

PassMark® Software Management Console

Quick Start Guide



Release: 2.0.1000
Date: 20 September 2022

Table of Contents

Overview	3
Cloud Subscription Setup.....	4
PassMark Products Supported.....	4
Requirements.....	4
BurnInTest – Management Console Cloud Setup.....	4
Check Subscription Status.....	5
Troubleshooting.....	5
On-Premise Setup	6
PassMark Products Supported.....	6
Software Requirements	6
Hardware Recommendations	6
Setup Example	6
Installation	6
Update existing database	7
BurnInTest – Management Console On-Premise Setup	7
MemTest86 – Management Console On-Premise Setup	8
Notes.....	9
Troubleshooting.....	9
Management Console Usage.....	10
Summary.....	10
Details	11
Reports.....	13
Configuration	14
Support	15
Pricing	15
Appendix A – Creating the Management Console database	16

Overview

When testing large numbers of systems, the Management Console Web server application allows system information, test status and test result information to be managed centrally for BurnInTest and MemTest86 tests.

Via a browser the Management Console allows information about tests to be displayed. This includes the status of currently running tests, system information, current and previous test results, test reports and test statistics.

There are two available options: Subscription and On-Premise install. A comparison of the two options below:

	On-Premise	Subscription
Management Level View of BurnInTest System Testing	O	O
Management Level View of MemTest86 System Testing	O	X
Live View of Current Tests	O	O
Centralized Test Reporting	O	O
Centralized Statistical Information	O	O
Centralized Historical Test Records	O	O
Responsible for Installation and Set Up	You	PassMark
Who supplies Database Server Hardware	You	PassMark
Upgrade and Patch Installation	You	PassMark
License Term	Perpetual	1 Month
Support and Upgrades	12 Months	1 Month

On-Premise Setup

PassMark Products Supported

Management Console supports the following products and versions:

BurnInTest Windows	8.1.1000 or newer
BurnInTest Linux	4.0.1000 or newer
MemTest86	7.5 or newer

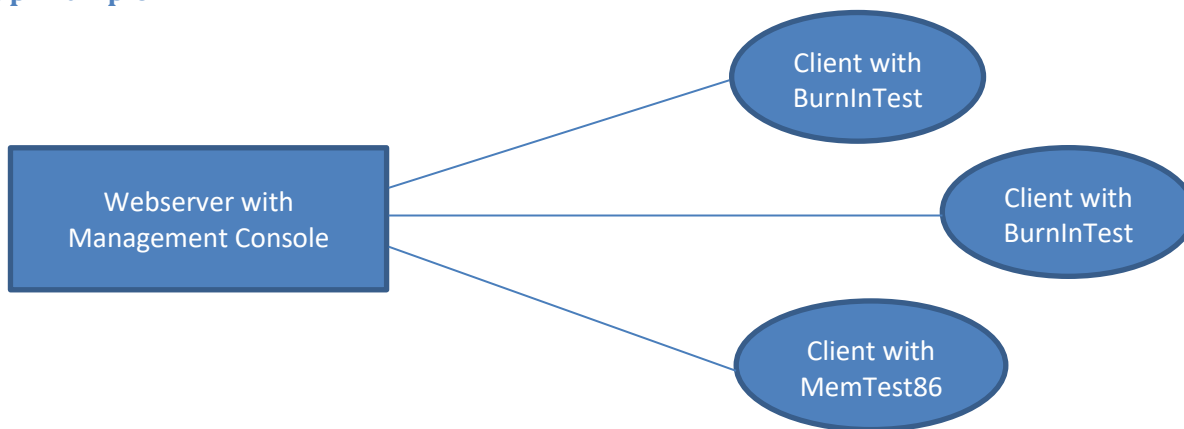
Software Requirements

- A web server running PHP (5.3.2 or later) and MySQL (5.2.47 or later).
- If using MemTest86, Python scripting support (3.6.3 or later) and a PXE server setup to boot MemTest86 across the network is required. The systems running MemTest86 must be able to PXE boot to UEFI.

