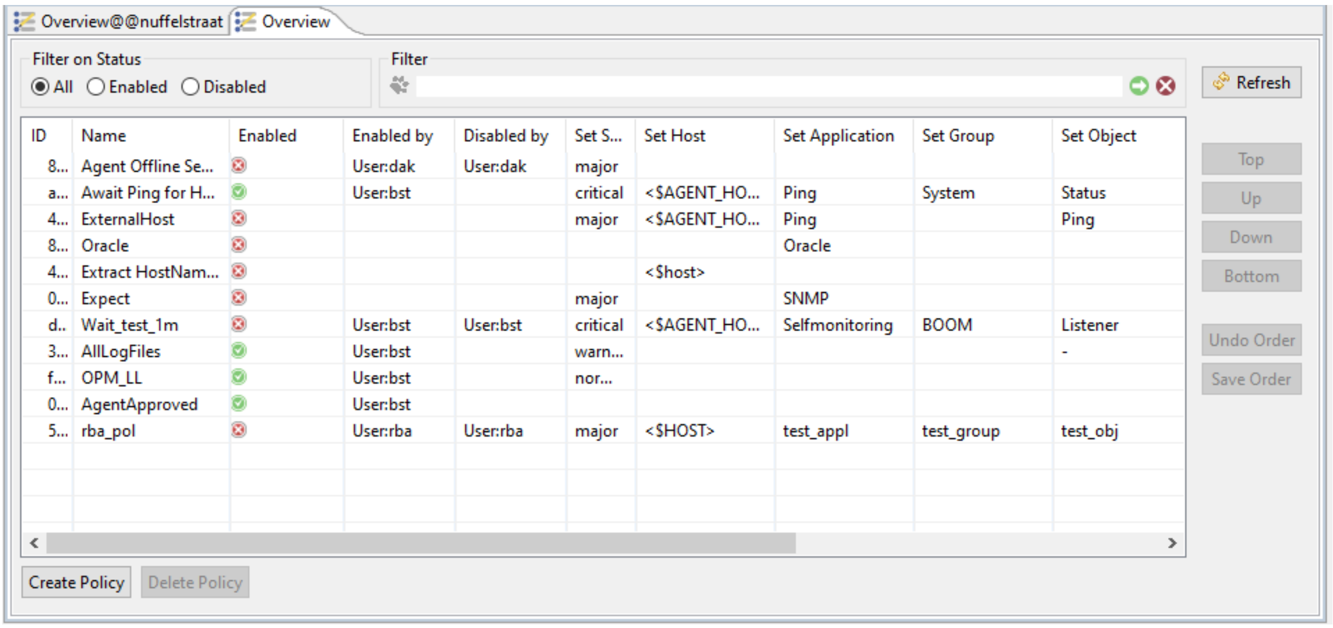
6.27.1. General Information

Server policies allow to modify incoming indications on the server before the Maintenance, de-duplication and correlation steps take place. Server Policies are used to change the incoming indication attributes. A typical use case is the unification of attributes, e.g. if multiple data sources deliver different names for the same application a modify policy might be used to override the application attribute.

Based on filter(s) the following indication attributes can be modified:

- Host
- Severity
- Application
- Group
- Object
- Text
- Key
- Close Mask
- AutoAction
- AutoAction Node
- AutoAction Timeout
- Force Insert As Closed
- Metric Flags
- Duplicates
- Custom Attributes

Use the **"Server Policies"** view to enrich/modify attributes of incoming indications.



ID	Name	Enabled	Enabled by	Disabled by	Set S...	Set Host	Set Application	Set Group	Set Object
8...	Agent Offline Se...	✗	User:dak	User:dak	major				
a...	Await Ping for H...	✓	User:bst		critical	<$AGENT_HO...	Ping	System	Status
4...	ExternalHost	✗			major	<$AGENT_HO...	Ping		Ping
8...	Oracle	✗					Oracle		
4...	Extract HostNam...	✗				<$host>			
0...	Expect	✗			major		SNMP		
d..	Wait_test_1m	✗	User:bst	User:bst	critical	<$AGENT_HO...	Selfmonitoring	BOOM	Listener
3...	AllLogFiles	✓	User:bst		warn...				-
f...	OPM_LL	✓	User:bst		nor...				
0...	AgentApproved	✓	User:bst						
5...	rba_pol	✗	User:rba	User:rba	major	<$HOST>	test_appl	test_group	test_obj

The Overview displays all configured Modification Policies. The displayed policies can be filtered by status and content:

Filter on status:

All - show all policies
Enabled - show only enabled policies
Disabled - show only disabled policies

Filter:

search for the string in the policy attributes

Processing Order:

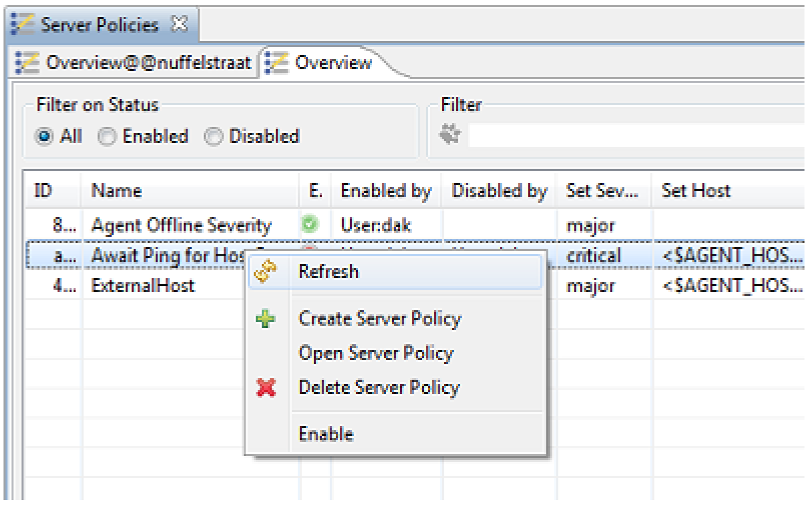
An incoming message is processed by all "Server Policies" according to their order. If an incoming indication matches several "Server Policies" the last attribute match is valid. e.g. If the first matching policy sets the

Application field to AAA, the second matching policy sets the Application field to BBB and one further matching policy sets it to CCC, then CCC will be the result.

ℹ	All Disabled "Server Policies" are ignored.

6.27.2. Server Policies Operations

1. Right-click to **open the context menu** or select the **Add/Delete** button

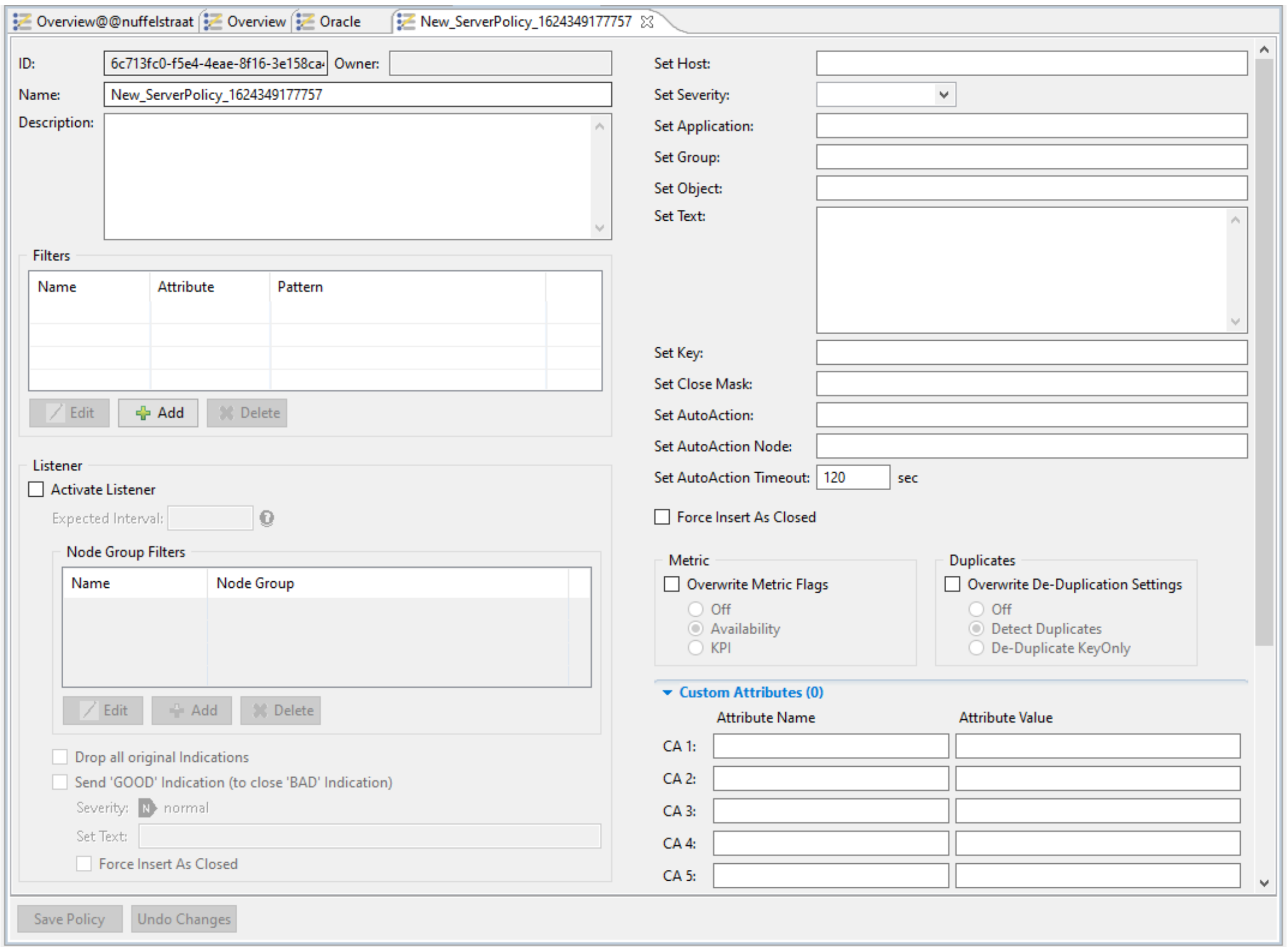


Filter on Status: [✓ All] Enabled Disabled

ID	Name	E.	Enabled by	Disabled by	Set Sev...	Set Host
8...	Agent Offline Severity	✓	User:dak		major	
a...	Await Ping for Hos				critical	<$AGENT_HOS...
4...	ExternalHost				major	<$AGENT_HOS...

Create:	Add a new Server Policy
Open:	Edit the selected Server Policy
Enable:	Enable the selected Server Policy
Disable:	Disable the selected Server Policy
Delete:	Delete the selected Server Policy

2. **Add a new Server Policy**
   Select **Create** to add a new "Server Policy":


Overview@@nuffelstraat | Overview | Oracle | New_ServerPolicy_1624349177757

ID: 6c713fc0-f5e4-4eae-8f16-3e158ca Owner:
Name: New_ServerPolicy_1624349177757
Description:

Filters

Name	Attribute	Pattern

Edit | Add | Delete

Listener
☐ Activate Listener
Expected Interval:

Node Group Filters

Name	Node Group

Edit | Add | Delete

☐ Drop all original Indications
☐ Send 'GOOD' Indication (to close 'BAD' Indication)
Severity: normal
Set Text:
☐ Force Insert As Closed

Set Host:
Set Severity:
Set Application:
Set Group:
Set Object:
Set Text:
Set Key:
Set Close Mask:
Set AutoAction:
Set AutoAction Node:
Set AutoAction Timeout: 120 sec
☐ Force Insert As Closed

Metric
☐ Overwrite Metric Flags
Off [✓ Availability] KPI

Duplicates
☐ Overwrite De-Duplication Settings
Off [✓ Detect Duplicates] De-Duplicate KeyOnly

Custom Attributes (0)

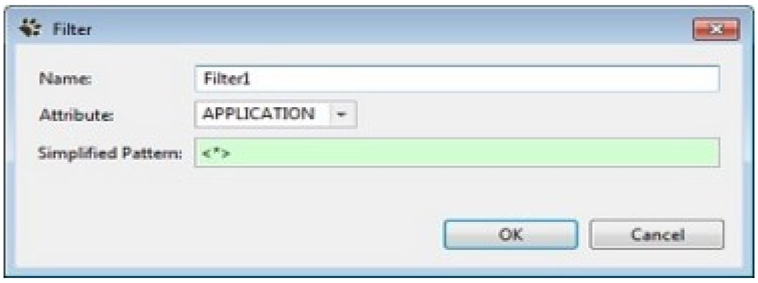
	Attribute Name	Attribute Value
CA 1:		
CA 2:		
CA 3:		
CA 4:		
CA 5:		

Save Policy | Undo Changes

On the left side of the tab the general policy information and the conditions are specified. On the right side specify the attribute values that should be set.

i	the Set AutoAction Timeout value will be only used if a new AutoAction is specified.

3. **Based on indication attribute filter(s) the attributes can be enriched/modified:**
Select **Add** to create a new filter to filter incoming indications:


Filter

Name: Filter1
Attribute: APPLICATION
Simplified Pattern: <*>

OK | Cancel

The following filter types are available. The Simplified pattern can be used to set the filter for the selected type field.

Type field	Description
APPLICATION	Application name
GROUP	Indication Group name
OBJECT	Object name
HOST	Node from where the Indication has been issued

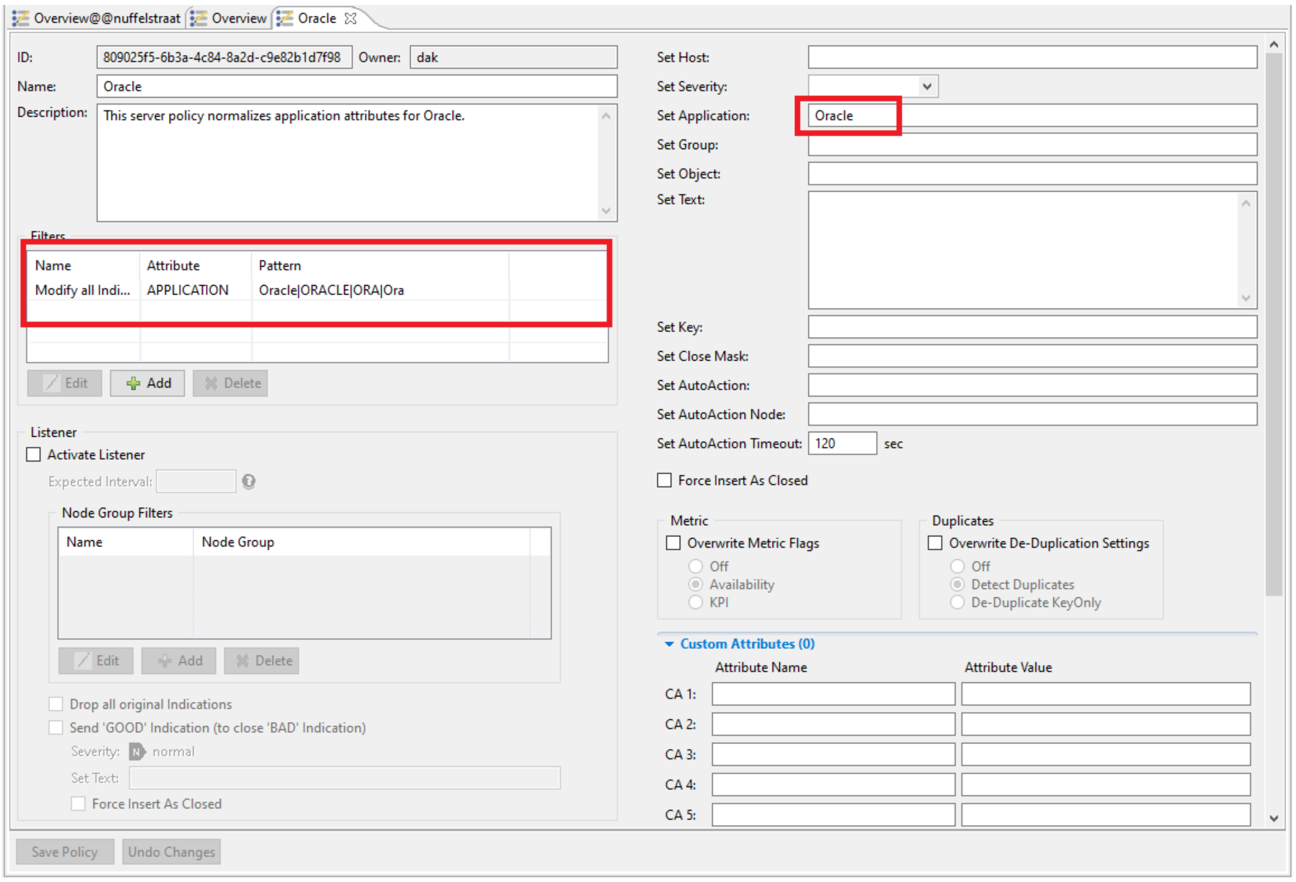
Type field	Description
SERVICE	reserved
CA1 - CA15	match the value of one specific CA from 1 to 15
NODEGROUP	NodeGroup name - use "Select" to filter one or more Node Groups
SOURCE	source policy
TEXT	Indication text
SEVERITY	severity of the Indication
KEY	Indication Key
AGENTID	Unique number to identify the *boom* agent - use "Select" to filter one or more Agents and/or Node Groups
AGENTHOST	Host Name of the *boom* agent
SLAVE_SERVER	Name of the *boom* Slave Server
CA	match the value of any CA from CA1 to CA15

4. To **activate** the "Server Policy" select "enable" in the context menu.

6.27.3. Usage

1. **Standardization of Attributes**
   Normalize the name of one or more incoming attibutes to a standard name.

   example:



ID: 809025f5-6b3a-4c84-8a2d-c9e82b1d7f98 Owner: dak
Name: Oracle
Description: This server policy normalizes application attributes for Oracle.

Filters

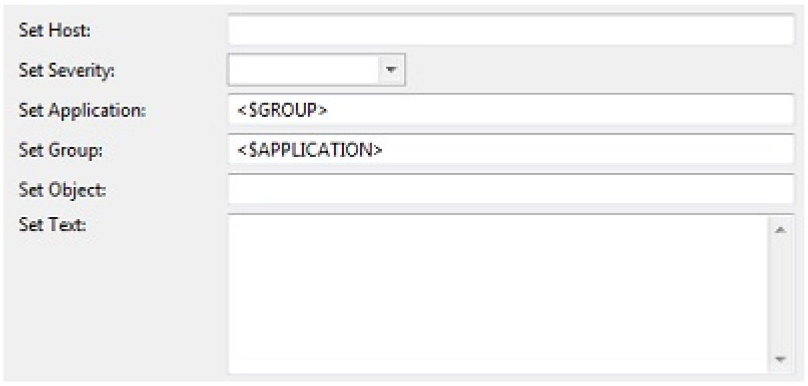
Name	Attribute	Pattern
Modify all Indi...	APPLICATION	Oracle|ORACLE|ORA|Ora

Set Host:
Set Severity:
Set Application: Oracle
Set Group:
Set Object:
Set Text:
Set Key:
Set Close Mask:
Set AutoAction:
Set AutoAction Node:
Set AutoAction Timeout: 120 sec

The Application name of all incoming indications with a similar application name "Oracle|ORACLE|ORA|ORADB" will be changed to "Oracle".

2. **Interchange of Attributes**
   Exchange the incoming indication attributes.

   example:

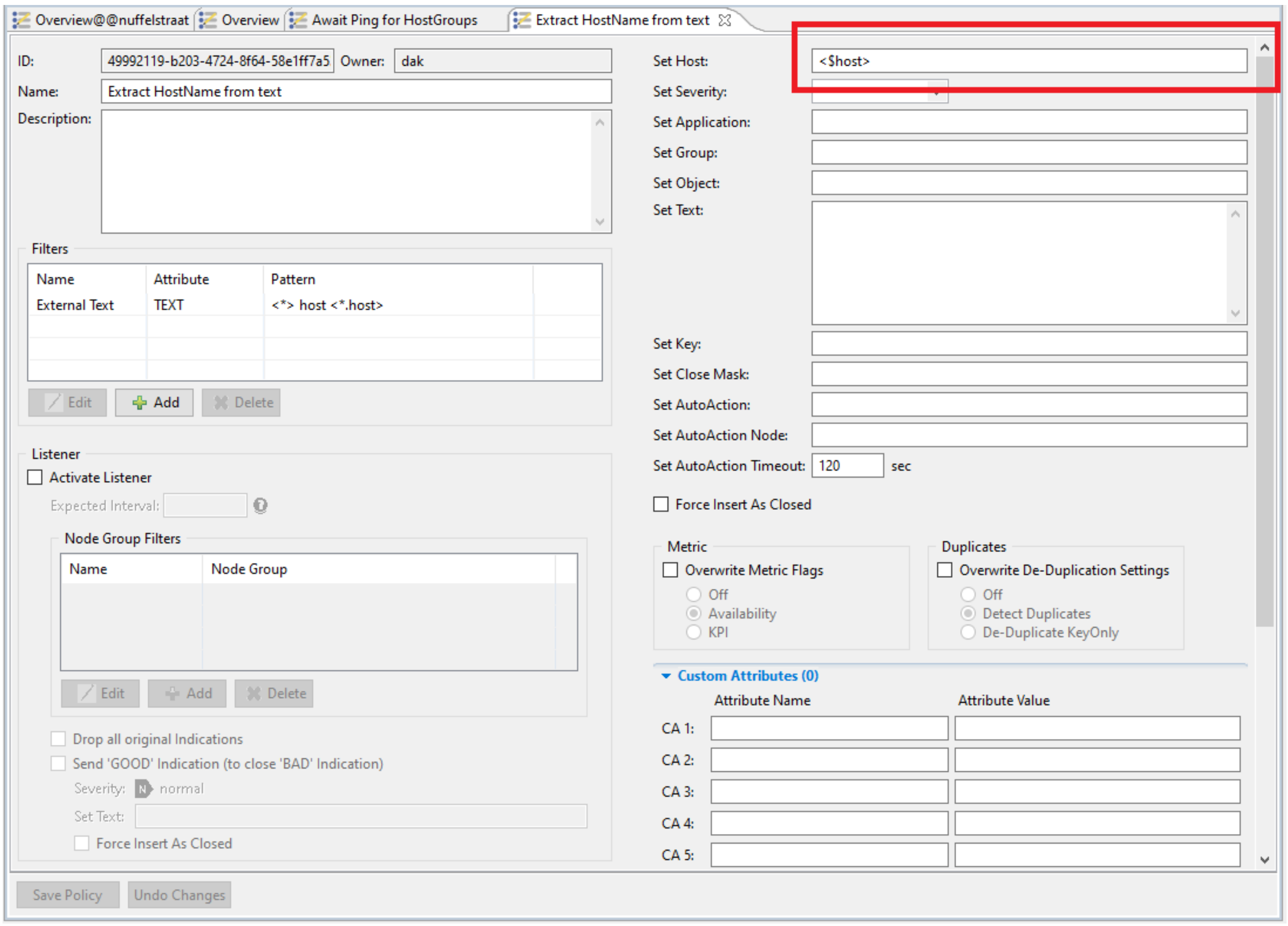


Set Host:	
Set Severity:	
Set Application:	<$GROUP>
Set Group:	<$APPLICATION>
Set Object:	
Set Text:	

   Use the incoming indication attribute Group name as Application attribute name and the Application attribute name as Group name.