Hardware Recommendations

- 4 core CPU
- 16GB RAM
- Fast SSD

Setup Example



Installation

If you are installing the Management Console on an existing Web server with PHP and MySQL, skip to step 6.

1. Install Apache 2.2.14 or later
2. Install PHP 5.3.2 or later
Note 1: When installing PHP, select Web server setup “Apache 2.2.x module” (and install everything).
Note 2: To install MySQL, you need to install PHP with extensions (to get ext\php_mysql.dll)
3. Install MySQL 5.2.47 or later
4. Configure Apache (httpd.conf):
 - a. Create a symbolic link in the apache installation folder to ‘htdocs’, e.g.:

```
Alias /htdocs "C:\Passmark\Software\BIT Console\mgtconsole"
```

```
<Directory "C:\Passmark\Software\BIT Console\mgtconsole">
  Options Indexes FollowSymLinks
  AllowOverride all
  Order allow,deny
  Allow from all
</Directory>
```

- b. Configure the PHP installation, e.g.:

```
LoadModule php5_module "C:/Program Files (x86)/PHP/php5apache2_2.dll"
AddType application/x-httpd-php .php
PHPIniDir "C:\Program Files (x86)\PHP"
```

5. Configure PHP (PHP.ini):
 - a. Set the default Time zone in PHP, e.g.:


```
[Date]
; Defines the default time zone used by the date functions
; http://php.net/date.timezone
date.timezone =Australia/Sydney
```
 - b. Configure the MySQL extension: extension=php_mysql.dll
6. Create the Management Console database using the MySQL administration tools using the SQL shown in “Appendix A – Creating the Management Console database”. There is also an SQL script included in the install files called “database_creation_script.sql” that can be used to create the initial database.
7. Copy the PassMark supplied Management Console files to the Web server, e.g., to htdocs. Set the error log file, ManagementConsole-errors.log, as writeable to all.
8. Setup the Management Console connection to MySQL:
 - a. The PHP file, settings.php, contains the host name, port and MySQL password. Open the settings.php file in a text editor and change the \$DATABASE_HOSTNAME, \$DATABASE_USERNAME, \$DATABASE_PASSWORD and \$DATABASE_PORT values to suit your MySQL setup.
9. You should now be able to access the dashboard via <http://<server address>/dashboard.php> (for example)

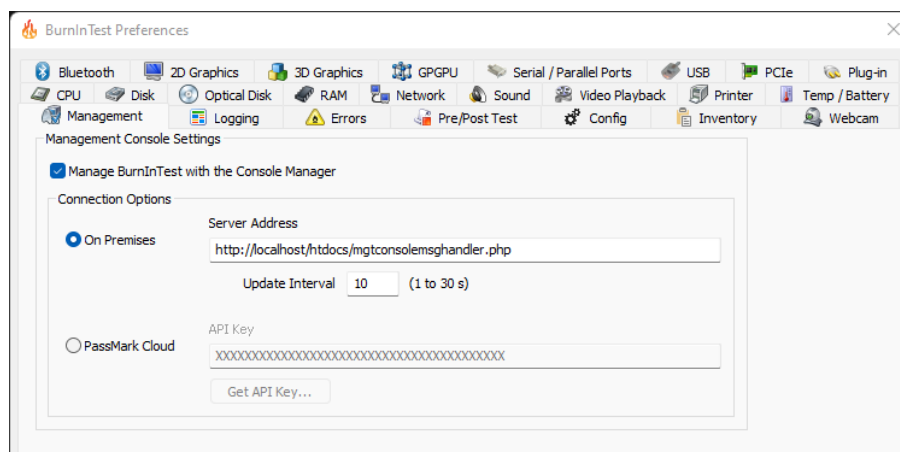
Update existing database

Sometimes columns will be added to existing tables and require existing databases to be modified.

If you previously installed Management Console and are updating to a new build, then please see the database_update.sql script and the Readme.txt file to see if any changes have occurred and if the update script needs to be run.

BurnInTest – Management Console On-Premise Setup

To manage BurnInTest with the Management Console, the following must be selected and setup in BurnInTest: Configuration > Test Preferences > Management



Manage BurnInTest with the Console Manager

When this option is selected, BurnInTest will send system information (on startup), test status (every 1-30 seconds) and test result information (at the completion of each test run) to the Management Console Web server application.