3. **Use of variables inside pattern field**
   Variables can be defined inside the pattern field of "Filters" and they can be reused in the "Set" fields.

   example:



ID:	49992119-b203-4724-8f64-58e1ff7a5
Owner:	dak
Name:	Extract HostName from text
Description:	

Filters

Name	Attribute	Pattern
External Text	TEXT	<*> host <*.host>

Listener

☐ Activate Listener

Expected Interval:

Node Group Filters

Name	Node Group

☐ Drop all original Indications

☐ Send 'GOOD' Indication (to close 'BAD' Indication)

Severity: normal

Set Text:

☐ Force Insert As Closed

Set Host:	<$host>
Set Severity:	
Set Application:	
Set Group:	
Set Object:	
Set Text:	
Set Key:	
Set Close Mask:	
Set AutoAction:	
Set AutoAction Node:	
Set AutoAction Timeout:	120 sec

☐ Force Insert As Closed

Metric

☐ Overwrite Metric Flags

Off [✓ Availability] KPI

Duplicates

☐ Overwrite De-Duplication Settings

Off [✓ Detect Duplicates] De-Duplicate KeyOnly

Custom Attributes (0)

	Attribute Name	Attribute Value
CA 1:		
CA 2:		
CA 3:		
CA 4:		
CA 5:		

The variable <$host> will be defined during parsing of the indication Text and can be used in "Set Host" .

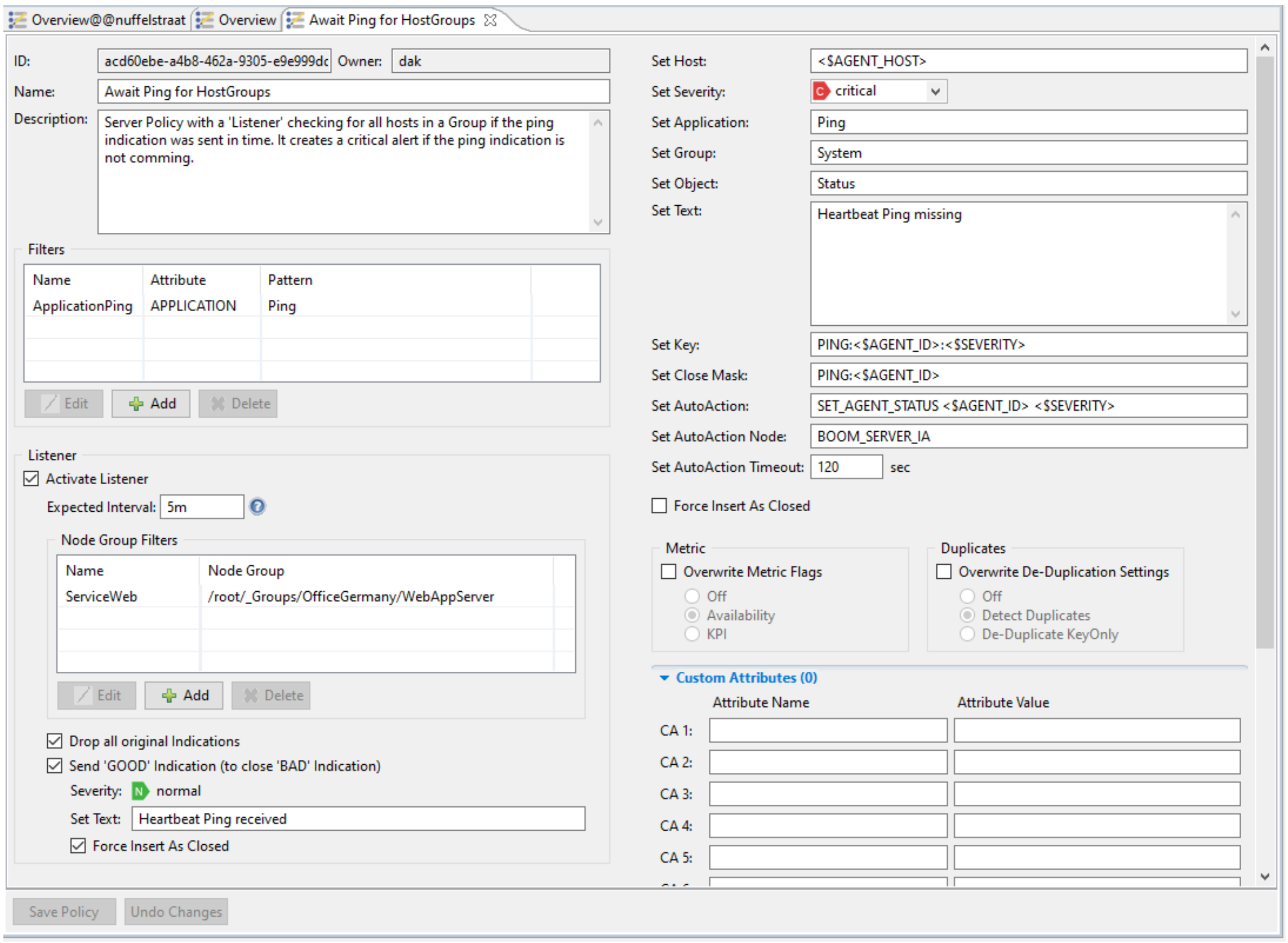
6.27.4. Supported Variables

The following default variables are supported in the "Set" fields:

Variable	Description
<$AGENT_HOST>	hostname of *boom* agent
<$AGENT_ID>	universal unique ID of a *boom* agent
<$AKEY>	close mask
<$APPLICATION>	Application
<$DUPLICATES>	number of Duplicates
<$FTIME_STR>	Agent time of first indication (yyyy-MM-dd HH:mm:ss)
<$FTIME>	Agent time of first indication (milliseconds)
<$GROUP>	Indication Group
<$HOST>	Node from where the indication has been issued
<$IID>	Indication ID
<$KEY>	Indication key
<$OBJECT>	Object
<$SEVERITY>	Severity
<$STIME_STR>	Server Time of indication/last Duplicate (yyyy-MM-dd HH:mm:ss)
<$STIME>	Server Time of indication/last Duplicate (milliseconds)
<$TIME_STR>	Agent Time of Last Duplicate Indication (yyyy-MM-dd HH:mm:ss)
<$TIME>	Agent Time of Last Duplicate Indication (milliseconds)
<$TEXT>	Indication Text
<$AA.attributeLabel>	Agent Attribute values, where "attributeLabel" is one of defined Agent Attribute labels

6.27.5. Server Policies - Listener

The Server Policies behave different if they are configured as "Listener Policy". These type of policies allow to implement so called application heartbeats or checkpoints.
In order to ensure that an application is working correctly it is in most cases not enough to verify that a process is running. An active check that tests the correct work of an application or an application heartbeat can be used to make sure that the application is not only running but is fully functional. For application heartbeats or checkpoints an application is customized or extended so that it emits in regular intervals or after an important regular operation switch (e.g. switching between dialog and batch mode) an indication to the control instance, in this case boom. If these indications are received in time it is ensured that the application is still working. Listener Policies are enabling to implement this type of application heartbeat and checkpoint monitoring.



ID: acd60ebe-a4b8-462a-9305-e9e999dc Owner: dak

Name: Await Ping for HostGroups

Description: Server Policy with a 'Listener' checking for all hosts in a Group if the ping indication was sent in time. It creates a critical alert if the ping indication is not comming.

Filters

Name	Attribute	Pattern
ApplicationPing	APPLICATION	Ping

Listener

[✓] Activate Listener

Expected Interval: 5m

Node Group Filters

Name	Node Group
ServiceWeb	/root/_Groups/OfficeGermany/WebAppServer

[✓] Drop all original Indications

[✓] Send 'GOOD' Indication (to close 'BAD' Indication)

Severity: normal

Set Text: Heartbeat Ping received

[✓] Force Insert As Closed

Set Host:	<$AGENT_HOST>
Set Severity:	critical
Set Application:	Ping
Set Group:	System
Set Object:	Status
Set Text:	Heartbeat Ping missing
Set Key:	PING:<$AGENT_ID>:<$SEVERITY>
Set Close Mask:	PING:<$AGENT_ID>
Set AutoAction:	SET_AGENT_STATUS <$AGENT_ID> <$SEVERITY>
Set AutoAction Node:	BOOM_SERVER_IA
Set AutoAction Timeout:	120 sec

[ ] Force Insert As Closed

Metric: [ ] Overwrite Metric Flags — Off / [● Availability] / KPI

Duplicates: [ ] Overwrite De-Duplication Settings — Off / [● Detect Duplicates] / De-Duplicate KeyOnly

Custom Attributes (0)

	Attribute Name	Attribute Value
CA 1:		
CA 2:		
CA 3:		
CA 4:		
CA 5:		

The filtering will be configured as described before. As soon as the Listener - **Activate Listener** flag is set, at least one node group filter must be specified in the **Node Group Filters** section. The **Expected Interval** field must be specified in minutes (m), hours (h), or days (d).
The Listen Policy expects for each host in the selected host groups filtered by the **Node Group Filters** at least once per **Expected Interval** an indication matching the **Filters** is received. If during the interval for an host no matching indication is received, a new failure indication with the attributes specified on the right side in the Set fields is generated. If an indication is received and the **Send GOOD Indication** flag is set, a new indication is generated that will close any before generated failure indication and has the attributes like the failure indication except the **Severity** (will be normal) and the text that can be overwritten with the **Set Text** field in this section. In this section the **Force Insert As Closed** flag specifies that the new indication is generated as closed indication.
The modifypolicy_listener.png flag specifies if an matching incoming indication will be dropped or further processed as any other indication. After ensuring that the policy is working as expected the original indications in most cases can be dropped as they will only fill up the database without providing additional information.

The example above implements an additional heartbeat mechanism for *boom* testing the processing chain from the monitor on the Agent to the server. Assume there is a monitor deployed on each host in the host group WebAppServer, which runs every 5 minutes, uses boomindi to submit an indication with the application attribute Ping and returns just a dummy OK value as monitor value. The Listener Policy expects every 5 minutes to get from each host in the host group WebAppServer an indication with the application attribute value Ping. If the indications are coming as expected, they will be simply dropped and nothing appears in the indication browser. If for some reason an host fails to submit the indication in time, a critical indication will be generated with the Host set to the expected host, Group set to "System", Object set to "Status" and Text "Heartbeat Ping missing". The configured automatic action will be performed on the *boom* server and set the Agent Status to offline (all severities except normal result in offline state).

If for such an host the first expected indication is received again, an indication with severity normal, Host set to the expected host, Group set to "System", Object set to "Status",Text "Heartbeat Ping received" and state "Closed" is generated. The configured automatic action will be performed on the *boom* server and set the Agent Status to online (severity normal). The configured Close Mask will automatically close the previously generated critical indication.

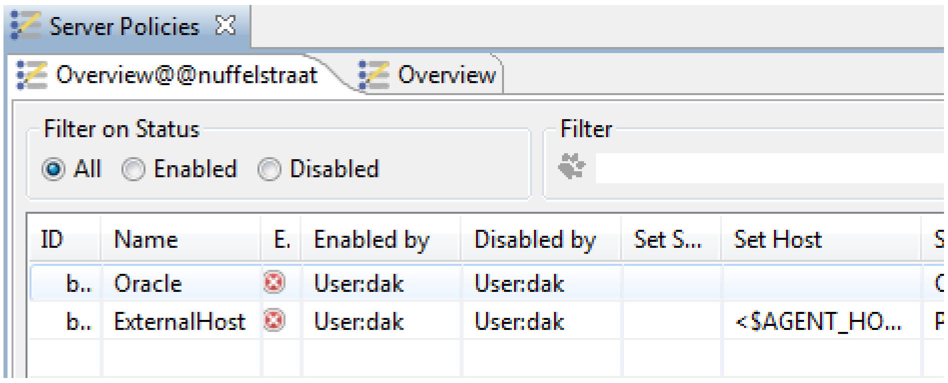
6.27.6. Server Policies in a Master/Slave environment

In a *boom* Master/Slave environment "Server Policies" are available on both Master and Slave and they are completely independent of each other. They can be configured on the Master server and/or on every Slave server individually. The Slave configuration can be done either on the Master server or the Slave server itself.

Slave configuration on Master server:

- If the user interface is connected to a master server with slaves the Server Policies View will contain tabs named "Overview@@<Slave>" for every slave. Each Slave can be configured individually by using the related tab.

e.g. Overview@@nuffelstraat

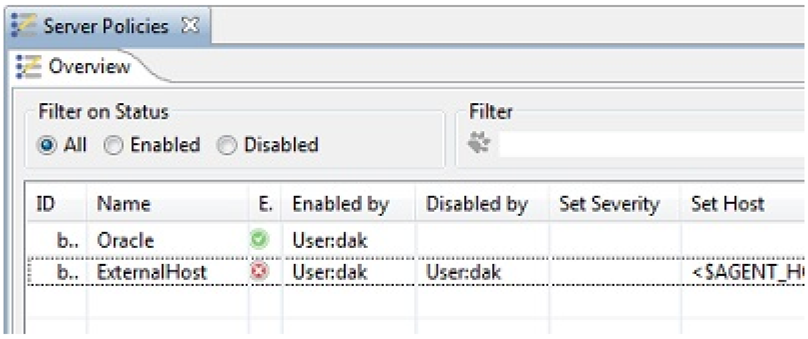


ID	Name	E.	Enabled by	Disabled by	Set S...	Set Host	S
b..	Oracle	⊗	User:dak	User:dak			C
b..	ExternalHost	⊗	User:dak	User:dak		<$AGENT_HO...	F

- The **Master server** itself has is configured via a predefined GUI tab **"Overview"**.

Slave configuration on Slave server itself:

- If the user interface is connected to the Slave server only the **Overview** tab is visible and is used to configure the Slave server.



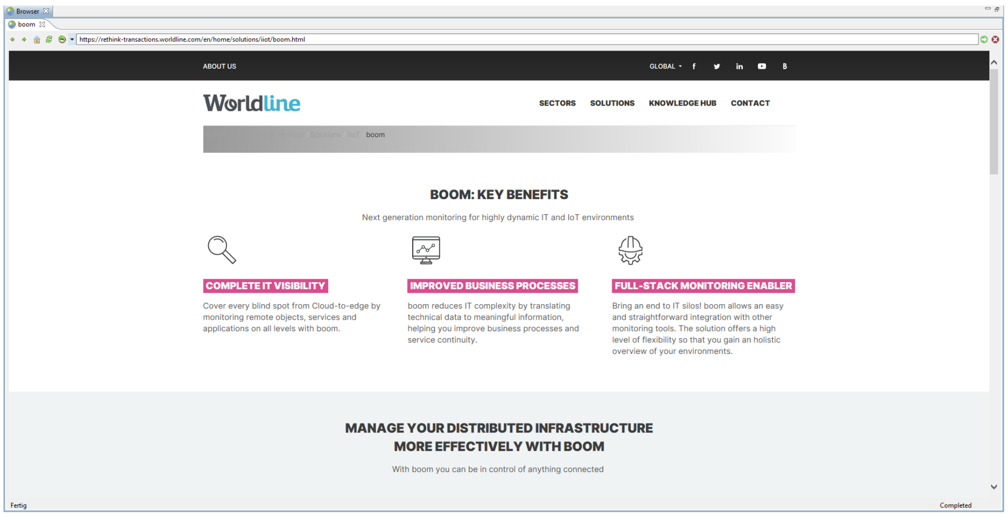
ID	Name	E.	Enabled by	Disabled by	Set Severity	Set Host
b..	Oracle	✓	User:dak			
b..	ExternalHost	⊗	User:dak	User:dak		<$AGENT_H

If an Agent is connected to a Slave Server the configured "Server Policies" on the Slave Server are processed, then the Slave server performs correlation and de-duplication steps before the indications are forwarded to the Master Server. The Master Server will perform it’s Server Policies and Maintenance Windows on the forwarded indications.

If no policies are configured on the Slave Server the configuration of the Master server is used. If "Server Policies" are available (and enabled) on both Master and Slave server the Slave "Server Policies" are processed before the Master "Server Policies". For the full processing chain, please refer to the chapter Indication Processing on the *boom* Server.

6.28. Browser

The "Browser" view provides an integrated WEB browser.



BOOM: KEY BENEFITS

Next generation monitoring for highly dynamic IT and IoT environments

COMPLETE IT VISIBILITY

Cover every blind spot from Cloud-to-edge by monitoring remote objects, services and applications on all levels with boom.

IMPROVED BUSINESS PROCESSES

boom reduces IT complexity by translating technical data to meaningful information, helping you improve business processes and service continuity.

FULL-STACK MONITORING ENABLER

Bring an end to IT silos! boom allows an easy and straightforward integration with other monitoring tools. The solution offers a high level of flexibility so that you gain an holistic overview of your environments.

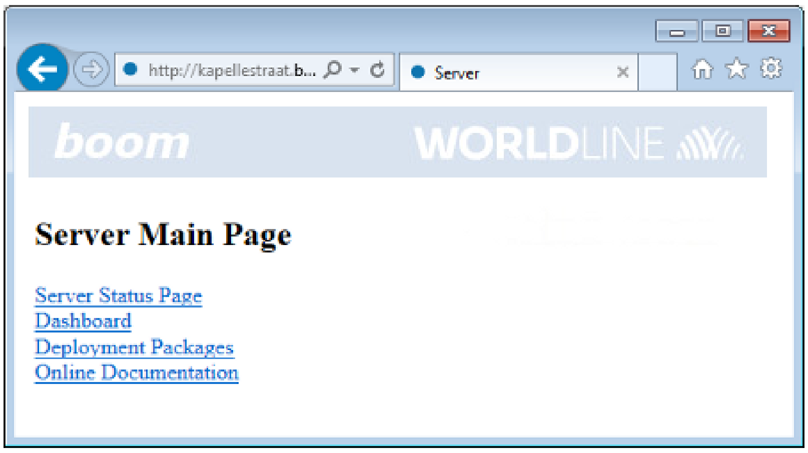
MANAGE YOUR DISTRIBUTED INFRASTRUCTURE
MORE EFFECTIVELY WITH BOOM

With boom you can be in control of anything connected

Chapter 7. *boom* Server Mainpage

When you login to the business open operations manager main page, you can access the following items:

- Server Status Page
- Dashboards
- Deployment Packages


Server Main Page

Server Status Page
Dashboard
Deployment Packages
Online Documentation

the web browser based View Active Indications (Read-only) is deprecated. Please us the new dashboards instead.

7.1. Web Browser Login

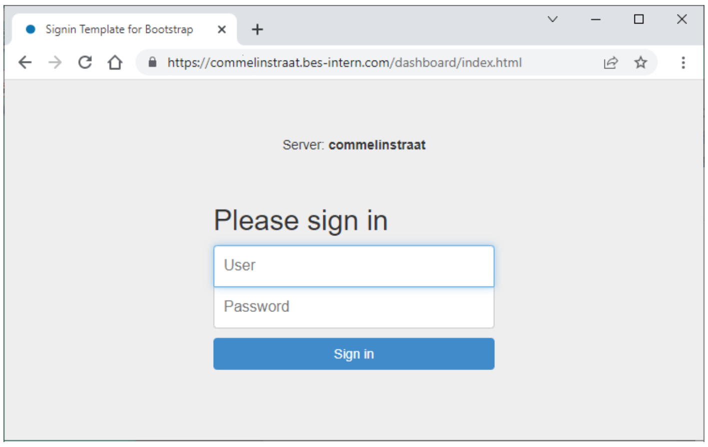
Open the following URL in a web browser in order to login to the *boom* server main page:

https://<management_server>

management_server
: Fully qualified hostname of the *boom* management server The port can be configured in the <boom_server_installdir>/boom.props file.

Enforced browser login to the 'boom server Status Page' and 'View Active Indications'

To successfully start the 'boom server status page' and 'View Active Indications' you are asked for login information.


Signin Template for Bootstrap
https://commelinstraat.bes-intern.com/dashboard/index.html
Server: commelinstraat
Please sign in
User
Password
Sign in

username
: Valid username. For a successful login the user has to exist in boom. The default user is admin (password: admin).

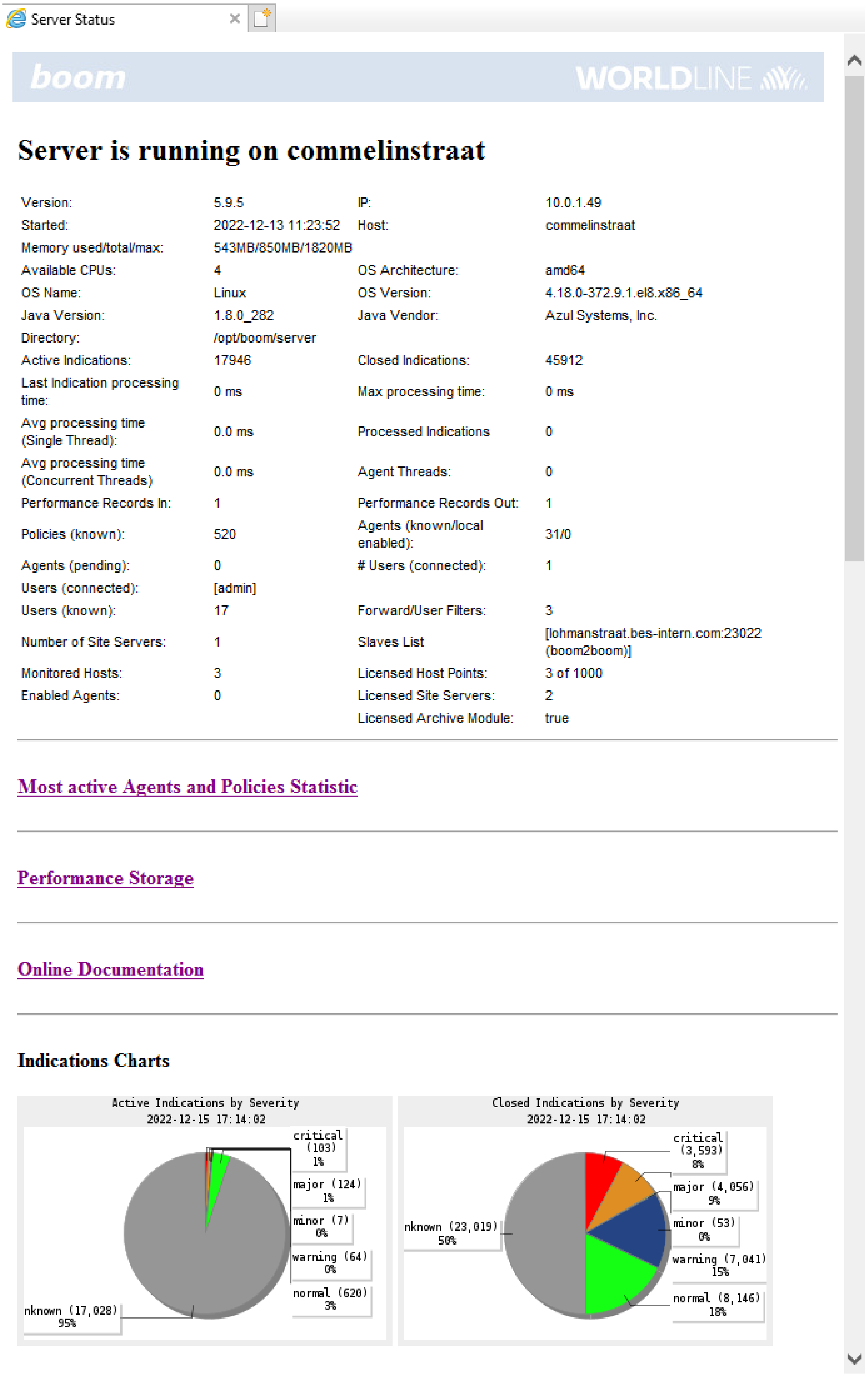
password
: password

See chapter User Management for more information on adding new users.

7.2. Server Status Page

In order to see the 'boom server status page' you have to login via web browser: Web Browser Login

The status page is split into multiple views which contain statistics information. The first view summarizes general information about the *boom* server. The further views contain performance data and various indication charts.



Server is running on commelinstraat

Version:	5.9.5	IP:	10.0.1.49
Started:	2022-12-13 11:23:52	Host:	commelinstraat
Memory used/total/max:	543MB/850MB/1820MB		
Available CPUs:	4	OS Architecture:	amd64
OS Name:	Linux	OS Version:	4.18.0-372.9.1.el8.x86_64
Java Version:	1.8.0_282	Java Vendor:	Azul Systems, Inc.
Directory:	/opt/boom/server		
Active Indications:	17946	Closed Indications:	45912
Last Indication processing time:	0 ms	Max processing time:	0 ms
Avg processing time (Single Thread):	0.0 ms	Processed Indications	0
Avg processing time (Concurrent Threads)	0.0 ms	Agent Threads:	0
Performance Records In:	1	Performance Records Out:	1
Policies (known):	520	Agents (known/local enabled):	31/0
Agents (pending):	0	# Users (connected):	1
Users (connected):	[admin]		
Users (known):	17	Forward/User Filters:	3
Number of Site Servers:	1	Slaves List	[lohmanstraat.bes-intern.com:23022 (boom2boom)]
Monitored Hosts:	3	Licensed Host Points:	3 of 1000
Enabled Agents:	0	Licensed Site Servers:	2
		Licensed Archive Module:	true

Most active Agents and Policies Statistic

Performance Storage

Online Documentation

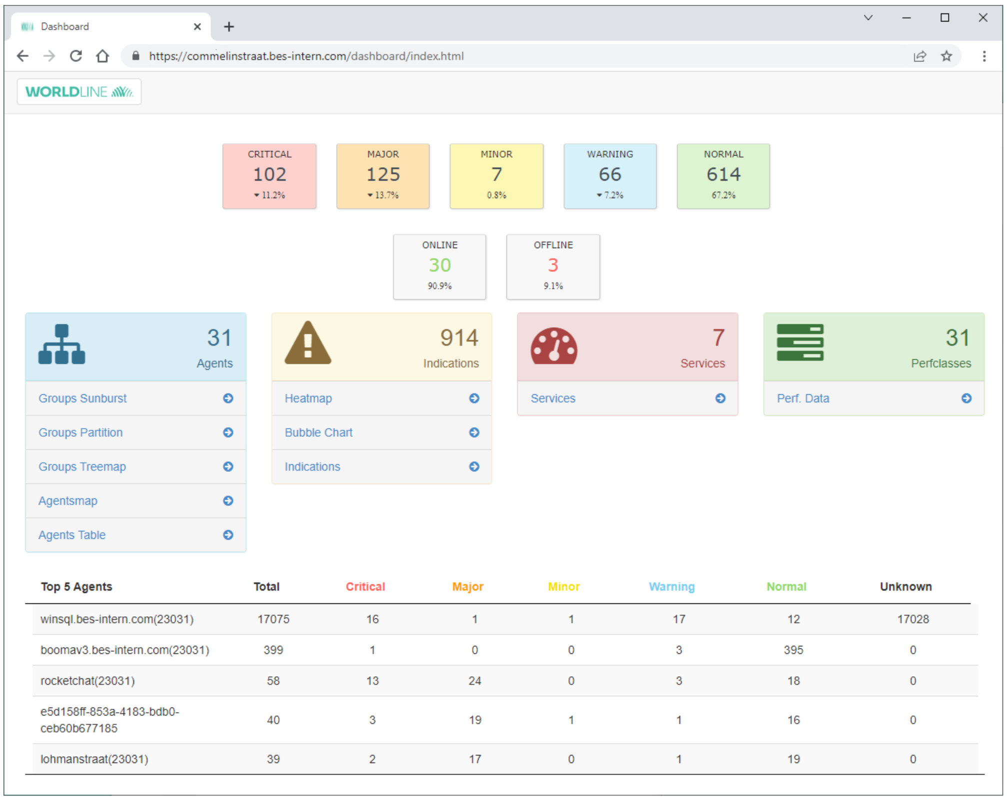
Indications Charts

See also chapter Statistics for more information and examples.

7.3. Dashboards

The web based dashboards visualize various aspects of the current state of the monitored environment. The available information reaches from the top level service overview to the details of an indication serving the different needs of the different types of users. The dashboards are implemented to support a wide range of devices from PC to mobile devices like tablets and smart phones with different operating systems like Android or IOS.

The dashboards adhere to the *boom* user role model. A user will only see the hosts and indications that are defined for his user role.

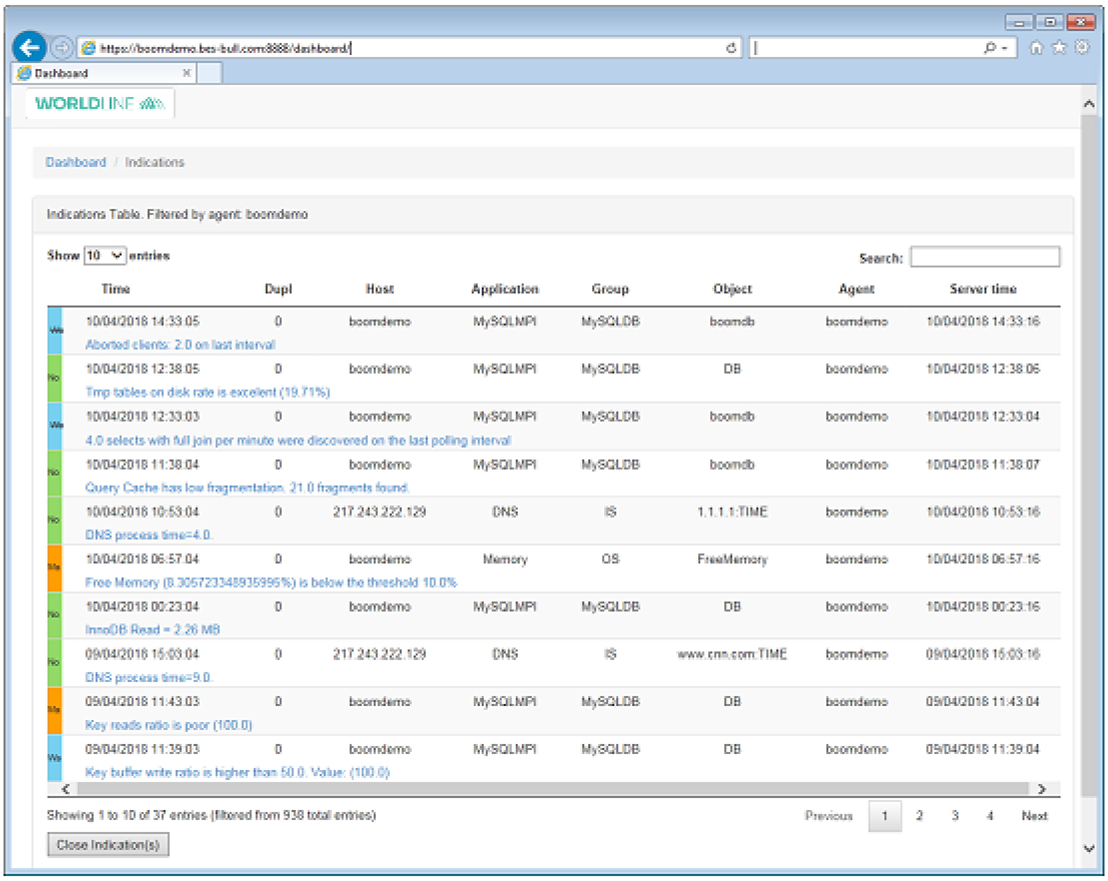


Top 5 Agents	Total	Critical	Major	Minor	Warning	Normal	Unknown
winsql.bes-intern.com(23031)	17075	16	1	1	17	12	17028
boomav3.bes-intern.com(23031)	399	1	0	0	3	395	0
rocketchat(23031)	58	13	24	0	3	18	0
e5d158ff-853a-4183-bdb0-ceb60b677185	40	3	19	1	1	16	0
lohmanstraat(23031)	39	2	17	0	1	19	0

The first dashboard presents an overview about the total number of indications per severity level, the number of hosts that are on/offline, the navigation area to the other dashboard views and the a list of the top 5 agents (based on total number of indications).

The navigation area contains in the left column dashboards related to agents, the next column contains dashboards based related to indications, the next column links to the service dashboard and the right column provides access to the collected history performance data.

A click (or double tick/click depending on the device) on the severity summary or agent in the table will open up a filtered indication view. This type of drill down navigation is available in most dashboards.



https://boomdemo.bes-bull.com:8888/dashboard/

Dashboard / Indications

Indications Table. Filtered by agent: boomdemo

Show 10 entries   Search:

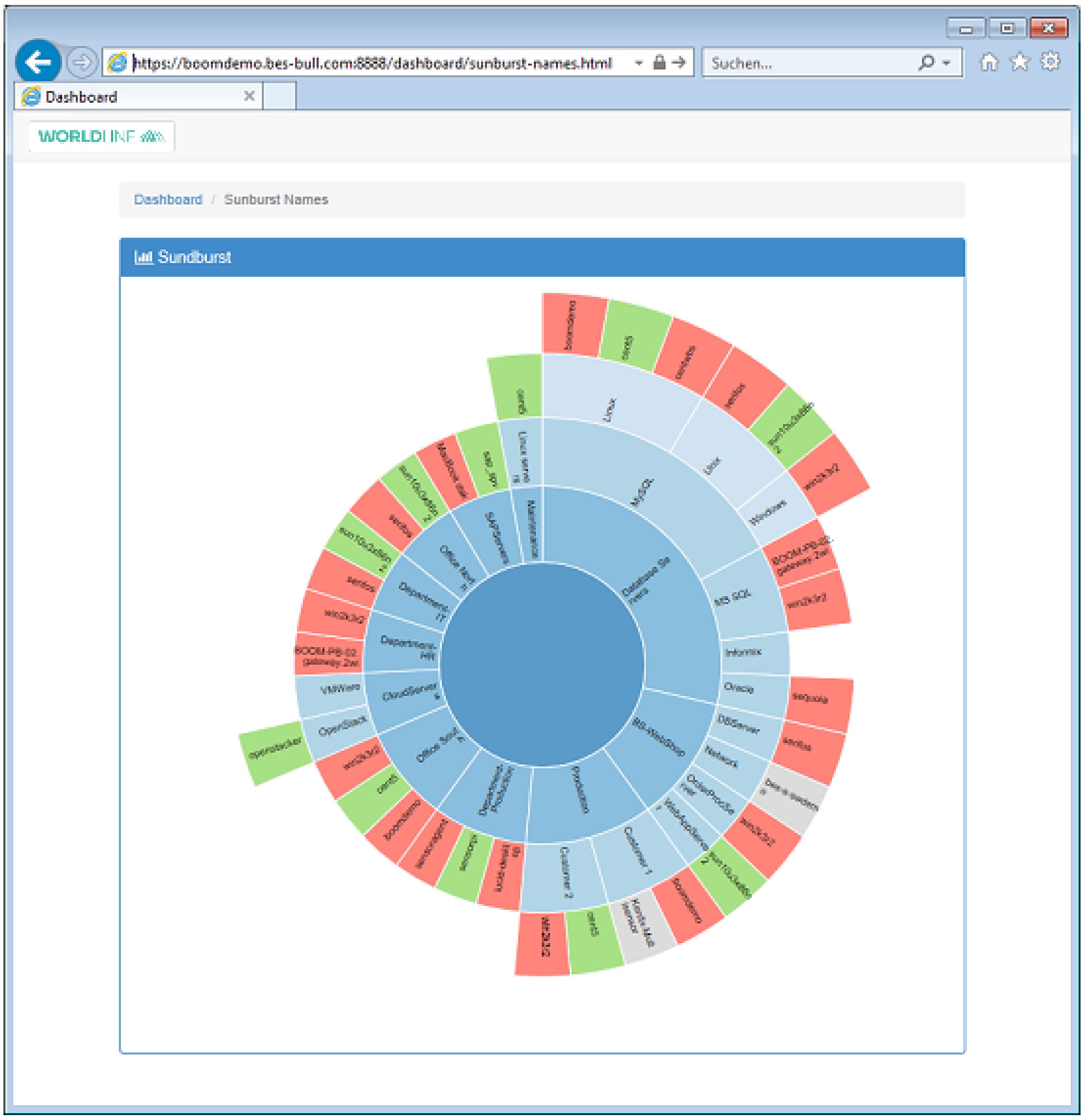
	Time	Dupl	Host	Application	Group	Object	Agent	Server time
We	10/04/2018 14:33:05	0	boomdemo	MySQLMPI	MySQLDB	boomdb	boomdemo	10/04/2018 14:33:16
	Aborted clients: 2.0 on last interval							
No	10/04/2018 12:38:05	0	boomdemo	MySQLMPI	MySQLDB	DB	boomdemo	10/04/2018 12:38:06
	Tmp tables on disk rate is excelent (19.71%)							
We	10/04/2018 12:33:03	0	boomdemo	MySQLMPI	MySQLDB	boomdb	boomdemo	10/04/2018 12:33:04
	4.0 selects with full join per minute were discovered on the last polling interval							
No	10/04/2018 11:38:04	0	boomdemo	MySQLMPI	MySQLDB	boomdb	boomdemo	10/04/2018 11:38:07
	Query Cache has low fragmentation. 21.0 fragments found.							
No	10/04/2018 10:53:04	0	217.243.222.129	DNS	IS	1.1.1.1:TIME	boomdemo	10/04/2018 10:53:16
	DNS process time=4.0.							
Ma	10/04/2018 06:57:04	0	boomdemo	Memory	OS	FreeMemory	boomdemo	10/04/2018 06:57:16
	Free Memory (8.305723348935995%) is below the threshold 10.0%							
No	10/04/2018 00:23:04	0	boomdemo	MySQLMPI	MySQLDB	DB	boomdemo	10/04/2018 00:23:16
	InnoDB Read = 2.26 MB							
No	09/04/2018 15:03:04	0	217.243.222.129	DNS	IS	www.cnn.com:TIME	boomdemo	09/04/2018 15:03:16
	DNS process time=9.0.							
Ma	09/04/2018 11:43:03	0	boomdemo	MySQLMPI	MySQLDB	DB	boomdemo	09/04/2018 11:43:04
	Key reads ratio is poor (100.0)							
We	09/04/2018 11:39:03	0	boomdemo	MySQLMPI	MySQLDB	DB	boomdemo	09/04/2018 11:39:04
	Key buffer write ratio is higher than 50.0. Value: (100.0)							

Showing 1 to 10 of 37 entries (filtered from 938 total entries)   Previous 1 2 3 4 Next

Close Indication(s)

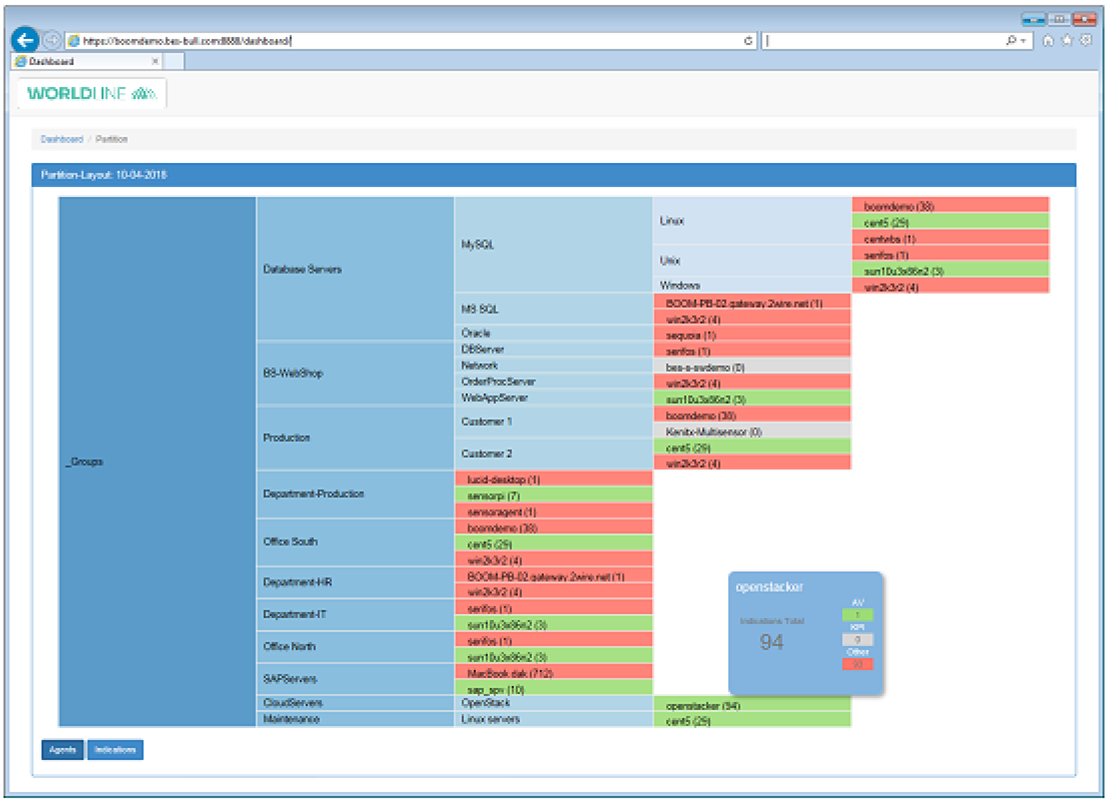
In the indication browser views it is possible to select indications and close them with the **Close Indication(s)** button. The indication details an be viewed by clicking on the indication text.

The agent related dashboards provide different representations of the host group structure, agents and the current severity level of each agent. The size of the different areas represent either the number of agents or the number of indications that are currently active. On some dashboards a selection on the bottom allows to switch between the sizing by number of agents or indications. Some samples of the dashboards are shown below.