Server address

Specifies the address of the Management Console Web server application. For example:

<http://<server address>/htdocs/mgtconsolemsghandler.php>

Notes

- `mgtconsolemsgshandler.php` is the Management Console web application that receives XML based connect, status and test result information from BurnInTest (via HTTP POST messages).
- Configuration->Report Information->Machine ID is used to uniquely identify the system in the Management Console. The MachineID is a unique identifier automatically generated and saved by BurnInTest for the system and must be between 8 and 15 characters. While it can be modified by a user, this is not recommended. If the MachineID is changed, then this system will appear as a different system in the Management Console.

MemTest86 – Management Console On-Premise Setup

For MemTest86 to communicate with the Management Console it needs to have access to the network while running, currently to do this MemTest86 needs to boot over the network via PXE. It can then upload status messages via TFTP to a folder on the PXE server. This folder is watched by a Python script for changes, on detection of a new file it will then send the XML contents of the file to the Management Console via a HTTP POST request.

Setting up the PXE Server

To configure PXE booting of MemTest86, a DHCP/PXE server must be present on the network to host the MemTest86 boot image for PXE boot-enabled client machines to acquire. Network booting of MemTest86 has been tested successfully with Serva PXE Server but other PXE servers should work as well. See the manual for your DHCP/PXE server for configuration instructions. The configuration instructions for Serva PXE Server are included in the following section.

Once the PXE server is configured, extract the files from the MemTest86 package to the appropriate directory for your PXE server configuration. For most cases, this is the TFTP root directory configured in the TFTP server. In the PXE/DHCP server settings, specify the boot image file to “BOOTX64.efi” for x86-64 client machines and “BOOTIA32.efi” for x86 client machines.

On the client machine, the UEFI BIOS must support booting from the network. In the BIOS setup, ensure that the “UEFI Network Stack” and “IPv4 PXE Support” features are enabled. If the PXE Server was successfully set up, the client machines should automatically boot MemTest86 on power-up.

Setting up and running the Python script

The [Python scripting environment](#) must be installed on the system used for the PXE boot. A Python script (`memtest_status_watcher.py`) needs to be running to transfer messages from the TFTP upload folder (where Memtest86 uploads its status messages) to the Management Console PHP files.

It contains several user configurable options; they can be edited by opening the file in a text editor.

MANAGEMENT_CONSOLE_URL

This is the address of the Management Console server e.g., <http://localhost/mgtconsole/mgtconsolemsgshandler.php>

USER

Username for web server authentication if required (leave empty if no authentication)

PASSWD

Password for web server authentication if required (leave empty if no authentication)

WATCHDIR

Directory to watch for files uploaded from MemTest86 (tftp server upload directory)

This script requires the “requests”, “watchdog” and “queues” Python libraries be installed, which can be done using the pip command e.g., “pip install requests” (on windows pip.exe is in the scripts directory of the Python install directory).

Configuring Serva for MemTest86 PXE Boot

Serva is a lightweight but powerful Windows PXE server that bundles all required services (e.g., DHCP, TFTP) to support UEFI-based network booting. Serva does not require an installation and can be setup in minutes.

Configuring Serva for Single-Image Boot is ideal for servers that require only a simple setup and do not need to distribute software images other than MemTest86. All necessary settings are configured within the Serva application and do not require any additional configuration files.

1. Open Serva and select 'Settings'
2. Click on the TFTP tab to setup the TFTP server
 - a) Ensure that 'TFTP Server' is checked
 - b) Specify the TFTP root directory. This should be the location where the files in the MemTest86 are to be extracted.
 - c) Set the TFTP Security to 'Standard' to allow MemTes86 report files to be uploaded to the server
3. Click on the DHCP tab to setup the DHCP server
 - a) If your network already has a DHCP server, check 'proxyDHCP'. Otherwise, check 'DHCP Server'.
 - b) If 'DHCP Server' is selected, specify the 'IP Pool 1st Addr', 'Pool size' and 'Subnet Mask' for the DHCP server.
 - c) Specify the 'Boot File' to be retrieved by the client. For 64-bit clients (most systems), enter 'BOOTX64.efi' as the boot file. For 32-bit clients, enter 'BOOTIA32.efi'
4. Press OK to save the settings.
5. Extract all files in the MemTest86 package in the folder specified in Step 2b.
6. Close and restart Serva to apply the settings.