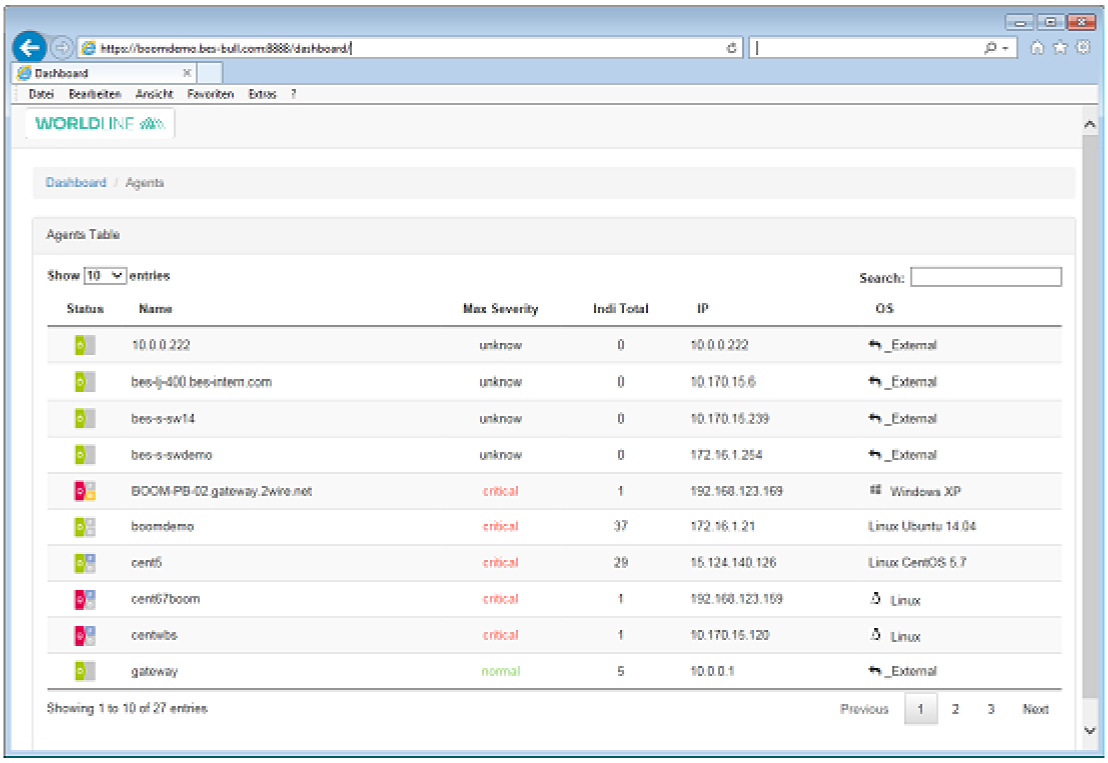


Hovering with the mouse over an item in the dashboard will display details like the actual number of indications.

On mobile devices a single tip on the item will open the pop up. On IOs the pop up will have an X at the top right corner to close the pop up, while on other devices a second tip will close it.



The Agent Table shows the agent status and system information for all agents. The symbol encodes the online/offline state, firewalled and disabled status.


https://boomdemo.bes-bull.com:8888/dashboard/
Dashboard
Datei Bearbeiten Ansicht Favoriten Extras ?
WORLDLINE
Dashboard / Agents

Agents Table

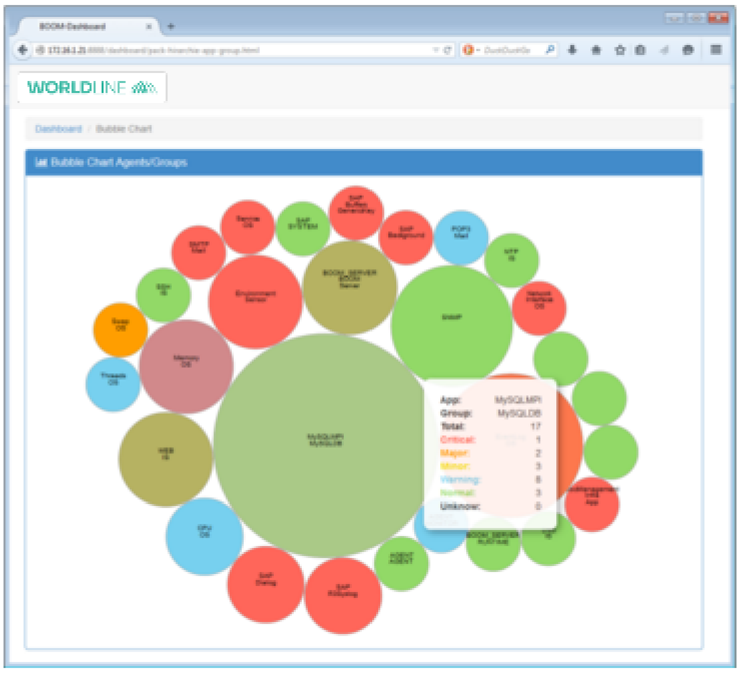
Show 10 entries

Search:

Status	Name	Max Severity	Indi Total	IP	OS
🟩	10.0.0.222	unknow	0	10.0.0.222	_External
🟩	bes-lj-400.bes-intern.com	unknow	0	10.170.15.6	_External
🟩	bes-s-sw14	unknow	0	10.170.15.239	_External
🟩	bes-s-swdemo	unknow	0	172.16.1.254	_External
🟥	BOOM-PB-02.gateway.2wire.net	critical	1	192.168.123.169	Windows XP
🟩	boomdemo	critical	37	172.16.1.21	Linux Ubuntu 14.04
🟩	cent5	critical	29	15.124.140.126	Linux CentOS 5.7
🟥	cent67boom	critical	1	192.168.123.159	Linux
🟥	centwbs	critical	1	10.170.15.120	Linux
🟩	gateway	normal	5	10.0.0.1	_External

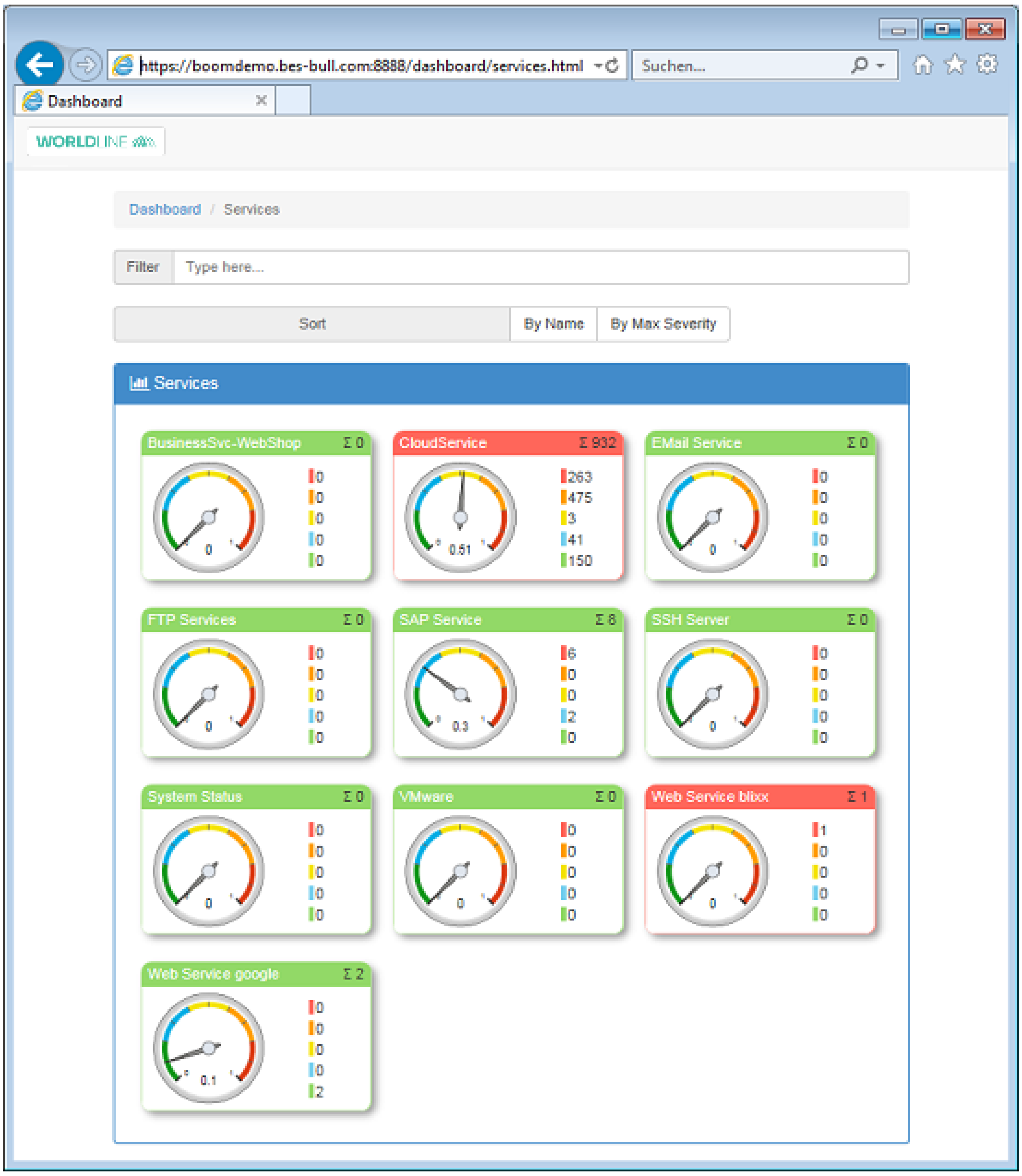
Showing 1 to 10 of 27 entries

Previous 1 2 3 Next

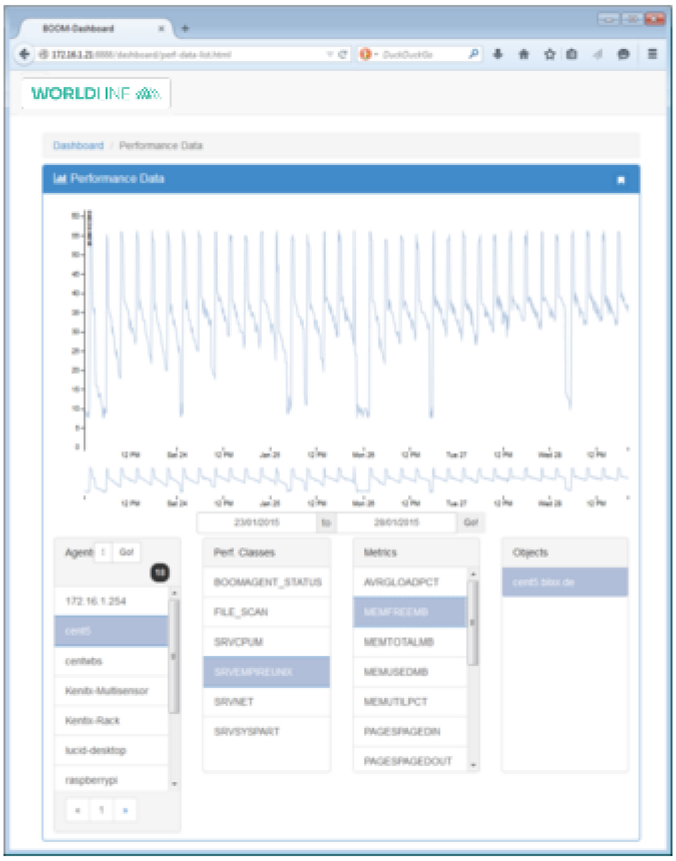


```
App:        MySQLMPI
Group:      MySQLDB
Total:            17
Critical:          1
Major:             2
Minor:             3
Warning:           8
Normal:            3
Unknow:            0
```

The service dashboard corresponds to the dashboard view in the *boom* java UI. It represents the availability state by the color of the service frame, the KPI state is represented by the gauge, the severity counts represent the number of indications without service impact.



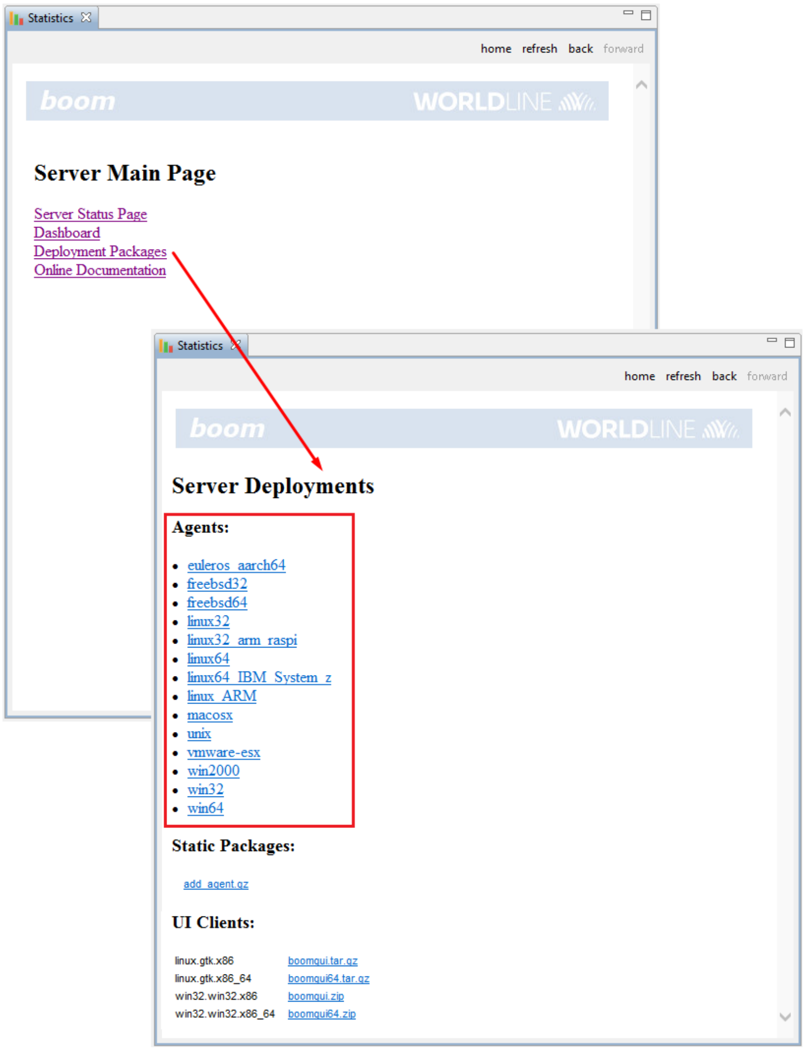
The performance data dashboard allows to select agent, performance class and metric that should be displayed. The displayed time range can be adjusted with the time slider below the chart.



7.4. Deployment Packages

The *boom* agent and client packages reside on the server and can be downloaded to the *boom* agent and client systems. The server will prepare the agent configuration files and generate the packages when a download request is received.

See also chapter Installation for more information on the *boom* client and agent installation.



Server Main Page

Server Status Page
Dashboard
Deployment Packages
Online Documentation

Server Deployments

Agents:

- euleros_aarch64
- freebsd32
- freebsd64
- linux32
- linux32_arm_raspi
- linux64
- linux64_IBM_System_z
- linux_ARM
- macosx
- unix
- vmware-esx
- win2000
- win32
- win64

Static Packages:

add_agent.gz

UI Clients:

linux.gtk.x86	boomgui.tar.gz
linux.gtk.x86_64	boomgui64.tar.gz
win32.win32.x86	boomgui.zip
win32.win32.x86_64	boomgui64.zip

Chapter 8. *boom* Administration

8.1. *boom* Server Administration

8.1.1. Starting and Stopping the *boom* Server

On Unix systems:

The *boom* server runs as daemon and is managed by the init scripts. To start or stop the server manually, enter the following command:

```
/etc/init.d/boom_srv start | stop | status
```

On Windows systems:

The *boom* server runs as **Windows service** named **BoomServer** and is managed by the service control. The Windows service controls the server's java process. If the Windows service shows the status **Paused** it actually means that the server's java process is not running and *boom* is not available. In this case first stop the Windows service and start it again.

The java process might take some milliseconds to start up. On some systems this causes an error message stating that the BoomServer Service failed to start and the service is displayed as stopped for a short time period. This message can be ignored. To verify if the service started successfully, wait a second and then press Refresh in the service control. After the refresh the service will be displayed with the state **Started**.

8.1.2. Checking the *boom* Server Process

On Unix systems:

To check the *boom* Server process, enter the following command:

```
# ps -eo args| grep BOOMSERVER

    <java17_repository>/bin/java -DBOOMSERVERPROC -Xmx2048M -Dfile.encoding=UTF8
-Dsun.net.inetaddr.ttl=300 -Djava.awt.headless=true -Dlog4j2.formatMsgNoLookups=true --add-modules=ALL
-SYSTEM --add-exports java.base/sun.security.pkcs10=boom.server --add-exports
java.base/sun.security.x509=boom.server --add-exports java.base/sun.security.pkcs=boom.server --add
-exports java.base/sun.security.util=boom.server -p srv/libs -m boom.server/com.blixx.server.ServerEngine
```

The parameters limit the Java process heap to 2048 MB memory (Xmx) and restrict the TTL of the Java internal DNS cache to 300 seconds. For production systems the Java heap size can be increased e.g. to 3GB "-Xmx3072M". The startup parameters can be changed in the startup configuration file `<boom_server_installdir>/boom_srv.cfg`.

To check the status of the *boom* Server, enter the following command:

```
/etc/init.d/boom_srv status
```

On Windows systems:

The status of the *boom* server is displayed in the service control as service **BoomServer**.
Also the server's java process can be checked. The process identification (ID) of this process is stored in the file `boom_server.pid`, taskmanager or the command `tasklist` can be used to check if a java process with the process ID is running.

8.1.3. *boom* Server Configuration Files

The *boom* server configuration file contains setup and configuration information for the server. The file is located in:

```
<boom_server_installdir>/boom.props
```

Configuration Parameters:

Parameter	Default	Description
AGENT_PORT	23021	Defines the default port on which the agent will listen. This parameter is used to generate the default *boom* Agent Configuration File when the agent deployment package is downloaded from the embedded HTTP server.
AGENT_TS_FILE	-	Defines location of agent's "truststore" file.
AGENT_TS_PASS_ENCODED	-	Encrypted password for reading agent's "truststore" file.
AUDITENABLE	false	(true|false) Set this flag to true to enable Auditing.
AUDITLOGCOUNT	5	Sets the maximum number of audit logfiles to e.g. 5.
AUDITLOGDIR	.	Directory for audit logfiles, server installation directory by default.
AUDITLOGSIZE	10	Sets the maximum size per audit logfile to e.g. 10 MB.
AUTO_APPROVAL	false	(true|false) If this flag is set to true the *boom* server will automatically approve new *boom* agents and skip the manual approval process.
AUTO_ARCHIVE_DAYS	-	(don't change here, use server job configuration instead!) Server will archive closed indications automatically after specified amount of days.
AUTO_CLOSE_DAYS	-	(don't change here, use server job configuration instead!) Server will close active indications automatically after specified amount of days.
AUTO_DELETE_DAYS	-	(don't change here, use server job configuration instead!) Server will delete archived indications automatically after specified amount of days.
AUTOCLOSE_FINISHED_ALERTS	false	(true|false) If set to true an indication that gets the finished alert status will be automatically closed.
AUTODETECT_EXTERNAL_HOSTS	true	(true|false) The *boom* server creates virtual Agent cards based on incoming values in the host attribute of the indications coming from connected agents.
AUTODETECT_EXTERNAL_HOSTS_FROM_SLAVES	false	(true|false) The *boom* server creates virtual Agent cards based on incoming values in the host attribute of the indications coming from connected slave servers.
BB	-	Licensed Site Servers "boom to boom" (Slave, Backup) key.

Parameter	Default	Description
CLUSTER_NODES	-	A list of hostnames or IP addresses of all *boom* servers that might need to communicate with the agent. All specified servers can perform remote actions and initiate heartbeats on the agent. This parameter is used to generate the default *boom* Agent Configuration File when the agent deployment package is downloaded from the embedded HTTP server.
CSR_STORAGE_DIR	./srv/etc/csrcache	Storage location for certificate signing requests.
CSR_TTL_MS	604800000	Time to live for certificate signing requests in milliseconds.
FAILED_LOGINS_COUNT_RETENTION_MS	60000	Retention of failed logins in milliseconds.
FW_AGENTS_OFFLINE_TIMEOUT	60000	Firewalled agent offline timeout in milliseconds.
GUI_PORT	23022	Defines the port on which the Server will listen for UI clients.
HB_CTIMEOUT	10000	Socket connect timeout in milliseconds.
HB_INTERVAL	10	Heartbeat interval in seconds.
HB_RTIMEOUT	10000	Socket read timeout in milliseconds.
HTTP_PORT	8888	Defines the port for the embedded HTTP server. In case of a change of this port all active GUI sessions have to be restarted.
HOSTNAMES_LOWERCASE	false	(true|false) Set this flag to automatically convert the data in the indication host field to lowercase.
IGNORE_PERF_DATA	false	(true|false) If this flag is set to true the *boom* server will ignore all performance data that is submitted by agents and would be otherwise stored in the performance database. Note: this has no impact on the threshold monitoring or indication processing.
INSTRUCTION_SERVER	-	Defines instruction server URL (HTTP/HTTPS) for instructions defined in the policies. The variable <$INSTRUCTION_SERVER> will be replaces with given string for each incoming indication if defined in the Instruction URL field (See Policies).
KS_FILE	-	Defines location of "keystore" file.
KS_PASS_ENCODED	-	Encrypted password for reading "keystore" file.
LISTEN_IP	-	This parameter should be set with caution. If this parameter is set to a valid IP address of the server system, the *boom* server will listen only on this address and ignore any other available IP cards and addresses.
LH	-	Licensed Host Point key.
LOGDIR	.	Directory for logfiles of the server.

Parameter	Default	Description
LOGLEVEL	1	Loglevel of the server 1 to 5 (1-errors only ... 5 support debug)
MAX_DEPLOY_THREADS	500	Max. number of threads for deploy operation.
MAIN_SERVER_IP	-	Normally this field does not need to be set since the server will auto detect its hostname and IP address. In some cases this might not be possible, e.g. due to virtual names or multihomed servers. Setting the IP of the server with this parameter will reflect in the contents of the agent configuration settings that are contained in the agent installation packages.
MAIN_SERVER_NAME	-	Normally this field does not need to be set since the server will auto detect its hostname and IP address. In some cases this might not be possible, e.g. due to virtual names or multihomed servers. Setting the name of the server with this parameter will reflect in the contents of the agent configuration settings that are contained in the agent installation packages.
MAIN_SERVER_PORT	23020	Defines the port on which the *boom* server listens for agent connections. This parameter is used to generate the default *boom* Agent Configuration File when the agent deployment package is downloaded from the embedded HTTP server.
NUM_FAILED_LOGINS_BEFORE_USER_LOCK	9	Max. number of failed logins before user locked.
PERF_TABLE_MAX_SIZE_MB	50	The *boom* server will send a "major" indication if one of the performance tables exceeds the specified size (in MB). The server will continue to log the performance data, but some of the history data should be removed.
PKI_CLASS	com.blixx.server.pki.FileBasedPKI	File based PKI (Public Key Infrastructure) Java class.
PKI_CONFIG_FILE	file_pki.props	File based PKI (Public Key Infrastructure) configuration file.
SHORT_LABELS	false	(true|false) If this flag is set to true the *boom* Server will automatically use the short host name (without the domain qualifier) as host label. The labels can be manually changed.
SSL_KEYSTORE, SSL_PASSWORD, SSL_KEYPASS	-	*deprecated*
TS_FILE	srv/etc/truststore.p12	Defines location of "truststore" file.
TS_PASS_ENCODED	-	Encrypted password for reading "truststore" file.

Parameter	Default	Description
USE_LAST_DUPLICATE_TEXT	false	(true|false) If this flag is set to true the *boom* Server will present duplicated indication in modified way that instead of indication contains text of first indication it includes text of last duplicate. This works in conjunction with "De-Duplicate KeyOnly" (assuming Indication Key not covering Text attribute).
USER_LOCK_RETENTION_MS	600000	Retention of locked user in milliseconds.
VERSION	5.10.0	boom Server version.

Startup Parameters:

The configuration file <boom_server_installdir>/boom_srv.cfg is used to overwrite some java startup parameters and the path to the java runtime. Any changes e.g. to the server memory setting should be done in this file instead of changing the boom_srv script directly, since any customizations done to boom_srv might be lost otherwise during the next upgrade.

Allowed entries:

JAVA_OPTS	Change any Parameter for Java in the line with JAVA_OPTS
	Default:
	`JAVA_OPTS="-Xmx1300M -Dfile.encoding=UTF8 -Dsun.net.inetaddr.ttl=300 -Djdk.tls.ephemeralDHKeySize=2048"`
	Example (increasing heap size/memory 4GB):
	`JAVA_OPTS="-Xmx4906M -Dfile.encoding=UTF8 -Dsun.net.inetaddr.ttl=300 -Djdk.tls.ephemeralDHKeySize=2048"`

JAVA_BIN	Optional parameter which is specifying the Java that should be used.
	Example Linux:
	`JAVA_BIN="/path/to/java/bin"`
	Example Windows:
	`JAVA_BIN="C:\path\to\java\bin"`

Please note: values must be double-quoted!

8.1.4. *boom* Database Configuration Files

The *boom* server allows you to configure one single database for fault management and performance data or to configure a separate database only for performance data.

The *boom* database configuration file db.props contains information which is necessary for the connection to the database. All fault management data are written to this database. The file is located in:

```
<boom_server_installdir>/db.props
```

Configuration parameters:

Parameter	Default	Description
DbName	-	Defines the name of the database (*boom*)
Driver	Oracle.jdbc.OracleDriver	Defines the driver for the *boom* DB connection.
EnPass	-	Defines the encrypted database password.

Parameter	Default	Description
Host	-	Defines the FQDN of the database host.
JDBC	-	Java Database Connection string: can be used instead of "SID"; syntax example connecting to a 2 node Oracle system (DBserver1, DBserver2): JDBC=jdbc\:oracle\:thin\:@(DESCRIPTION\=(CONNECT_TIMEOUT\=10)(RETRY_COUNT\=3)(ADDRESS_LIST\=(ADDRESS\=(PROTOCOL\=TCP)(HOST\=<DBserver1>)(PORT\=<port>))(ADDRESS\=(PROTOCOL\=TCP)(HOST\=<DBserver2>)(PORT\=<port>)))(CONNECT_DATA\=(SERVICE_NAME\=<DB_sid>)(SERVER\=DEDICATED)))
Login	-	Login name of the database user
Password	-	Password field for encryption
Port	-	Listener port
SID	-	Oracle system ID

The *boom* server allows you to setup a separate database for performance data. The performance database configuration file db_perf.props contains information which is necessary for the connection to the performance database and can be configured separately from the main db.props file. All performance data are written to this database. Both db.props and db_perf.props can point to the same database. The file is located in:

```
<boom_server_installdir>/db_perf.props
```

Configuration parameters:

Parameter	Default	Description
DbName	-	Defines the name of the database (*boom*)
Driver	Oracle.jdbc.OracleDriver	Defines the driver for the *boom* DB connection.
EnPass	-	Defines the encrypted database password.
Host	-	Defines the FQDN of the database host.
JDBC	-	Java Database Connection string: can be used instead of "SID"; syntax example connecting to a 2 node Oracle system (DBserver1, DBserver2): JDBC=jdbc\:oracle\:thin\:@(DESCRIPTION\=(CONNECT_TIMEOUT\=10)(RETRY_COUNT\=3)(ADDRESS_LIST\=(ADDRESS\=(PROTOCOL\=TCP)(HOST\=<DBserver1>)(PORT\=<port>))(ADDRESS\=(PROTOCOL\=TCP)(HOST\=<DBserver2>)(PORT\=<port>)))(CONNECT_DATA\=(SERVICE_NAME\=<DB_sid>)(SERVER\=DEDICATED)))
Login	-	Login name of the database user
Password	-	Password field for encryption
Port	-	Listener port
SID	-	Oracle system ID

8.1.5. *boom* Server Logfiles

The *boom* server writes information according to the set of the loglevel into a logfile.

LOGLEVEL:

Set the server LOGLEVEL via the GUI action "Set Server Log Level". In this case no server restart is required. You can also edit the boom.props file. In this case a restart is required (stop the server before changing the file). The LOGLEVEL can be set from 1 to 5 (1-errors only … 5 support debug).

Logfile:

Every time the *boom* server starts the following log file is created: /opt/boom/server/BOOMServer_<date>_#.log The logfiles are rotated very day. Logfiles older than 8 days are deleted.

8.1.6. *boom* Agent Administration

8.1.6.1. Starting and Stopping the *boom* Agent

On Unix systems:

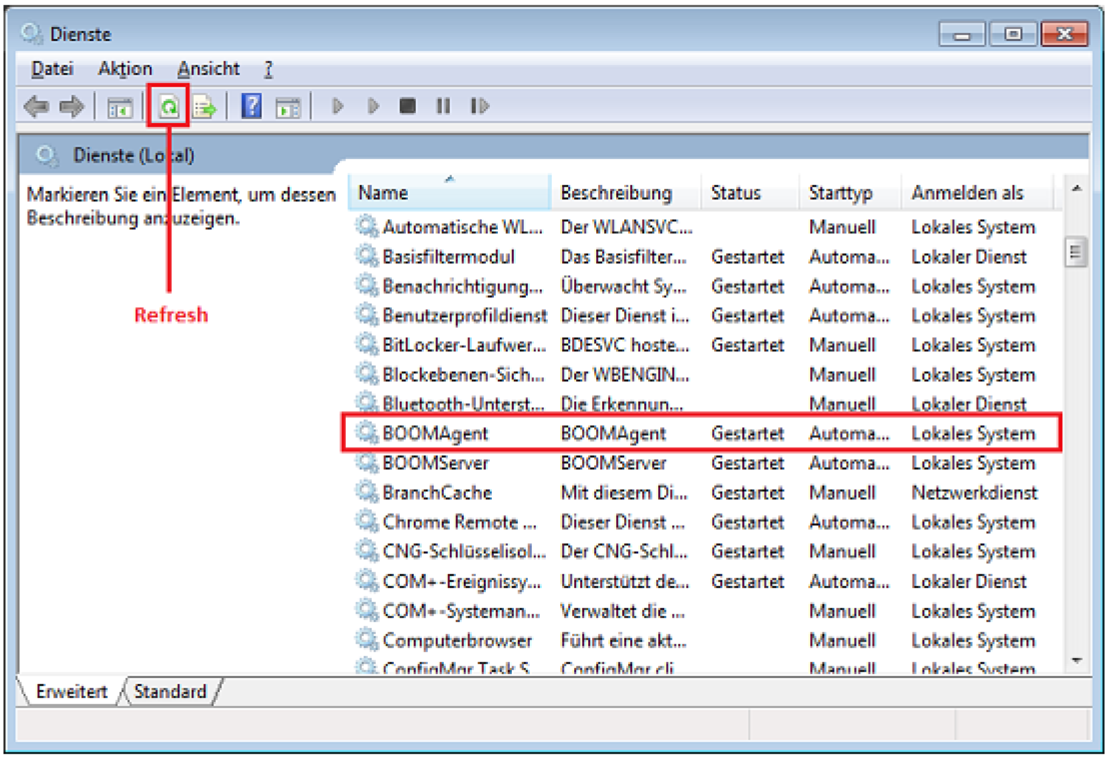
The *boom* agent runs as a java process and is managed by the init scripts. To start, restart or stop the agent manually, enter the following command:

```
/etc/init.d/boom_agt start | restart | stop | status
```

On Windows systems:

The *boom* agent runs as **Windows servic**e named **BoomAgent** and is managed by the service control. The Windows service controls the agent’s java process. If the Windows service shows the status **Paused** it actually means that the agent’s java process is not running and the *boom* agent is not available. In this case first stop the Windows service and start it again.

Note: The java process might take some milliseconds to start up. On some systems this causes an error message stating that the BoomAgent Service failed to start and the service is displayed as stopped for a short time period. This message can be ignored. To verify if the service started successfully, wait a second and then press Refresh in the service control. After the refresh the service will be displayed with the state **Started**.



Markieren Sie ein Element, um dessen Beschreibung anzuzeigen.

Name	Beschreibung	Status	Starttyp	Anmelden als
Automatische WL...	Der WLANSVC...		Manuell	Lokales System
Basisfiltermodul	Das Basisfilter...	Gestartet	Automa...	Lokaler Dienst
Benachrichtigung...	Überwacht Sy...	Gestartet	Automa...	Lokales System
Benutzerprofildienst	Dieser Dienst i...	Gestartet	Automa...	Lokales System
BitLocker-Laufwer...	BDESVC hoste...	Gestartet	Manuell	Lokales System
Blockebenen-Sich...	Der WBENGIN...		Manuell	Lokales System
Bluetooth-Unterst...	Die Erkennun...		Manuell	Lokaler Dienst
BOOMAgent	BOOMAgent	Gestartet	Automa...	Lokales System
BOOMServer	BOOMServer	Gestartet	Automa...	Lokales System
BranchCache	Mit diesem Di...	Gestartet	Manuell	Netzwerkdienst
Chrome Remote ...	Dieser Dienst ...	Gestartet	Automa...	Lokales System
CNG-Schlüsselisol...	Der CNG-Schl...	Gestartet	Manuell	Lokales System
COM+-Ereignissy...	Unterstützt de...	Gestartet	Automa...	Lokaler Dienst
COM+-Systeman...	Verwaltet die ...		Manuell	Lokales System
Computerbrowser	Führt eine akt...		Manuell	Lokales System
ConfigMgr Task S...	ConfigMgr cli...		Manuell	Lokales System

8.1.6.2. Checking the *boom* Agent Process

On Unix systems:

To check the *boom* Agent process, enter the following command:

```
# ps -eo args| grep BOOMAGENT

    java -DBOOMAGENTPROC -Xmx512M -jar boom_agent.jar
```

The parameters limit the Java process heap to 512 MB memory (Xmx). The Java heap size can be increased e.g. to 1GB "-Xmx1024M". The startup parameters can be changed in the startup configuration file `<boom_agent_installdir>/boom_agt.cfg`.

To check the status of the *boom* Agent, enter the following command:

```
/etc/init.d/boom_agt status
```

On Windows systems:

The status of the *boom* Agent is displayed in the service control as service **BoomAgent**.

Also the agent's java process can be checked. The process identification (ID) of this process is stored in the file `boom_agent.pid`, taskmanager or the command tasklist can be used to check if a java process with the process ID is running.

8.1.6.3. *boom* Agent Configuration File

The install script creates the *boom* agent configuration file automatically. It contains setup and configuration information for the agent. The file is located in:

```
<boom_agent_installdir>/conf/agent.conf
```

!	Please stop the *boom* Agent before modifying the configuration file. The agent overwrites any changes during shutdown. It's recommended to use the integrated GUI BOOM_AGENT Action to modify the configuration.

Configuration parameters:

Parameter	Default	Description
ACTIONS_ENABLED	true	(true|false) This flag is used to control remote actions. Set the flag to FALSE to reject all remote actions, auto-actions and operator initiated actions. Internal action won't be disabled e.g. BOOM_AGENT GET_ID. Manual steps are necessary to enable remote actions again: 1. stop the agent 2. set ACTIONS_ENABLED=true in agent.conf or remove the parameter 3. start the agent
AGENT_HOST	auto|manual	Defines the agents hostname. It will be determined automatically unless the FIXED_NAME parameter is set to true.
AGENT_ID	auto	Do not change this field manually. It is created automatically during first clean start of the *boom* agent.
AGENT_IP	auto|manual	Defines the agents IP address that it will report back to the server. It will be determined automatically unless the FIXED_IP parameter is set to true.
AGENT_PORT	23021	Defines the port on which the agent will communicate essentially.
AGENT_TLS_PORT	23021	Defines the TLS (Transport Layor Security) port on which the agent will communicate.
BACKUP_SERVER_IP	manual	The *boom* Backup server IP address. This parameter has to be set (srv/deploy/agent/agent.conf) before agent package will be created.
BACKUP_SERVER_NAME	manual	The *boom* Backup server hostname. This parameter has to be set (srv/deploy/agent/agent.conf) before agent package will be created.
BACKUP_SERVER_PORT	23020	Defines the port on which the *boom* Backup server listens. This parameter has to be set (srv/deploy/agent/agent.conf) before agent package will be created.
CLONING_SUPPORT	false	*deprecated*

Parameter	Default	Description
CLUSTER_NODES	-	A list of hostnames or IP addresses of all *boom* servers that might need to communicate with the agent. All specified servers can perform remote actions and initiate heartbeats on the agent. + If the *boom* Server is installed on a cluster, the active cluster node might use it’s physical IP instead of the package/virtual/service IP to communicate with the agent, so all cluster node IPs should be listed. This list can be used to implement other concepts like central development or configuration servers.
COMM_TYPE	SOCKET	(SOCKET|TLS) If necessary, the agent can switch communication to use "TLS protocol" which is slower than default direct SOCKET.
DISABLED	false	(true|false) This parameter is set by the *boom* server if an agent needs to be disabled (e.g. for "System Maintenance"). In disabled mode the agent does not trigger any monitors and does not send any indications, but it will react on action requests from the server.
FIXED_IP	false	(true|false) This flag turns the automatic detection of the IP address off or on. This IP address is send to the server and defines how the server can reach the agent. + true - skips the detection of the IP address and use the IP that is specified in the AGENT_IP parameter. This can be used for multi-homed environments, where the agent might detect an internal IP, which the server can not reach. + false - the agent will automatically detect its IP address.
FIXED_NAME	false	(true|false) This flag turns the automatic detection of the agent hostname off or on. This hostname is send to the server and defines under which name the server can reach the agent. + true - skips the detection of the hostname and use the name that is specified in the AGENT_HOST parameter. This can be used for multi-homed environments, where the agent might detect an internal system name, which the server can not resolve. + false - the agent will automatically detect its hostname.
FT_TIMEOUT_MINS	-	*unused*
HB_CTIMEOUT	10000	Socket connect timeout in milliseconds.
HB_INTERVAL	10	Heartbeat interval in seconds.
HB_RECONNECT	-	*unused*
HB_RTIMEOUT	10000	Socket read timeout in milliseconds.
KS_FILE	-	Defines location of "keystore" file.
KS_PASS_ENCODED	-	Encrypted password for reading "keystore" file.
LOGCOUNT	5	Maximal number of logfiles.

Parameter	Default	Description
LOGDIR	.	Specify the logfile directory for the *boom* agent. Without this key the agent will create the logfiles in its installation directory. For agents on Windows systems use doubled backslashes and double quote around the path. I.e. (Windows) LOGDIR="C:\\temp\\boom Log Dir" (Unix) LOGDIR="/var/log/boom" (Note: directory must be writable for agent's user)
LOGLEVEL	1	Loglevel of the agent 1 to 5. (1-errors only ... 5 support debug)
LOGSIZE	10	Maximal size in MB of each logfile.
MAIN_SERVER_IP	auto|manual	The *boom* server IP address. The agent will determine the IP based on the server name, if it is not defined.
MAIN_SERVER_IP_MASK	-	*unused*
MAIN_SERVER_NAME	auto|manual	The *boom* server hostname. This parameter is set by the *boom* server when it creates the agent package.
MAIN_SERVER_PORT	23020	Defines the port on which the *boom* server listens. This parameter is set by the *boom* server when it creates the agent package. Use port 443 for HTTPS.
MAX_BUFFERED_MESSAGES	3000	Maximum number of indication or monitor values that can be buffered by agent when communication with server lost. This value sets the limit for each kind of indications separately. In other words: 3000 indications, 3000 monitor values can be buffered. When the buffer is filled, the oldest indications and values will be dropped.
MAX_SYM_MONITOR_CALLS	30	Maximum number of monitor calls that can be started parallel. It is not recommended to change this parameter.
MODE	0	(0|7) The "Mode Flag" is used to specify the firewall mode of the agent. 0 indicates that agent will actively communicate with the server. It will try to recognize if the server is online, offline or firewalled and submit data to the server. 7 switches the agent to a passive, listen only mode. The agent will not actively send any heartbeat or data to the server. The server will take the responsibility to send the heartbeats and poll the data from the agent. This mode is recommended to avoid unnecessary load on the firewall in environments where incoming traffic to the *boom* server is not allowed and must be blocked by a firewall.
MSD_BLOCK_MINUTES	5	Message Storm Detection's blocking period in minutes.
MSD_MAX_PER_MINUTE	1000	Message Storm Detection's maximum messages per minute.

Parameter	Default	Description
NO_DEPLOY_ALLOWED	false	(false|true) If set to "true" agent will reject any incoming deployments from server.
OFFER_DATA_TO_ANY_CLUSTER_NODE	false	(true|false) This flag allows the agent to offer data to any node from the cluster list (CLUSTER_NODES) that actually heartbeats the agent if the main server is not reachable from the agent (the agent can't send any data to the configured main server but the main server sends heartbeats to the agent).
PROTOCOL_VERSION	2	(0|2) Is used to specify the encryption method for agent communication. 0 means no encryption. Should be used if COMM_TYPE=TLS to avoid duplication of encryption effort. 2 switches to AES (Advanced Encryption Standard) based encryption. Should be used if COMM_TYPE=SOCKET.
PROXY_HOST, PROXY_PORT, PROXY_USER, PROXY_PASS	-	*deprecated*
SHARED_IP	false	(true|false) Activate sending the agent ID for NAT based environments. Setting this flag to true will instruct the agent to include its ID in all communication to the server, so that the server can differentiate the agents behind a NAT device. This does increase the network traffic, so it should only be turned on if necessary.
TS_FILE	conf/truststore.p12	Defines location of "truststore" file.
TS_PASS_ENCODED	-	Encrypted password for reading "truststore" file.
VER	-	*deprecated*
WS_PORT_AGENT	-	*unused*

Startup Parameters:

The configuration file `<boom_agent_installdir>/boom_agt.cfg` is used to overwrite some java startup parameters, the path to the java and/or perl runtime as well as the user account under which the agent process will run. Any changes e.g. to the agent memory setting should be done in this file instead of changing the boom_agt script directly, since any customizations done to the boom_agt script will be lost when a full agent upgrade or re-installation takes place.

Allowed entries:

JAVA_OPTS	Change any Parameter for Java in the line with JAVA_OPTS
	Default:
	`JAVA_OPTS="-Xmx512M"`
	Example (increasing heap size/memory 800MB:
	`JAVA_OPTS="-Xmx800M -server -Dfile.encoding=UTF8"`

JAVA_BIN	Optional parameter which specifying the Java that should be used.
	Example Linux:
	`JAVA_BIN="/path/to/java/bin"`
	Example Windows:
	`JAVA_BIN="C:\path\to\java\bin"`

PERL_PATH	Optional parameter which is specifying the PERL version that should be used. Example: `PERL_PATH="/opt/boom/bin"`
USER	Optional parameter which is specifying the user account under which the agent should run. During the startup the boom_agt script will do a switch user (su) command to start the java process with the specified user credentials. Example: `USER="boom"`

Please note: values must be double-quoted!

8.1.6.4. *boom* Agent Logfiles

LOGLEVEL:

Set the agent LOGLEVEL via the GUI action "Set Agent Log Level". In this case no agent restart is required.
You can also edit the agent.conf file. In this case a restart is required (stop the server before changing the file).
The LOGLEVEL can be set from 1 to 5 (1-errors only … 5 support debug).