Notes

1. Management Console errors are logged in the web server file: ManagementConsole-errors.log
2. All date/times stored in the database are UTC offsets from January 1, 1970. When viewing date/times on the Management Console the date/times are adjusted for the time zone of the web server. If the web server time zone is different to the time zone of the systems under test, then it should be remembered that the times are not the test system's local time, but the Web server's local time.

Troubleshooting

General Debugging

By adding the parameter “debugmode=1” to the dashboard address PHP errors and warnings will be enabled that may help troubleshoot any problems, e.g., <http://localhost/mgtconsole/dashboard.php?debugmode=1>

Errors can be logged to the ManagementConsole-errors.log file which can help debug database connection errors.

MySQL Database errors

If you see an error like “Warning: mysqli::mysqli(): (HY000/1045): Access denied for user 'username'@'localhost' (using password: YES)” when you navigate to the dashboard.php page then you may not have not setup the required username and password for the MySQL database. Open the settings.php file in a text editor and change the \$DATABASE_HOSTNAME, \$DATABASE_USERNAME, \$DATABASE_PASSWORD and \$DATABASE_PORT values to suit your MySQL setup.

Management Console Usage

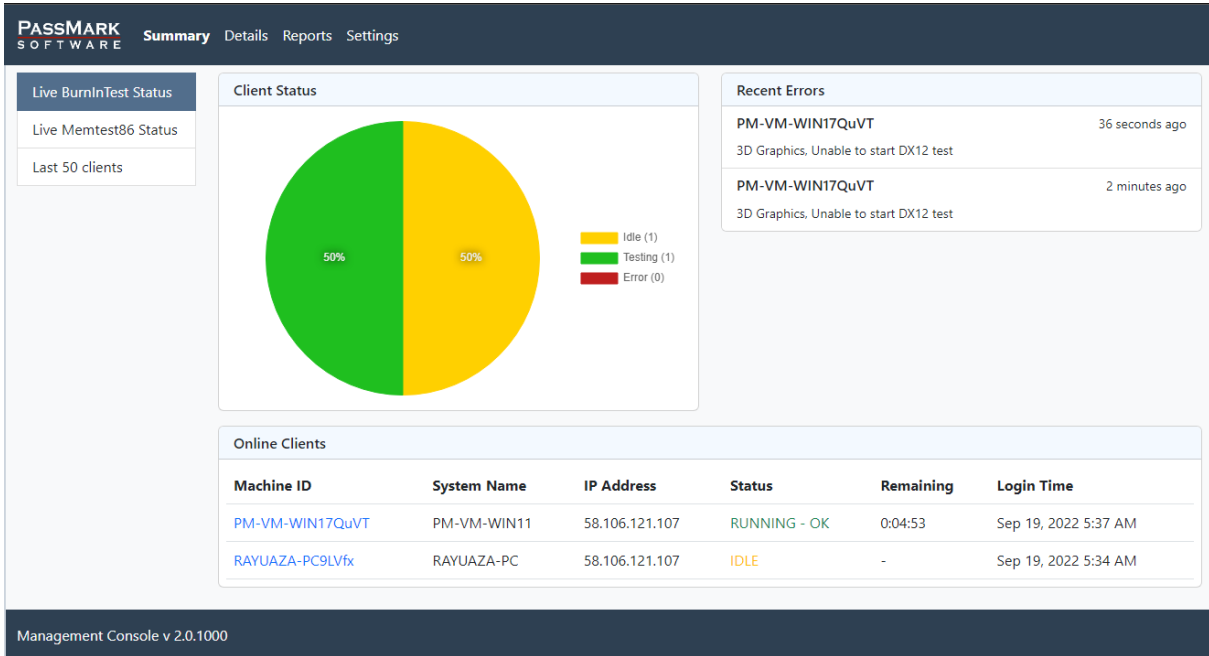
The Management Console user interface (dashboard.php) has 4 main sections: Summary, Details, Reporting and Configuration.