Logfile:

Every time the *boom* agent starts the following log file is created:

```
/opt/boom/agent/BOOMAGENT_<date>_#.log
```

The logfiles rotate very day. Logfiles older than 8 days are deleted. At least one log file entry is written daily for agents running in loglevel=1 even if the agent is idle.

8.1.6.5. *boom* Agent Configuration on Cluster Nodes (Unix/Linux)

Cluster nodes might have additional virtual network interfaces and virtual names.

The *boom* server - agent communication is name based, but also checks the originating IP addresses.

Depending on OS and cluster software, the communication from agent to server might use a virtual interface, leading to an address update on the server side

Failover might cause confusion in the IP resolution if both cluster nodes are having agents

For cluster nodes set the following flags in the agent.conf

```
SHARED_IP=true      #indicates that multiple agents might share one address
FIXED_IP =true      #skip IP detection on agent and bind connection to the configured address
FIXED_NAME=true     #skip hostname detection on agent and report configured name to server
AGENT_HOST=abc.domain.com
AGENT_IP=10.168.10.199
CLUSTER_NODES=list of boom servers  #e.g. datac-boo01, datac-boom02, datac-boomvirtual all cluster nodes
(physical and virtual) to which the agent connects
```

8.1.6.6. *boom* Agents Connecting to a Clustered *boom* Server Environment

***boom* agents connecting to a clustered *boom* server environment**

For *boom* agents that are connecting to a *boom* server, which runs in a cluster environment, the following parameter in the agent.conf file have to be set:

```
CLUSTER_NODES=list of boom servers
```

e.g. datac-boo01, datac-boom02, datac-boomvirtual -→ all cluster nodes (physical and virtual) to which the agent connects

8.1.6.7. *boom* Agent Configuration in a SUN Solaris Global Zone

In a SUN Solaris global zone only the agent PID must be used to control the agent (e.g. stop agent). Please proceed the following steps to configure PID-only agent control:

1. Stop the agent: `/etc/init.d/boom_agt stop`
2. Set the variable PID_ONLY in the `/opt/boom/agent/boom_agt` control file:
   `PID_ONLY=true`
3. Start the agent: `/etc/init.d/boom_agt`

8.2. *boom* User Interface Administration

The GUI settings are stored in an **xml formatted** file. This configuration file can be a server side config file, that is stored on the server or it can be a local file, that is located on the user side. During the start, the *boom* GUI checks if a server side config file does exist for the current user role.

> GUI settings are defined on a user role level!

Local profiles are stored for each user/*boom* server combination.

8.2.1. User Interface Configuration Files

Server side Config File:

The server side config file is always the master! If this file exist, the local config file will be ignored. In case there is no server side config file, the local config file will be used. If there is no server and no local config file, a new local config file will be created.

UI settings can only be defined by users that belong to the administrator role. The server side config file is read only for all other users. Changes to the UI (done by the user) will not be saved to the server config file! The ability of the user to reorganize the *boom* GUI depends on the settings inside the server config file.

Local Config File:

The local UI config file is only used, if the server side UI config file does not exist. The local config file is created after the first start of the *boom* GUI. In this case, the user is allowed to reorganize the GUI without any restrictions. All changes made to the *boom* workbench (i.e. moving views, opening new "Indication Browsers", …) will be saved to the local config file.

File Name

To make sure that the user roles are using the correct config file, the name of the xml file must be equal to the user role id. You can get the user role id from your database.

user role id:	1ded5a54-7f8c-4301-9821-9fb98ed81079

file name:	1ded5a54-7f8c-4301-9821-9fb98ed81079.xml

File Path

Put your server side config file in the following directory: **\srv\profiles\**

The local config file is called **profile.xml** and can be found in the **local user directory**.

8.2.2. XML Attributes

The following table contains all available attributes that can be defined in the UI config file. All attributes that are set to '0' are NOT allowed. All values other than '0' will be enabled for the specific action.

attribute_value* = *0	Action is NOT allowed
attribute_value* >= *1	Action is allowed

Global Attributes	
Global attributes apply to all indication tabs.	
CAN_MOVE	Possible values: 0|>=1 This attribute defines if the user is allowed to move "Indication Tabs" within the "Indication View" or the "Indication View2".
CAN_SORT	Possible values: 0|>=1 Here you can define whether the user is allowed to sort all indication tables or not.
CAN_SWITCH	Possible values: 0|>=1 This attribute describes if the user is allowed to switch the perspective of indication tables between active and closed indications.
CAN_MODIFY_FILTERS	Possible values: 0|>=1 You can decide if the user is allowed or not allowed to change the filtering of indication tables.

Attributes on View level	
The following attributes apply to "Indication View1" and "Indication View2".	
VIEW_ID	Every "Workbench View" can be identified on the basis of its unique View ID. Here is the list of available View IDs: Indication View1: "com.blixx.boom.gui.views.msgview" Indication View2: "com.blixx.boom.gui.views.msgview:MsgView2"
CLOSABLE	Possible values: 0|>=1 Describes if the user is allowed to close the view with the given view ID.
MOVEABLE	Possible values: 0|>=1 Describes if the user is allowed to move this view within the *boom* workbench.
ADD_NEW_TABS	Possible values: 0|>=1 This attribute defines if the user is allowed to open/close new indication tabs inside the "Indication View" with the given view ID.

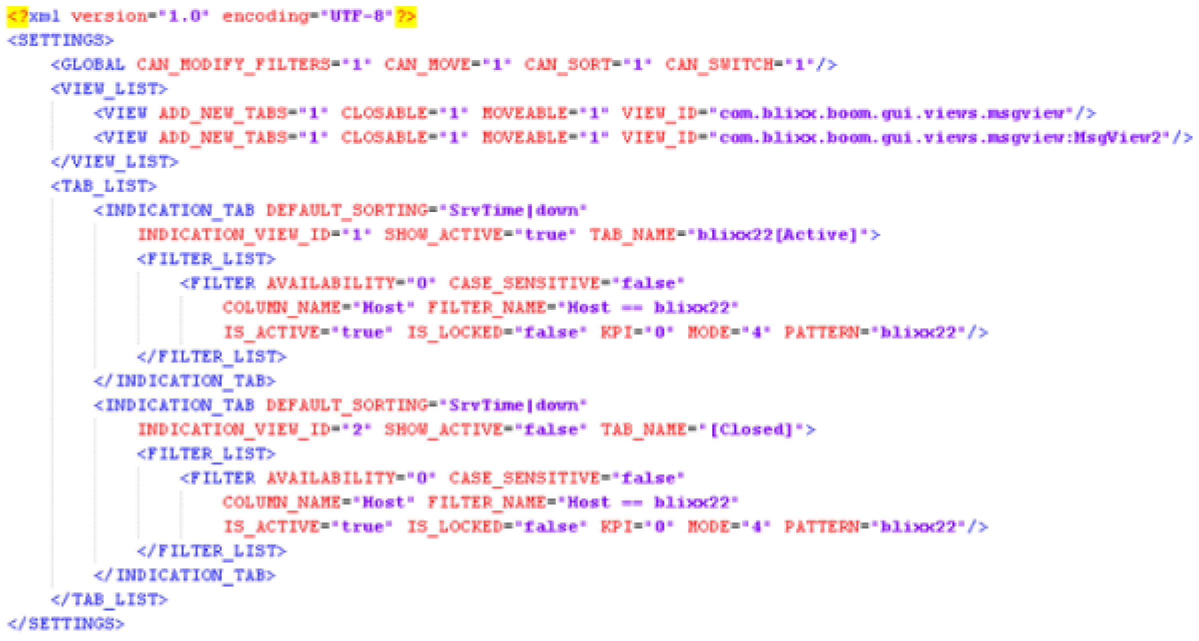
Attributes on Tab level	
The following attributes apply to all Indication Tabs.	
INDICATION_VIEW_ID	Possible values: 1|2 It defines, if the tab will be opened in the "Indication Browser1" or the "Indication Browser2".
TAB_NAME	Name of the tab.
SHOW_ACTIVE	Possible values: true|false Set attribute to true if you want the user to see all active indications in this tab.
DEFAULT_SORTING	Possible values: column|sortdirection You can define the default sorting of the indication table. Example: "SrvTime|down" → The table will be sorted downwards by the column ServerTime.

Filter Attributes
The Filter Attributes apply to all indication tables. For more information about filters see chapter Indication Filtering.

8.2.3. XML File Example

How to create a GUI config file?

Login with a user who does not belong to the administrator role. Make sure that this user or rather this user role must not have a server side GUI config file! If you are logged in you can organize your workbench: open new indication browser and add possible filters to them. Check the profile.xml file in the user directory. The user side GUI config file has been created automatically and keeps all necessary user settings. You can now adapt this xml file for your needs and move it to the server.



```
<?xml version="1.0" encoding="UTF-8"?>
<SETTINGS>
    <GLOBAL CAN_MODIFY_FILTERS="1" CAN_MOVE="1" CAN_SORT="1" CAN_SWITCH="1"/>
    <VIEW_LIST>
        <VIEW ADD_NEW_TABS="1" CLOSABLE="1" MOVEABLE="1" VIEW_ID="com.blixx.boom.gui.views.msgview"/>
        <VIEW ADD_NEW_TABS="1" CLOSABLE="1" MOVEABLE="1" VIEW_ID="com.blixx.boom.gui.views.msgview:MsgView2"/>
    </VIEW_LIST>
    <TAB_LIST>
        <INDICATION_TAB DEFAULT_SORTING="SrvTime|down"
            INDICATION_VIEW_ID="1" SHOW_ACTIVE="true" TAB_NAME="blixx22[Active]">
            <FILTER_LIST>
                <FILTER AVAILABILITY="0" CASE_SENSITIVE="false"
                    COLUMN_NAME="Host" FILTER_NAME="Host == blixx22"
                    IS_ACTIVE="true" IS_LOCKED="false" KPI="0" MODE="4" PATTERN="blixx22"/>
            </FILTER_LIST>
        </INDICATION_TAB>
        <INDICATION_TAB DEFAULT_SORTING="SrvTime|down"
            INDICATION_VIEW_ID="2" SHOW_ACTIVE="false" TAB_NAME="[Closed]">
            <FILTER_LIST>
                <FILTER AVAILABILITY="0" CASE_SENSITIVE="false"
                    COLUMN_NAME="Host" FILTER_NAME="Host == blixx22"
                    IS_ACTIVE="true" IS_LOCKED="false" KPI="0" MODE="4" PATTERN="blixx22"/>
            </FILTER_LIST>
        </INDICATION_TAB>
    </TAB_LIST>
</SETTINGS>
```

8.2.4. Setting Default Login Parameters

It is possible to configure the *boom* client to use specified parameters like *boom* server hostname, port, login name and password as default values. Just add the following parameters to the boomgui.ini configuration file which is located in the *boom* client installation directory:

boomgui.ini

Parameter	Description
BOOMGUI_LOGLEVEL	Use loglevel 1 to 4
BOOM_HOST	Default host name is used for the very first login. After the first successful login, the 'Login Dialog' remembers the last host. If no default host is configured, 'localhost' is the default.
BOOM_PORT	Default port is used for the very first login. After the first successful login, the 'Login Dialog' remembers the last port. If no default port is configured, '23022' is the default.
BOOM_USER	User name - Predefine the user name in the 'Login Dialog'
BOOM_PASS	User password - Predefine the password in the 'Login Dialog'.
BOOM_AUTOLOGIN	true/false - automatically login the user with the credentials specified in BOOM_USER and BOOM_PASS
BOOM_LOGDIR	Directory for UI logfiles. Please note: this directory must exist before logging into the *boom* UI. Default: <boom client installation directory>:<login_user>_<host>
BOOM_PROFILE_DIR	Directory for the *boom* user profile. Default: <boom client installation directory>:<login_user>_<host>
BOOM_TITLE	Activates a title bar for the login window with maximize, minimize and close button.
BOOM_IGNORE_CLOSED_INDICATIONS	"Ignore Closed Indication" check box will be selected by default when the UI starts (closed indications can be manually loaded in the UI)
BOOM_IGNORE_ALL_INDICATIONS	"Ignore All Indication" check box will be selected by default when the UI starts
vm	Path to a specific JVM which should be used to run the gui.
vmargs	All options which are directly parsed to JVM. Must be separated by newline and "-"

Example default boomgui.ini:

```
-data
@noDefault
-BOOMGUI_LOGLEVEL
1
-vmargs
-Dfile.encoding=UTF8
-Xmx512M
```

Example customized boomgui.ini:

```
-data
@noDefault
-BOOMGUI_LOGLEVEL
2
-BOOM_HOST
localhost
-BOOM_PORT
23022
-BOOM_USER
op1
-BOOM_PASS
op1
-BOOM_AUTOLOGIN
true
-BOOM_LOGDIR
C:\temp\boomguui\log
-BOOM_PROFILE_DIR
C:\temp\boomguui\profile
-vmargs
-Dfile.encoding=UTF8
-Xmx512M
```

By default, the *boom* client uses default JVM installed on the system. You can specify a specific JVM for *boom* client by adding **-vm** parameter in the boomgui.ini file.

The -vm option must occur after the other specific options and before -vmargs options, since everything after -vmargs is passed directly to the JVM.

Example customized JVM in boomgui.ini:

```
-data
@noDefault
-BOOMGUI_LOGLEVEL
1
-vm
C:\Java\JDK\1.8\bin\javaw.exe
-vmargs
-Dfile.encoding=UTF8
-Xmx512M
```

All *boom* specific parameters are case sensitive and need to be inserted before the Java specific part (that means before -vmargs) and after "@noDefault". Furthermore every parameter, name and value has to be inserted into its own line.

8.3. Internal Indications / Operational States

8.3.1. *boom* Agent Internal Indications

The following table gives an overview of all internal *boom* agent indications.

Table of internal indications (v2.66.000 +)

Application	Group	Object	Severity	Text
AGENT	<callType>	<policyName>	warning	Trigger for policy <policyName> failed to init.<*>

Application	Group	Object	Severity	Text
AGENT	AGENT	AGENT	warning	Agent disabled
AGENT	AGENT	AGENT	critical	Clone detected. Old ID=<IDBeforeClone> new ID=<newID>
AGENT	AGENT	AGENT	critical	Agent has <size>MB memory left
AGENT	AGENT	AGENT	normal	Agent enabled
AGENT	AGENT	AGENT	warning	Agent disabled
AGENT	AGENT	INVENTORY	normal	Inventory change detected
AGENT	DEPLOY	<pkgName>	normal	Package <pkgName> successfully deployed.
AGENT	DEPLOY	<pkgName>	major	Deployment of package <pkgName> failed. <*>
AGENT	DEPLOY	<pkgName>	warning	File <filename> is locked. The Agent will try to redo action
AGENT	DEPLOY	<policyName>	normal	Policy <policyName> v<version> was deployed.
AGENT	DEPLOY	<policyName>	critical	Deployment of Policy <policyName> failed.
AGENT	JAVA_ MONITOR	<policyName>	warning	Trigger can’t be initialized: <MonProgJavaCall><*>
AGENT	JAVA_ MONITOR	<policyName>	warning	Monitor can’t be initialized: <className><*>
AGENT	JAVA_ MONITOR	<policyName>	minor	Java monitor failed during execution.<*>
AGENT	MONITOR	<policyName>	warning	Monitor failed to start.<*>
AGENT	MONITOR	<policyName>	warning	Monitor does not submit a value during the last polling interval
AGENT	RESTART	AGENT	normal	Agent restarting.
AGENT	Storm Detection	<policyName>	critical	Policy blocked. Message Storm Detected
AGENT	Storm Detection	<policyName>	normal	Policy unblocked.
AGENT	UNDEPLOY	<pkgName>	normal	Package <pkgName> successfully undeployed.
AGENT	UNDEPLOY	<pkgName>	major	Undeployment of package <pkgName> failed.<*>
AGENT	UNDEPLOY	Message:<policyName>	warning	Message policy file can’t be deleted. <filename>

Application	Group	Object	Severity	Text
AGENT	UNDEPLOY	Message:<policyName>	warning	Can't remove mon <policyName>. Not assigned on this node.
AGENT	UNDEPLOY	Message:<policyName>	normal	Removed message policy: <policyName>
AGENT	UNDEPLOY	Message:<policyName>	normal	Removed mon task <policyName>
AGENT	UNDEPLOY	Monitor:<policyName>	warning	Policy file can't be deleted. <filename>
AGENT	UNDEPLOY	Monitor:<policyName>	warning	Can't remove mon task: <policyName>
AGENT	UNDEPLOY	Monitor:<policyName>	warning	Can't remove mon task: <policyName>. Not assigned on this node.
NAGIN	NAGIN	<*>	<*>	<*>

8.3.2. *boom* Agent Operational States and Indications

In the GUI a status symbol is assigned to every agent according to the agent's operational state. An internal indication is created when the operational state is set/changes.

Agent Status Symbol and Created indication:

Agent is running

Agent is online

Agent is running and Agent is firewalled

Agent is online
no extra indication for firewalled status

Agent is not running

Agent is offline

Agent is not running and Agent is firewalled

Agent is offline
no extra indication for firewalled

Agent is disabled

Agent disabled

Agent is waiting for approval

no indication

8.4. Message Storm Detection

The *boom* agent is able to detect and stop message storms to avoid an indication flood on the *boom* server. The single policy which caused the storm will be automatically blocked for a specified time interval if the configured message rate is reached within 1 minute.

Agent specific Configuration Variables

The agent's message storm detection component uses two configuration variables to prevent message storms:

MSD_MAX_PER_MINUTE	maximum messages per minute
MSD_BLOCK_MINUTES	blocking period in minutes

MSD_MAX_PER_MINUTE specifies how many messages can arrive per minute before a message storm is detected and the single policy is blocked for the time interval MSD_Block_MINUTES. The agent drops the messages which are matched by this policy for the specified blocking period.

Example:

MSD_BLOCK_MINUTES=5
MSD_MAX_PER_MINUTE=500

The policy which caused the message storm is blocked for 5 minutes if more than 500 indications per minute are created.

Only MATCHED MESSAGES and SEND MESSAGES are taking into account. So it EXCLUDES unmatched messages and messages matched with STOP condition.

Set/Deactivate/Check Message Strom Detection

Use the remote agent actions in the GUI to Set/Deactivate/Check the Message Storm Detection parameters on an agent basis. Select the "Actions" button in the *boom* GUI. In the BOOM_AGENTS_v2.65 tree you will find two agent actions:

Check the Message Storm Detection settings:

Use **"Get Message Storm Detection settings"** to get the current settings.

Action Command:

BOOM_AGENT GET_MSD_SETTINGS

Set Message Storm Detection:

Use **"Set Message Storm Detection settings"** to enable Message Storm Detection.

Command:

BOOM_AGENT SET_MSD_SETTINGS

Optional Parameters:

[maximum number of messages per minute] [blocking period in minutes]
e.g. 500 5

Deactivate Message Storm Detection:

Use **"Set Message Storm Detection settings"** to deactivate Message Storm Detection.

To deactivate Message Storm Detection set [maximum number of messages per minute] and [blocking period in minutes] to 0:

Command:

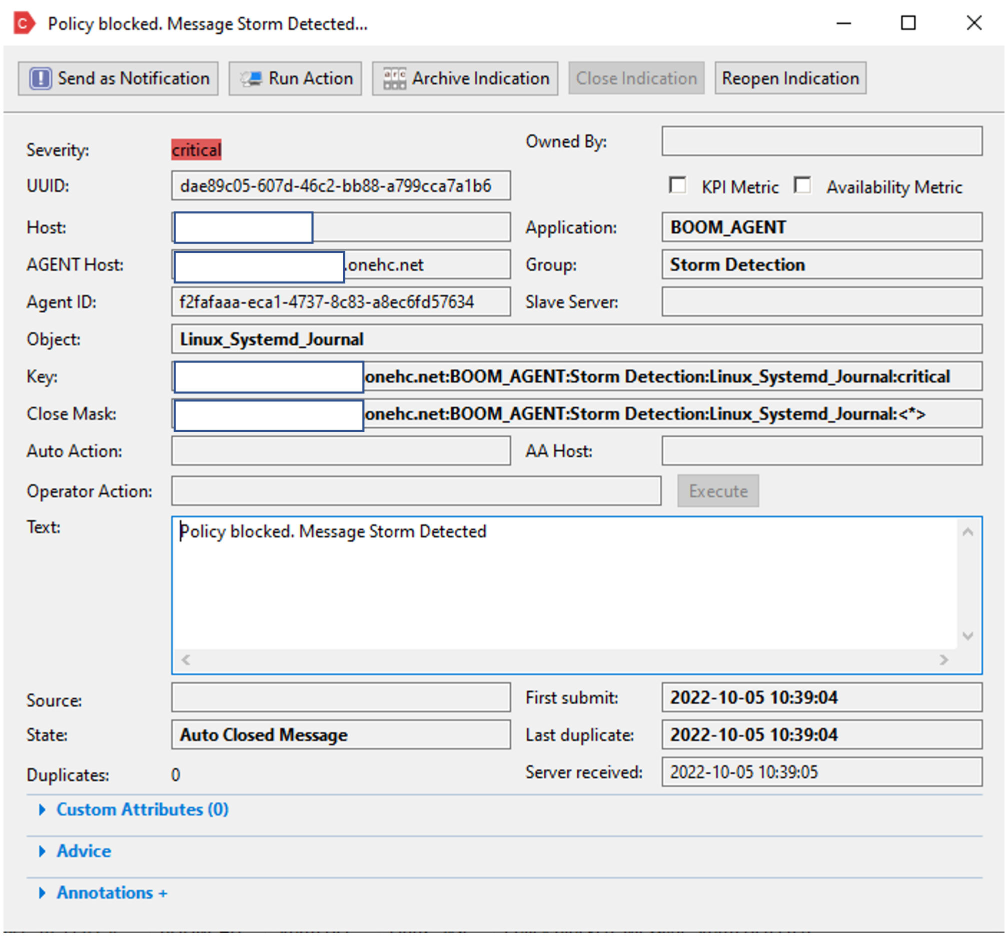
BOOM_AGENT SET_MSD_SETTINGS

Optional Parameters:

0 0

Indications generated on the *boom* Server

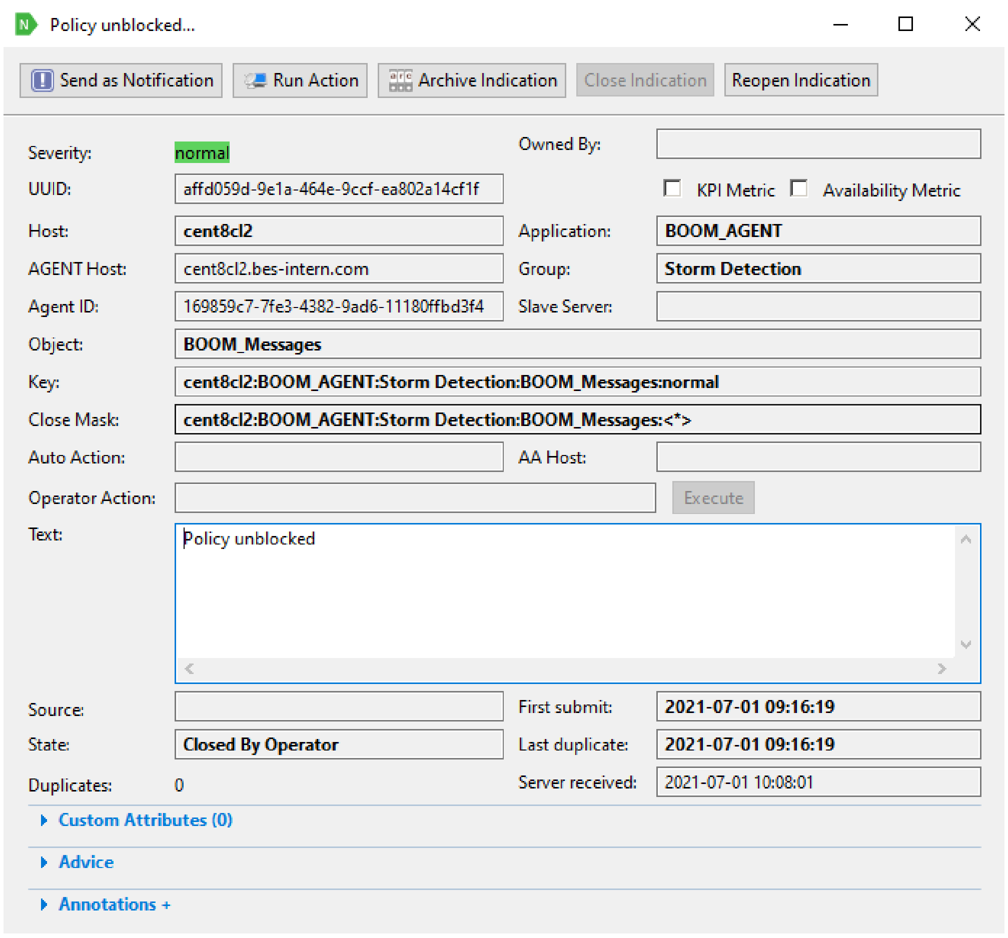
A **critical indication** is generated when a message storm for an agent is detected.



Policy blocked. Message Storm Detected...

Severity:	critical	Owned By:	
UUID:	dae89c05-607d-46c2-bb88-a799cca7a1b6	☐ KPI Metric	☐ Availability Metric
Host:	[redacted]	Application:	BOOM_AGENT
AGENT Host:	[redacted].onehc.net	Group:	Storm Detection
Agent ID:	f2fafaaa-eca1-4737-8c83-a8ec6fd57634	Slave Server:	
Object:	Linux_Systemd_Journal		
Key:	[redacted]onehc.net:BOOM_AGENT:Storm Detection:Linux_Systemd_Journal:critical		
Close Mask:	[redacted]onehc.net:BOOM_AGENT:Storm Detection:Linux_Systemd_Journal:<*>		
Auto Action:		AA Host:	
Operator Action:			
Text:	Policy blocked. Message Storm Detected		
Source:		First submit:	2022-10-05 10:39:04
State:	Auto Closed Message	Last duplicate:	2022-10-05 10:39:04
Duplicates:	0	Server received:	2022-10-05 10:39:05

An **normal indication** is generated when the message storm is over.

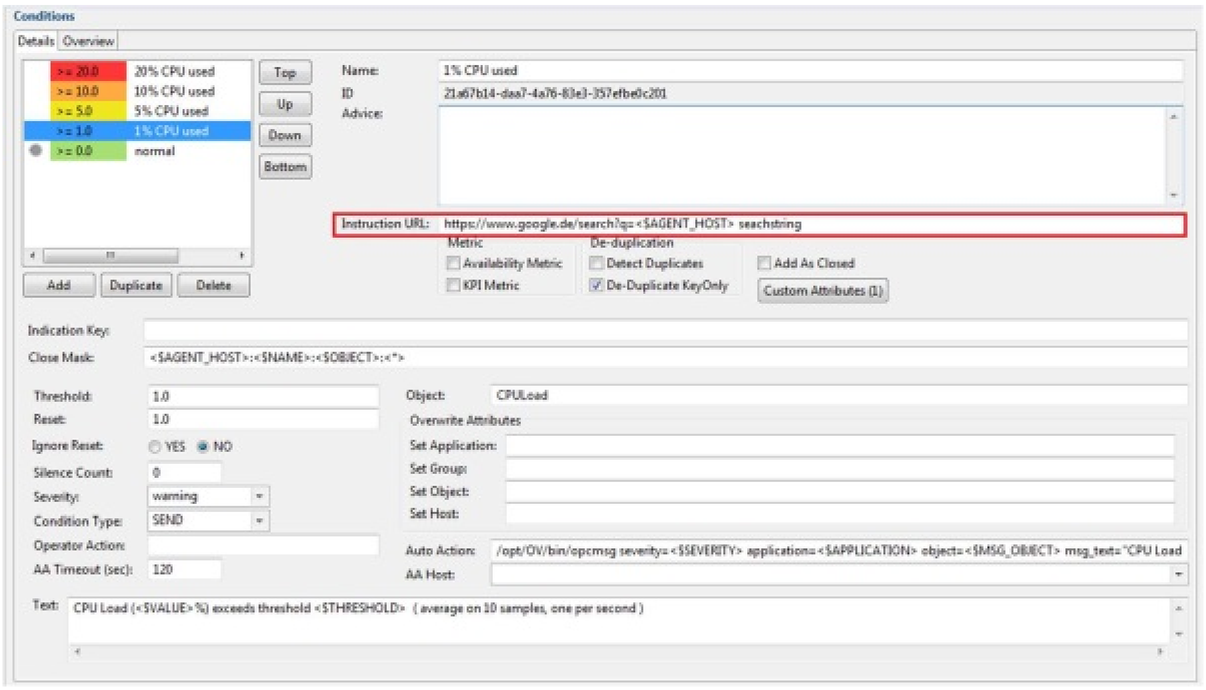


Severity:	normal	Owned By:	
UUID:	affd059d-9e1a-464e-9ccf-ea802a14cf1f	☐ KPI Metric	☐ Availability Metric
Host:	cent8cl2	Application:	BOOM_AGENT
AGENT Host:	cent8cl2.bes-intern.com	Group:	Storm Detection
Agent ID:	169859c7-7fe3-4382-9ad6-11180ffbd3f4	Slave Server:	
Object:	BOOM_Messages		
Key:	cent8cl2:BOOM_AGENT:Storm Detection:BOOM_Messages:normal		
Close Mask:	cent8cl2:BOOM_AGENT:Storm Detection:BOOM_Messages:<*>		
Auto Action:		AA Host:	
Operator Action:			
Text:	Policy unblocked		
Source:		First submit:	2021-07-01 09:16:19
State:	Closed By Operator	Last duplicate:	2021-07-01 09:16:19
Duplicates:	0	Server received:	2021-07-01 10:08:01

8.5. Defining an Instruction URL

- **Instruction URL**
  Instruction URL is basically a URL-link which you can add to your conditions and which appears in the generated indications. The possible usage of this link ranges from offering a more detailed description of the error and/or countermeasures, to open a ticket auto filling some or even all data.

  The Instruction URL is defined in the condition part of the policy:



🟥 >= 20.0	20% CPU used
🟧 >= 10.0	10% CPU used
🟨 >= 5.0	5% CPU used
>= 1.0	1% CPU used
🟩 >= 0.0	normal

Name: 1% CPU used
ID: 21a67b14-daa7-4a76-83e3-357efbe0c201
Advice:
Instruction URL: https://www.google.de/search?q=<$AGENT_HOST> seachstring
☐ Availability Metric ☐ KPI Metric ☐ Detect Duplicates ✓ De-Duplicate KeyOnly ☐ Add As Closed
Indication Key:
Close Mask: <$AGENT_HOST>:<$NAME>:<$OBJECT>:<*>
Threshold: 1.0
Reset: 1.0
Ignore Reset: YES [✓ NO]
Silence Count: 0
Severity: warning
Condition Type: SEND
Operator Action:
AA Timeout (sec): 120
Object: CPULoad
Set Application:
Set Group:
Set Object:
Set Host:
Auto Action: /opt/OV/bin/opcmsg severity=<$SEVERITY> application=<$APPLICATION> object=<$MSG_OBJECT> msg_text="CPU Load
AA Host:
Text: CPU Load (<$VALUE>%) exceeds threshold <$THRESHOLD> (average on 10 samples, one per second)

- **Limitations**
  Every URL can be used; the only limitation is that the URL will be truncated to 512 chars.
- **Variables**
  It is possible to insert agent/message variables in the URL, they will be replaced by their contents on the generated indication. Depending on the policy type (monitor or indication) some restrictions to the available variables may apply, for more information please refer to the *boom* documentation chapter Supported Variables and Functions.

Instruction Server Variable

For the Instruction URL *boom* recognizes the variable <$INSTRUCTION_SERVER>.
This variable can be configured in the boom.props file of the *boom* server:

```
INSTRUCTION_SERVER=
```

Instead of adding the entire URL to the Instruction URL in the policy conditions, it is possible to add only a specific part to that condition.

Example:

boom.props:

```
INSTRUCTION_SERVER=http://myserver.net:8912/queryInstruction.cgi?
```

Set Instruction URL in the condition:

```
<$INSTRUCTION_SERVER>errorcode=8192
```

Instruction URL will resolve as:

```
http://myserver.net:8912/queryInstruction.cgi? errorcode=8192
```

This will be useful for companies which have their own knowledge base server.

Customized Attributes

The Instruction URL supports the use of customized attributes. For more information on customized attributes please refer to chapter Monitor Policy in the GUI part of the documentation.

- How it works
  Open a corresponding indication and open the URL specified in the instruction field.
  The default system browser will be opened (if it is not already open) and a new window/tab will be created with the contents of the Instruction URL.

  Be aware that the installed browser has to be firefox until a system variable defining the default browser under linux or other UNIX-derivates is set.

URL Examples:

```
http://www.google.de/search?q=test
```

Will open www.google.de and search for "test".

```
http://www.google.de/search?q=<AGENT_HOST> test
```

Will open www.google.de and search for "the hostname of the agent" and "test".

```
http://www.ticketsystem.de/openticket?host=<AGENT_HOST>&ip=<$AGENT_IP>&policyname=<$NAME>&messagetext=<$MS
G_TEXT>
```

This will open www.ticketsystem.de, open a new ticket and fill data like hostname, ip, pocliy name and message text (this example will not work because there is no www.ticketsystem.de but it should illustrate the possibilities)

```
http://www.oracle.com/pls/db10g/error_search?search=ORA-04094
```

This will search for the Oracle Database error code ORA-04094.

8.6. Database Issues

8.6.1. Database Insert of Indications

Succesfully database insert of indications

Normally all incoming indications are inserted in blocks directly into the database without any problems.

Database insert of faulty/failed indications

If the block insert into the database fails the server continues in the following way:

- If the w**hole block insert** into the database fails for all indications the server continues trying to insert blocks. The server assumes that there is for example a connection problem to the database or no space on the filesystem etc.
    - The exception is raised as a critical indication: "DBERROR"
    - A Log File Entry DBERROR is written into ".../server/BOOMserver_<date>.log"
    - If for example the DB connection is reestablished, the missing faulty inserts will be reapplied.
- If only a **part of the block insert** fails then the failed/faulty indications are moved into a **"failover queue"**. This queue is stored on ".../srv/etc/failed_inserts.dat."
    - The exception is raised as a critical indication: e.g. "DB Error: Data truncation: Data too long for column 'SRV' at row 1. Affected indications (number of indication)"
    - A Log File Entry DBERROR is written into ".../server/BOOMserver_<date>.log"
    - The "failover queue" will be loaded in case of a server restart hence no indications (due to failed inserts) will be lost.
    - If such active indications (failed inserts) are closed in the GUI, the indications are still in the "failover queue" and will be loaded after a server restart. In case of archiving the closed indications (due to failed inserts) they will be deleted in the failover queue and exported! Exported indications will not be visible in the GUI and are stored in ".../server/srv/etc/afi/failed_inserts.*" The file rotates after 1MB (size of rotation is not configurable).

8.7. SNMP Trap Listener

The assignment group **"SNMP-TrapReceiver"** or at least the **"SNMP"** package and the **"SNMPTrapd_Trigger"** policy must be deployed to the specified agent to enable traps receiving.

The main responsibility of the **SNMPTrapd_Trigger** policy is to start the **SNMP Trapd Listener**. **No** trap matching conditions has to be specified in this policy. This policy will be skipped during processing! The default SNMP listener port is **162** specified in the Monitor Call field of the policy: **-p 162**

Enter the following command to check if the listener is running on the agent:

```
# netstat -an | grep 162
```

To change the default listener port just modify the parameter **-p <port>**. Redeploy the **SNMPTrapd_Trigger** policy to start listener on the new port.

Hint: when Linux non-root agent act as SNMP Trap Receiver it can not listen to standard port 162 because this is not allowed by OS and therefore port must be changed to value higher than 1024. Additionally, port redirecting from UDP 162 to new port has to be configured at Linux agent system by OS administration.

8.8. Primary-Backup Connection

8.8.1. Setting up a Primary/Backup Connection

Establish a Primary – Backup connection

To connect one *boom* server to another one, follow the steps below:

1. The Primary server has to be able to connect to the UI port (default 23022) of the Backup server.
2. A *boom* user in role Administrator has to be created at the Backup server. These credentials are used by the Primary to connect to the Backup. `e.g.: user b2b with password b2bsecret` see chapter User Management
3. Make sure that you have sufficient licenses for your designed *boom* - scenario. see chapter Licensing
4. Start the *boom* GUI and login to the Primary server
   → open the Actions view
   → select "BOOM_SERVER"
   → start the action: "Add new Source Server".
5. In the server action "Add new Source Server" select the following values:

   Server:
   Select the Backup server to which the Primary should be connected.

   Parameters:

   The Backup server name, user, password and the scenario (mode 8) have to be added in the **"Parameters"** field. Port and isTLS are optional parameters, in case not specified 23031 and true are going to be used.

   ```
   boom-test.mybes.local      b2b    b2b        8          23031  true
   <full qualified hostname>  <user> <password> <scenario> <port> <isTLS>
   ```

Two servers with Primary / Backup connection (mode 8) have to use the same certificate authority (ca) on both servers in order to ensure TLS communication of all agents to both servers.
For self-signed certificates this would mean that after installation of both servers and before connecting both servers, the cakeystore.p12 of the Primary has to be copied to the Backup server and its key- and truststore have to be recreated.

On the Backup:

```
/opt/boom/server/boom_srv -stop
rm -f /opt/boom/server/srv/etc/cakeystore.p12
rm -f /opt/boom/server/srv/etc/keystore.p12
rm -f /opt/boom/server/srv/etc/truststore.p12
cp /tmp/<cakeystore.p12_from_master> /opt/boom/server/srv/etc/cakeystore.p12
/opt/boom/server/boom_srv -start
```

For file-based pki certificates we expect that all certificate signing requests are signed by the same external certificate authority.

Scenario:

The scenario describes the control mode that is used for the cooperation of the servers.

8	Primary-Backup	All activities from the Primary (Master) are forwarded to the Backup (Slave) server. In case of an outage of the Primary Server the agent will start to communicate with the Backup server to ensure service availability. If the Primary is working again, data is not resynchronized between Primary and Backup.

6. Select **"Execute"** to run the action. In the server action **"Add new Source Server"** select the following values:

   The **Hosts** view in the *boom* UI (user interface) will show the Slave Server as soon as the connection was established.

Agent configuration must be adjusted to enable communication with Primary and Backup server at the same time. Below, you find a short example of properties used to enable proper communication:

```
CLUSTER_NODES=<primaryServerHost>,<primaryServerIP>,<backupServerHost>,<backupServerIP>
OFFER_DATA_TO_ANY_CLUSTER_NODE=true
MAIN_SERVER_IP=<primaryServerIP>
MAIN_SERVER_NAME=<primaryServerHost>
BACKUP_SERVER_IP=<backupServerIP>
BACKUP_SERVER_NAME=<backupServerHost>
```

Agent is switching to the Backup server after more than 3 consecutive failed heartbeats to the Primary server.

To delete a Backup server select the server action **"Remove Source Server":**

This action removes a Backup. The <backup server name> and the <user> has to be specified in the **"Parameters"** field.

Reset to default state:

All Backup (slave) settings can be found on the corresponding master under:

```
<boom_server>/srv/slaves
```

To restore the default state (no slaves connected) delete all files in that folder: (Be careful, this should be seen only as a last resort.)

8.9. Master/Slave Connection

8.9.1. Setting up a Master/Slave Connection

Establish a master – slave connection

To connect one *boom* server to another one, follow the steps below:

1. The master server has to be able to connect to the UI port (default 23022) of the slave server.
2. A *boom* user has to be created at the slave server. These credentials are used by the master to connect to the slave. `e.g.: user b2b with password b2bsecret` see chapter User Management
3. Make sure that you have sufficient licenses for your designed *boom* - scenario. see chapter Licensing
4. Start the *boom* GUI and login to the master server
   → open the Actions view
   → select "BOOM_SERVER"
   → start the action: "Add new Source Server".
5. In the server action "Add new Source Server" select the following values:

   Server:
   Select the master server to which the slave should be connected. For the first added slave only the master server is available in the list.

   Parameters:
   The slave server name, user, password and the scenario (mode) have to be added in the **"Parameters"** field. Port and isTLS are optional parameters, in case not specified 23031 and true are going to be used.

```
boom-test.mybes.local       b2b    b2b         1           23031   true
<full qualified hostname>   <user> <password>  <scenario>  <port>  <isTLS>
```

Two servers with Primary / Backup connection (mode 8) have to use the same certificate authority (ca) on both servers in order to ensure TLS communication of all agents to both servers.
For self-signed certificates this would mean that after installation of both servers and before connecting both servers, the cakeystore.p12 of the Primary has to be copied to the Backup server and its key- and truststore have to be recreated.

On the slave:

```
/opt/boom/server/boom_srv -stop
rm -f /opt/boom/server/srv/etc/cakeystore.p12
rm -f /opt/boom/server/srv/etc/keystore.p12
rm -f /opt/boom/server/srv/etc/truststore.p12
cp /tmp/<cakeystore.p12_from_master> /opt/boom/server/srv/etc/cakeystore.p12
/opt/boom/server/boom_srv -start
```

For file-based pki certificates we expect that all certificate signing requests are signed by the same external certificate authority.

Scenario:

The scenario describes the control mode that is used for the cooperation of the servers.

1	Proxy Slave	the Indications are forwarded with control to this Master Server. Close activities are synchronized between the Source and Master Server. Indications that are closed on the Master get archived on the Source Server. Configuration information like Policies, Packages, Assignment Groups, Host Groups, Actions are pushed from the Master to the Source Server.
2	Mirror	the Indications are forwarded with control to this Master Server. Close activities are synchronized between the Source and Master Server.
3	Read-only	the Indications are forwarded "FYI" to this Master Server. Close and archive activities are forwarded to the Master Server, but closing/archiving on the Master Server is not reflected on the Source Server (i.e. the Source Server remains in control of its Indications).
8	Primary-Backup	All activities from the Primary (Master) are forwarded to the Backup (Slave) server. In case of an outage of the Primary Server the agent will start to communicate with the Backup server to ensure service availability. If the Primary is working again, data is not resynchronized between Primary and Backup.

6. Select **"Execute"** to run the action. In the server action **"Add new Source Server"** select the following values:

   The **Hosts** view in the *boom* UI (user interface) will show the Slave Server as soon as the connection was established.

Agent configuration must be adjusted to enable communication with Primary and Backup server at the same time. Below, you find a short example of properties used to enable proper communication:

```
CLUSTER_NODES=<primaryServerHost>,<primaryServerIP>,<backupServerHost>,<backupServerIP>
OFFER_DATA_TO_ANY_CLUSTER_NODE=true
MAIN_SERVER_IP=<primaryServerIP>
MAIN_SERVER_NAME=<primaryServerHost>
BACKUP_SERVER_IP=<backupServerIP>
BACKUP_SERVER_NAME=<backupServerHost>
```

Agent is switching to the Backup server after more than 3 consecutive failed heartbeats to the Primary server.

To delete a slave server select the server action **"Remove Source Server":**

This action removes a slave. The <slave server name> and the <user> has to be specified in the **"Parameters"** field.