Summary

The Summary section display an overview of online BurnInTest and MemTest86 clients.

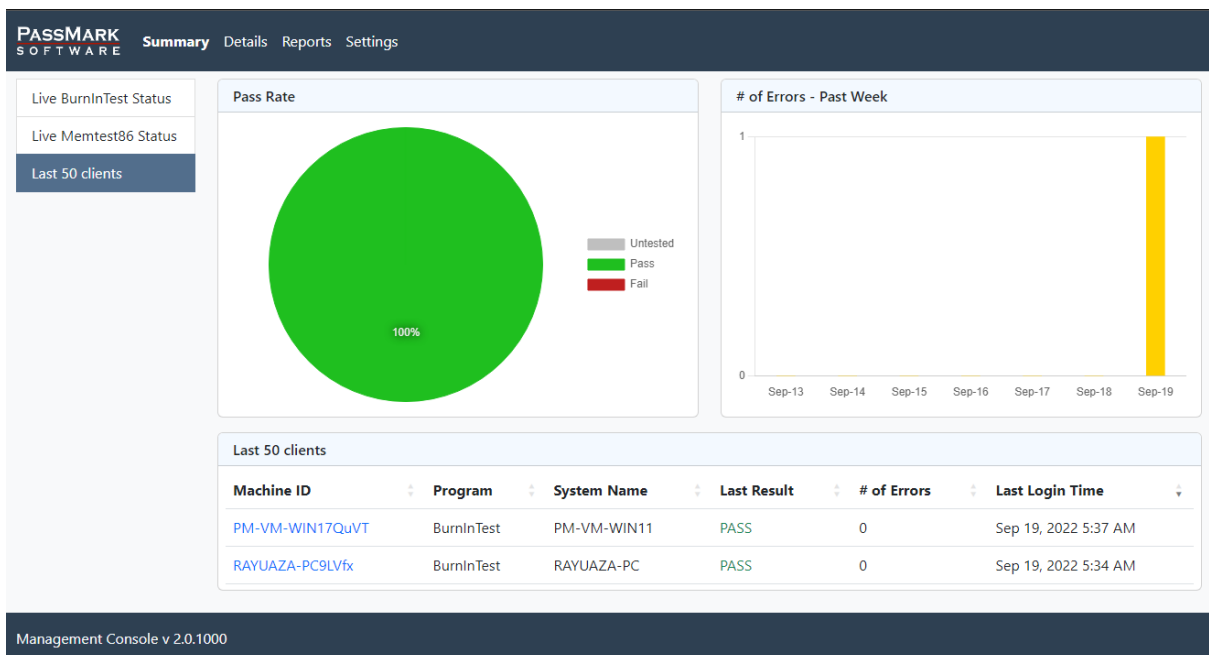
Live BurnInTest/Memtest86 Status

Displays graph of status of connected clients, list of recent errors and a list of all connected clients.



Last 50 Clients

Displays information of the last 50 clients that connected to Management Console and the number of errors in the past week.



Details

The Details section provides information for a selected test system, including the Current Status, System Information, Last Test Result, and a searchable results history and per test run deletion.

Clicking in the search icon on the top right opens a search box to allow searching for machines not in the dropdown.

The screenshot shows the PASSMARK SOFTWARE Management Console interface. At the top, there are navigation tabs: Summary, **Details**, Reports, and Settings. Below the navigation, a search bar contains the Machine ID "PM-VM-WIN17QuVT". A "Find Machine ID" section includes input fields for Machine ID, System Name, and Date Range (From and To). Below these fields are "Filter" and "Clear" buttons. A table shows the search results with columns for Machine ID, System Name, and Last Login Date. The table contains two entries: "PM-VM-WIN17QuVT" (System Name: PM-VM-WIN11, Last Login Date: Sep 19, 2022 5:27 AM) and "RAYUAZA-PC9LVfx" (System Name: RAYUAZA-PC, Last Login Date: Sep 19, 2022 5:26 AM). Below the table, there are "Previous", "1", and "Next" navigation buttons. On the left, a sidebar shows "Current Status", "System Information", and "Results History". The main content area displays "Status: OFFLINE". At the bottom, it says "Management Console v 2.0.1000".