Reset to default state:

All slave settings can be found on the corresponding master under:

```
<boom_server>/srv/slaves
```

To restore the default state (no slaves connected) delete all files in that folder: (Be careful, this should be seen only as a last resort.)

8.10. Configuring *boom* Auditing

Auditing records the following user actions to the file BOOMAudit_<date>.log:

+

Assignment tree changes	create, rename, delete elements
Policies	add, modify, delete
Policy tree update	create, rename, delete elements
Agent Card	add, enable/disable, Set Firewalled ON/OFF
Maintenance: Modify Server Policies	add, modify, delete, enable, disable
Deployments	Deploy, Re-Deploy, UnDeploy, Queue Deployments
Server/Agent: Synchronizing the configuration	Trigger Synchronization

Defaults for File based Auditing:

Name of the Auditing logging file: **BOOMAudit_<date>.log**

Default directory for logging file: <**boom installation directory**>

To **enable Auditing**, set the following parameter in the *boom* server configuration file *boom.props*:

```
    AUDITENABLE=true
    AUDITLOGDIR=<log directory> e.g. "."
    AUDITLOGSIZE=<log size in MB> e.g. 12
    AUDITLOGCOUNT=<number of log files> e.g. 3
```

To **disable Auditing**, set the following parameter in the *boom* server configuration file *boom.props*:

```
AUDITENABLE=false
```

8.11. Configuring Outages

8.11.1. Configuring Agent based Maintenances

Any user or script can submit a Maintenance via "boom_agent_cli.jar" (*boom* package) which has to be deployed to corresponding agent.

Enable an Maintenance (Add an Agent Maintenance rule):

```
java –jar boom_agent_cli.jar Maintenance -e <options>
Agent will generate an Maintenance ID and prints it as result to STDOUT. This ID must be used for
disabling the Maintenance.
```

<options>:

-ed <interval>	expected duration of the Maintenance Format of interval: 1w2d3h4m5s - 1 week, 2 days, 3 hours, 4 min, 5 seconds minimum interval: 1 minute (Server will create an indication if the agent does not disable the Maintenance in the expected period of time)
-du <interval>	(optional) MAX duration of the Maintenance (Server will disable the Maintenance automatically after this interval even if the agent does not send stop). Format of interval: 1w2d3h4m5s - 1 week, 2 days, 3 hours, 4 min, 5 seconds minimum interval: 1 minute (Server will create an indication if the agent does not disable the Maintenance in the expected period of time)
-ac [HIDE|DROP]	action (what's to do with the matched indications) HIDE - means an indication goes to the Maintenance browser. DROP - means an indication is dropped.
-pa [NONE|DROP|PUBLISH]	only valid if action = HIDE Specifies what the post action has to perform after the end of the Maintenance. NONE - the indications will stay in the Maintenance browser after the Maintenance is ended DROP - the indications will be dropped from the Maintenance browser at the end of the Maintenance. PUBLISH – the indications will be moved to the active browser at the end of the Maintenance
<AttributeName>=<Simplified Pattern>	set 1..n filter(s) filter indication attribute for setting into Maintenance **Supported Filter Attributes:** APPLICATION GROUP OBJECT HOST (alias: NODE) CA CA1 – CA15 NODEGROUP SOURCE TEXT SEVERITY KEY

Disable an Maintenance:

```
java –jar boom_agent_cli.jar Maintenance -d -id <Maintenance ID>
```

or

Use the **"Show Agent Maintenances view"** in the GUI to disable the agent Maintenance policy.

Check the Maintenance:

Use the "Show Agent Maintenances view" in the GUI to check the agent Maintenance policy.

Maintenance Browser:

An **"Maintenance indication browser"** exists in parallel to the active indication browser. It holds all HIDDEN indications (action=HIDE). Any UI client is able to start the "Maintenance browser" by opening it in the "Indication Tab content menu" (right click -→ Add Maintenance Tab)

Example1:
Create an Maintenance with Expected end only (unlimited duration): `java –jar boom_agent_cli.jar Maintenance -e -ed 1h -ac HIDE -pa NONE APPLICATION="Test app<*>" GROUP=grp1`
Output:
`9dcffe27-77db-4267-a434-ae6b7432e476`

Stop the Maintenance:
`java –jar boom_agent.cli.jar Maintenance -d -id 9dcffe27-77db-4267-a434-ae6b7432e476`

Example2:
Create an Maintenance with limited duration `java –jar boom_agent_cli.jar Maintenance -e -ed 1h -du 2h -ac HIDE -pa NONE APPLICATION="Test app<*>" GROUP=grp1`
Output:
`9dcffe27-77db-4267-a434-ae6b7432e476`

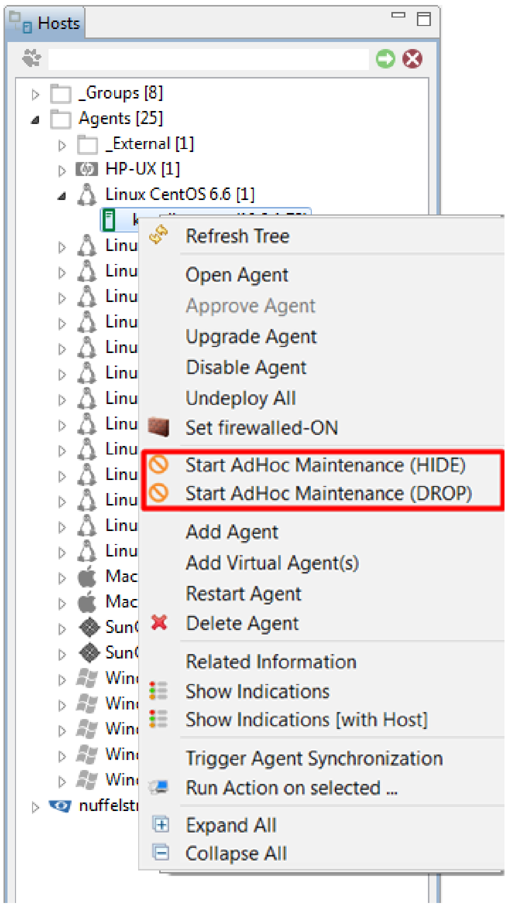
stop Maintenance: `java –jar boom_agent.cli.jar Maintenance -d -id 9dcffe27-77db-4267-a434-ae6b7432e476`

8.11.2. Configuring Server based Maintenances

Server based Maintenances are configured with help of the GUI **Show Server Maintenances view**. For more information see chapter Maintenance Window Management.

There are two possibilities to **create** an server Maintenance:

1. Open the Show **Server Maintenances view -→ new**
2. Open the Hosts view → select Agent → right-click → **Start Maintenance(HIDE)** | **Start Maintenance (DROP)** Creates an server Maintenance with filters = [Agent: <agent>]



8.12. Using custom certificates for built-in web server

By default, the built-in web server which is used e.g. to distribute agent packages runs with the self-signed certificate generated by the boom server. When opening the website in a browser, normally a certificate issue pops up, because the browser does not trust the root CA of the server. Sometimes it is necessary to replace the certificate with a custom one or one provided by e.g. letsencrypt, in order to trust the webpage of the boom server.

Therefore, the keystore.p2 file has to be replaced with a custom keystore file, using keytools utilities. Important: the alias (name) of the certificate in the keystore must be "server".

```
/opt/boom/server/boom_srv -stop
rm -f /opt/boom/server/srv/etc/keystore.p12
cp /tmp/keystore.p12_letsencrypt /opt/boom/server/srv/etc/keystore.p12
```

Change alias /srv/etc/keystore.p12 to <server>

```
keytool -changealias -alias bes-worldline.com -destalias server -keystore
/opt/boom/server/srv/etc/keystore.p12
```

Furthermore, the intermediate certificate of letsencrypt has to be extracted using either command line tools or keytool explorer in Windows. In the example below the intermediate certificate is called R3_ISRG_Root_X1_.cer. Import the exported certificate to srv/etc/truststore.

```
keytool -importcert -file /tmp/R3_ISRG_Root_X1_.cer -keystore /opt/boom/server/srv/etc/truststore.p12
-alias "r3"
```

Delete the old certificate from agents truststore (optional sometimes agents are firewalled without this step).

Afterwards the boom server needs to be restarted.

```
/opt/boom/server/boom_srv -start
```

The keystore of the server does not contain a <ca> cert afterwards and the cakeystore is not used but still containing the old self-singed certificate!

Chapter 9. Command Line Interface

9.1. *boom* Server CLI

The *boom* server comes with a comprehensive command line interface (CLI) which can be found under <boom_server_install_dir>/srv/cli/boom_cli.jar. You can start right away by checking the available commands and options by having a look at the help/usage.

***boom* CLI Usage**

```
java -jar boom_cli.jar -h (prints usage)
java -jar boom_cli.jar -c <Command> -h (prints individual command usage)
java -jar boom_cli.jar -v (prints version)
```

Possible Flags:

-h	General Help
-c <Command> -h	Command specific help
-v	Prints version
-c	Command
-cc	Option
-a	Parameters
-s	Hostname (default: localhost)
-u	Boom user
-p	Password of the specified boom user
-t	Port (default: 23021)
-fs	Path to config file
-x	Output Separator (default: \t)

Supported Commands:

ACTION	Runs a remote action on agent or server
ADD_ANNOTATION	Adds an annotation to a specified indication
ADD_ANNOTATION_FROM_FILE	Adds a file content as annotation to an indication
APPROVE_AGENT	Approves an agent
ARCH_INDI	Archives indications
BOOM_AGENT_PUTFILE	Puts a file to an agent
CLOSE_INDI	Creates an agent
CMD_DISOWN	Disown indications
CMD_OWN	Marks indications as owned
CREATE_AGENT	Creates an agent

CREATE_FILE	Uploads a new file in specified binary package on the server
CREATE_NODEGROUP	Creates a binary package
CREATE_PACKAGE	Path to config file
DELETE_AGENT	Deletes an agent
DELETE_FILE	Deletes a file in specified binary package on the server
DELETE_FOLDER	Deletes a folder in specified binary package on the server
DELETE_INDI	Deletes indications
DELETE_NODEGROUP	Deletes a node group
DEPLOY	Deploys a policy, binary package or assignment group to specified agent
DEPLOY_IF_NOTDEPLOYED	Deploys a policy, binary package or assignment group to specified agent, if not deployed
DISABLE_AGENT	Disables an agent
ENABLE_AGENT	Enables an agent
EXPORT_POLICIES	Exports policies to .xml format
GETAGENTCARDS	Prints agent cards
GETAGENTCARDSALL	Prints agent cards of agents and virtual agents
GETAGENTCARDSEXT	Prints virtual agent cards
GETAGENTID	Lookup an agent ID by specified hostname
GETAGENTIDS4NG	Prints a list of agent IDs for specified node group path (non-recursive)
GETAGENTNODEGROUPS	Prints a list of node groups where specified agent is linked
GET_AGENT_POLICIES	Prints a list of deployed policies for specified agent
GET_AGENTS_STATUS	Prints agents statuses
GET_ALL_ASSIGNMENTS	Returns assignments
GETNODEGROUPS	Prints node groups available on the server
GET_ONLINE_USERS	Prints list of online users
GETPOLICIES	Prints a list of policies available on the server
IMPORT_POLICIES	Imports policies in .xml format
LINK_AGENT	Links agents to a node group
LIST_PACKAGES	Lists binary packages
MARK_AS_DEPLOYED	Marks a policy, binary package or assignment group as deployed on a specified agent
PUSH_PACKAGE	Uploads binary packages to all proxy slave servers
PUT_FILE	Uploads and replaces a file in a specified binary package on the server
RENAME_NODEGROUP	Renames a node group

SEND_MESSAGE	Sends messages to users
SET_AGENT_ATTR	Sets custom attribute values of an agent
SET_CA	Sets custom attribute of a specified indication
UNDEPLOY	Undeploy a policy, binary package or assignment group from specified agent
UNLINK_AGENT	Unlinks agents from a node group

General Syntax of a CLI Command

```
java -jar boom_cli.jar [-x <outputseparator>] -c <command> -cc <option> -a <argument1> -a <argument2> …
```

The exact usage is dependent on the command and can be retrieved by its individual help from the command line.

boom CLI Server Connection

In order to login to a boom server the command line interface needs a boom user with appropriate rights. By default, the CLI will try to connect to a boom server running under localhost. However, any server connection can be specified with the flags specified above.

Create Configuration File
The login credentials can be saved to a file which then can be used to enable the connection for further commands. To create a config file via CLI specify a file path using the -fs flag and provide the connection settings and the login credentials.

```
java -jar boom_cli.jar -s <servername> -u <user> -p <password> -t <port> -fs boom_cli.cfg
```

To load the config file use the same flag with the path to the config file and provide the command and its parameters.

```
java -jar boom_cli.jar -fs boom_cli.cfg -c <command> -cc <option> -a <argument1> -a <argument2> …
```

Please note that passwords provided via command line are visible in the servers history and therefore we recommend the use of a config file. In case this is a security issue for you please contact our support in order to set up a totally encrypted connection.

boom CLI Output Format

By default, the output of any Boom Server Cli command is separated by tabulators. To change the output format use the -x flag.

```
java -jar boom_cli.jar -fs boom_cli.cfg -c <command> -cc <option> -x "\t""
java -jar boom_cli.jar -fs boom_cli.cfg -c <command> -cc <option> -x ";""
```

9.2. *boom* Agent Utilities

The agent comes with a set of utilities (formerly called Agent Commands) that allow a fast data exchange with the agent and to control operations.

boomindi

Allows to submit indications for processing. For details refer Submitting Monitor Values.

boommon

Allows to submit monitor values. For details refer Submitting Indications.

boomperfstore

Allows to submit performance data for history data collection. For details refer Submitting Performance Data.

boomperfreg

Allows to register performance class declarations for later submitting performance data. For details refer Agent command based performance data collection.

boomgetvar

Allows to retrieve agent variables that are used e.g. in conditions for policy activation.

Input:

[varname] - specify zero or one variable name

Output:

Prints the list of all pairs of <varname>=<value> or the selected one. The error VARIABLE_NOT_DEFINED will be displayed if the specified variable does not exist. The return value will be 0 for success and 1 for error.

e.g.

```
/opt/boom/agent/spi/>boomgetvar

IPCMD=yes
JOURNALCTL=no
MPSTATCMD=no
NETSTATCMD=yes
SARCMD=no
TOPCMD=yes
```

boomremvar

Allows to remove agent variables.

Input:

<varname> - specify one variable name

Output:

Prints the result of the operation. The return value will be 0 for success and 1 for error. E.g.

```
/opt/boom/agent/spi/>boomremvar TESTVAR

var 'TESTVAR' removed
or in case of error:
var 'TESTVAR' not found
```

boomsetvar

Allows to set the value of an agent variable.

Input:

<varname>=<value>

Output:

No output, return value of the command will be 0. In case of an syntax error the usage will be displayed and the return value will be 1.
`/opt/boom/agent/spi/>boomsetvar TESTVAR=100`

boomcmd

Allows to execute an agent command. Refer to the available Agent Actions Overview to get a list of possible agent commands.

Input:

cmd [option]

Output:

Prints the results of the operation or if some error occurred an error message indicating the error cause. The return value will be 0 for success and 1 for error. E.g.

```
/opt/boom/agent/spi/>boomcmd GET_LOGLEVEL
current loglevel = 5
```

boomstartmt

Allows to start an AdHoc Maintenance from the agent. The Agent will inform the *boom* server, that a maintenance has been started for indications matching the defined filters. The end of the maintenance can be triggered by boomstopmt command or automatically by specifying "–du" option.
An *boom* administrator can forcibly end the maintenance from the *boom* UI.

Input:

`–ed <ExpectedDurationInterval> [-du <MaxDurationInterval>] -ac {HIDE [–pa {NONE|DROP|PUBLISH}] | DROP} <attribute>="<SimplifiedPattern>" [ <attribute>="<SimplifiedPattern>"]*`

- -ed <ExpectedDurationInterval> - expected duration of the maintenance. An Indication will be triggered by server to inform if expected duration is overdue.
- -du <MaxDurationInterval> - MAX duration of the maintenance. The server will be informed to disable this maintenance after specified interval, in case boomstopmt command was not performed before.

  Supported interval formats:
  ?w?d?h?m - a combination of weeks, days, hours and minutes (Minimum interval: 1m).

  Examples:
  2d3h4m - 2 days, 3 hours, 4 min
  3d5h - 3 days and 5 hours
  100m - a hundred minutes

- -ac <action> - required action on the server for matched indications during maintenance.
  actions:
  HIDE - Indications need to be displayed in the Maintenance browser
  DROP - Indications must be dropped by the server
- -pa <postAction> - Valid only with options "-ac HIDE": Action to be performed after the end of the maintenance.

  postActions:

  NONE - the indications will stay in the Maintenance browser after the Maintenance is ended
  DROP - the indications will be dropped from the Maintenance browser at the end of the maintenance window
  PUBLISH - the indications will be moved to the active browser at the end of the maintenance window

- <attribute>="<SimplifiedPattern>" - one or more filters, based on attribute name and simplified patterns to match indications targeted for maintenance processing.

  Supported attributes:

  APPLICATION, GROUP, OBJECT, HOST (alias: NODE), CA, CA1 … CA15, NODEGROUP, SOURCE, TEXT, SEVERITY, KEY

  Examples:
  APPLICATION="ORACLE<*>" - matches any indications with attribute Application started with ORACLE
  SEVERITY="<[critical|major]>" - matches any indications with severity critical or major

TEXT="<*>" - matches any indications

Output:

Prints the UUID of the created maintenance or if some error occurred an error message indicating the error cause. The return value will be 0 for success and 1 for error. E.g.

```
boomstartmt -ed 2h -du 2h30m -ac HIDE -pa PUBLISH SEVERITY="<*>"
79142341-b492-45ff-b11c-48c309c8072c
```

boomstopmt

Input:

<UUID of the Maintenance>

Output:

Prints the UUID of the stopped maintenance or if some error occurred an error message indicating the error cause. The return value will be 0 for success and 1 for error. E.g.

```
boomstopmt 79142341-b492-45ff-b11c-48c309c8072c
79142341-b492-45ff-b11c-48c309c8072c
```

Chapter 10. Copyright & Trademarks

Third-Party Software Information English / English Note to Resellers: Please pass on this document to your customers to avoid license infringements. Third-Party Software Information This product, solution or service ("Product") contains third-party software components listed in this document. These components are Open Source Software licensed under a license approved by the Open Source Initiative (www.opensource.org) or similar licenses as determined as ("OSS") and/or commercial or freeware software components. With respect to the OSS components, the applicable OSS license conditions prevail over any other terms and conditions covering the Product. The OSS portions of this Product are provided royalty-free and can be used at no charge.

Open Source Software and/or other third-party software contained in this Product: Please note the following license conditions and copyright notices applicable to Open Source Software and/or other components (or parts thereof) BOOM.

Technical use of OSS within the *boom* development - Programming language and runtime is Java - 3rd party components/libraries have not been modified (no own compiling or change of 3rd party components/libraries source code) - Functionalities of 3rd party components/libraries are from *boom* code only referred and used with calling methods or inheritance - *boom* Code (=own code) is provided in compiled form in own libraries, which containing exclusively *boom* Code (boom libraries) - 3rd party components/libraries and *boom* libraries are separated dynamically linked units (Java Jars + Wars) which are loaded and used at runtime in Java Classloader within a single, common Java JVM process.

Software Component	License
Apache Commons CSV	Apache License 2.0
Apache Commons Email	Apache License 2.0
Apache Log4J	Apache License 2.0
Apache Xerces2 J	Apache License 2.0
Bootstrap	MIT License
D3.js	BSD 3
DataTables	MIT License
exp4j	Apache License 2.0
Font-Awesome	MIT License
google-gson	Apache License 2.0
Draw2D	Eclipse Public License 1.0
Eclipse RCP	Eclipse Public License 1.0
jackson-core	Apache License 2.0

Software Component	License
Jakarta Mail	Common Development and Distribution License 1.1
Java Servlet API	Common Development and Distribution License 1.1
Java Mail API	Common Development and Distribution License 1.1
JCommon	GNU Lesser General Public License v2.1
Jericho HTML Parser	Eclipse Public License 1.0
Jetty	Eclipse Public License 1.0
JFreeChart	GNU Lesser General Public License v2.1
jQuery	MIT License
JSch	JSch License
Grammatica Java	BSD 3
MySQL Connector/J	GNU General Public License v2.0
Netty Project	Apache License 2.0
NVD3.js	Apache License 2.0
Oracle Database JDBC Drivers	FUTC License
org.eclipse.zest.layouts	Eclipse Public License 1.0
org.talend.libraries.jfreechart.jars	Apache License 2.0
SLF4J API Module	MIT License
SLF4J Binding	Apache License 2.0
SNMPj4	Apache License 2.0
trilead-ssh2	BSD 3
xercesImpl	Apache License 2.0

Java and all Java-based trademarks are trademarks of Oracle in the United States, other countries, or both.

HP and Operations Manager are registered trademarks of the Hewlett-Packard Company in the United States and other jurisdictions.

Linux is a U.S. registered trademark of Linus Torvalds.

Microsoft, Windows, Windows 2000, Windows Server 2003, Windows NT and WMI are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

MySQL is a registered trademark of 2000-2008 MySQL AB, 2008 Sun Microsystems, Oracle. in the United States, the European Union and other countries. MySQL Network is a trademark of 2000-2008 MySQL AB, 2008 Sun Microsystems.

All SPARC trademarks are trademarks or registered trademarks of SPARC International, Inc. in the United States and other countries.

UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd.

Chapter 11. Contact

equensWorldline SE Germany
Headquarter:
Lyonerstraße 15
60528 Frankfurt
Germany
Phone: +49 69 6657-10
Fax: +49 69 6657-1211

https://worldline.com

boom:
Email: support@bes-worldline.com
Phone: +49 69 256 552 201