Current Status

Displays status of the selected client, additional information is displayed when machine is running a test.

The screenshot shows the PASSMARK SOFTWARE Management Console interface. At the top, there are navigation tabs: Summary, **Details**, Reports, and Settings. Below the navigation, a search bar contains the Machine ID "PM-VM-WIN17QuVT". On the left, a sidebar shows "Current Status", "System Information", and "Results History". The main content area displays "Status: RUNNING - OK" in a green banner. Below the banner, it says "Last Updated: 0:00:08 ago Refresh". A "BurnInTest Results" section shows the following information: Test Start Time: Sep 19, 2022 5:37 AM; Duration: 0:03:08; Remaining time: 0:01:52. Below this is a table with columns: Test, Cycles, Operations, Errors, and Last Error Description. The table contains one row: CPU, 10, 301 Billion, 0, No errors. Below the table is an "Event Log" section with three entries: "Sep 19, 2022 5:37 AM, LOG NOTE, Status, Main Tests started", "Sep 19, 2022 5:37 AM, LOG NOTE, Status, Using Management Console [machineID: PM-VM-WIN17QuVT Session ID: 5]", and "Sep 19, 2022 5:37 AM, LOG NOTE, BurnInTest, BurnInTest Management Console - Session ID: 5 - Result ID: 37". At the bottom, it says "Management Console v 2.0.1000".

System Information

The system information for the system is stored each time a system connected to the Management Console so hardware changes will be stored across test runs. This will display the system information from the most recent test session.

The screenshot shows the 'System Information' page in the Management Console. The page title is 'PASSMARK SOFTWARE' with navigation links for 'Summary', 'Details', 'Reports', and 'Settings'. The 'Machine ID' is 'PM-VM-WIN17QuVT'. The left sidebar has 'Current Status', 'System Information' (selected), and 'Results History'. The main content area is titled 'System Summary' and contains the following information:

Operating System	Windows 11 Home build 22000 (64-bit)
CPU	1 x AMD Ryzen 9 5900X 12-Core Processor [3701.2 MHz]
Memory	8GB RAM,
Graphics	VMware SVGA 3D,
Hard Disk	30.00GB VMware Virtual NVMe Disk, 80.00GB VMware Virtual NVMe Disk, 1.00GB VMware Virtual NVMe Disk
Optical Disk	DVD,
General	
System Name	PM-VM-WIN11
System Model	VMware7,1
Motherboard Manufacturer	Intel Corporation
Motherboard Name	440BX Desktop Reference Platform
Motherboard Version	None
BIOS Manufacturer	VMware, Inc.

Results History

Displays list of last 10 results, clicking on the test date or event log links would display further details of the test run in a new window. Results can be filtered by several means above the results list. Users can also delete individual results by clicking on the check box and then on the Delete Selected button.

The screenshot shows the 'Results History' page in the Management Console. The page title is 'PASSMARK SOFTWARE' with navigation links for 'Summary', 'Details', 'Reports', and 'Settings'. The 'Machine ID' is 'PM-VM-WIN17QuVT'. The left sidebar has 'Current Status', 'System Information', and 'Results History' (selected). The main content area is titled 'Filter Results History' and contains the following information:

Filter Results History

Date Range: From 2021-Sep-19 To 2022-Sep-19

Components tested: Any

Result: Any

Filter

Last 10 results displayed, use filter settings to see specific results.

Test Date	Test Duration	# of Errors	Result	Delete
Sep 19, 2022 5:37 AM	0:05:11	0 (Event Log)	PASS	<input type="checkbox"/>
Sep 19, 2022 5:35 AM	0:01:11	1 (Event Log)	FAIL	<input type="checkbox"/>
Sep 19, 2022 5:31 AM	0:03:45	0 (Event Log)	PASS	<input type="checkbox"/>

Delete Selected

Management Console v 2.0.1000

Reports

The Reports section provides statistical reports across all tested systems, filtered by date, system, component type or customer.

Graphs

Includes Errors vs. time, Failures vs. time, Tests performed vs. time and the Pass rate. Report parameters can be adjusted at the top. Print option at the bottom opens a separate window to facilitate printing the report.

The screenshot displays the PASSMARK SOFTWARE interface. At the top, there are navigation tabs: Summary, Details, Reports (selected), and Settings. On the left, there are two main sections: 'Graphs' and 'Lists'. Under 'Graphs', there are sub-sections: Overall Reports (selected), System Reports, Component Reports, and Customer Reports. Under 'Lists', there are sub-sections: RAM, HDD, and CPU. The main content area is titled 'Report Parameters' and includes the following fields:

- Date Range: From 2022-Sep-1 to 2022-Sep-30
- Report Type: Errors vs Time
- Program Type: BurnInTest

A 'Generate Report' button is located below these fields. Below the parameters is the 'BurnInTest Report' section, which includes a 'Report Overview' table:

Report Date	Sep 19, 2022 5:46 AM
Report Type	Errors vs Time
Date Range	2022-Sep-1 to 2022-Sep-30

Below the overview is a 'Chart' section. The chart shows a single yellow bar representing the number of errors on Sep-19, with a value of 1. The x-axis is labeled with dates from Sep-01 to Sep-29, and the y-axis ranges from 0 to 1.

Below the chart is a 'Statistics' table:

Number of Clients	2
Number of Tests Performed	4
Average # Errors / day	0.03
Most Errors Date	2022-Sep-19 (1 errors)
Total # of Errors	1

A 'Print...' button is located at the bottom of the report section. At the very bottom of the page, there is a footer: 'Management Console v 2.0.1000'.

Lists

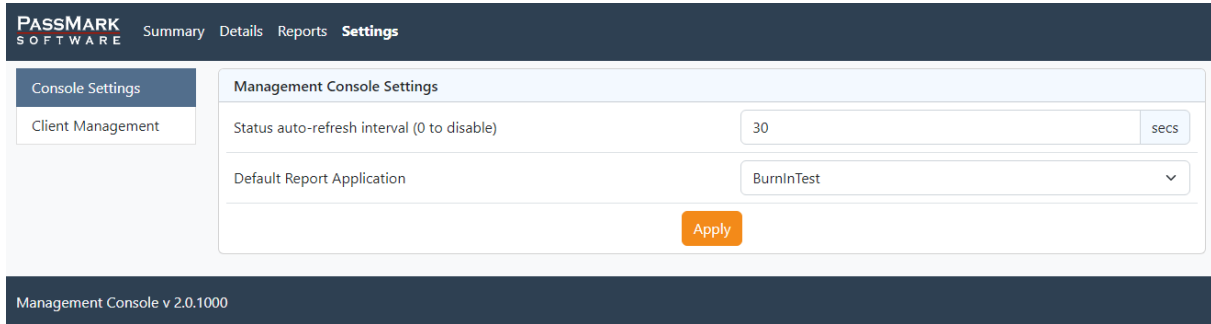
Displays a list of RAM, HDD or CPUs used for each test run with report parameters applied. Print option at the bottom opens a separate window to facilitate printing the report.

Configuration

The Configuration section allows the auto refresh of the live Management Console status web pages to be set. It also allows systems to be deleted from the Management Console database.

Console settings

Allows user to adjust auto-refresh interval in the Summary and Details > Current Status sections of the dashboard. You can also set the default report application here which affects what appears by default in the Report > Report Parameters > Program Type dropdown (On-Premise only).



PASSMARK SOFTWARE Summary Details Reports **Settings**

Console Settings
Client Management

Management Console Settings

Status auto-refresh interval (0 to disable) secs

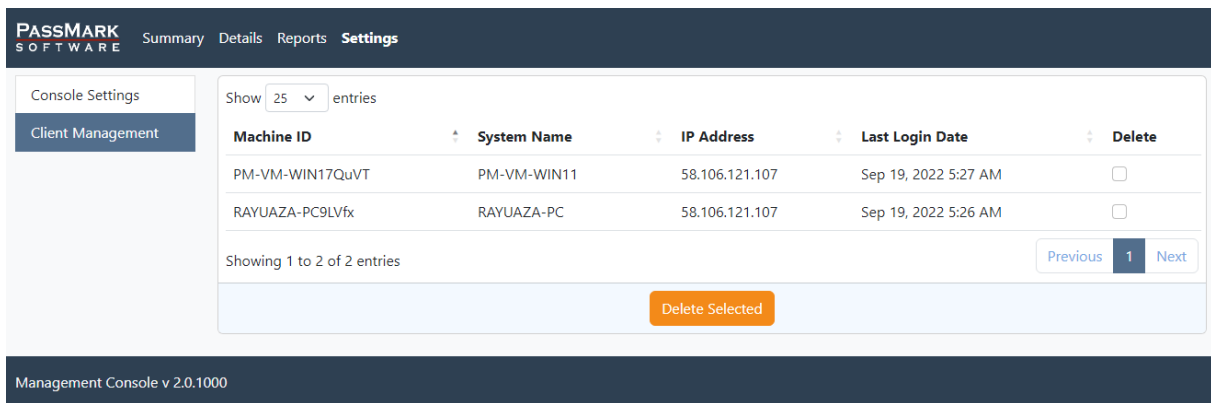
Default Report Application

Apply

Management Console v 2.0.1000

Client Management

Allows user to delete any systems and associated test results from the Management Console database.



PASSMARK SOFTWARE Summary Details Reports **Settings**

Console Settings
Client Management

Show 25 entries

Machine ID	System Name	IP Address	Last Login Date	Delete
PM-VM-WIN17QuVT	PM-VM-WIN11	58.106.121.107	Sep 19, 2022 5:27 AM	<input type="checkbox"/>
RAYUAZA-PC9LVfx	RAYUAZA-PC	58.106.121.107	Sep 19, 2022 5:26 AM	<input type="checkbox"/>

Showing 1 to 2 of 2 entries

Previous 1 Next

Delete Selected

Management Console v 2.0.1000

API Key (Cloud Subscription only)

Allows user to view their API Key which can be entered into BurnInTest to connect to Management Console Cloud.

Support

For technical support, questions, suggestions, please check our support page:

<https://www.passmark.com/support/index.php>

Pricing

Visit our pricing page:

<https://www.passmark.com/products/bitmgtconsole/price.php>

Appendix A – Creating the Management Console database

```
CREATE DATABASE `passmark_mgtconsole`;
```

```
CREATE TABLE `passmark_mgtconsole`.`test_session` (  
  `session_id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `machineid` VARCHAR(24) CHARACTER SET latin1 COLLATE latin1_general_cs NOT NULL,  
  `session_timestamp` INTEGER NULL ,  
  `application_type` tinyint(1) DEFAULT '0',  
  `status` VARCHAR(16) NULL ,  
  `timestart` INTEGER NULL ,  
  `timeupdate` INTEGER NULL ,  
  `duration` INTEGER UNSIGNED NULL ,  
  `remaining` INTEGER UNSIGNED NULL ,  
  `ip` VARCHAR(16) NULL ,  
  `os` VARCHAR(100) NULL ,  
  `cpu` VARCHAR(100) NULL ,  
  `memory` VARCHAR(16) NULL ,  
  `graphics` VARCHAR(100) NULL ,  
  `opticaldisk` VARCHAR(100) NULL ,  
  `sysname` VARCHAR(100) NULL ,  
  `sysmodel` VARCHAR(100) NULL ,  
  `mbmanufacturer` VARCHAR(100) NULL ,  
  `mbname` VARCHAR(100) NULL ,  
  `mbversion` VARCHAR(100) NULL ,  
  `biosmanufacturer` VARCHAR(100) NULL ,  
  `biosversion` VARCHAR(100) NULL ,  
  `biosdate` VARCHAR(100) NULL ,  
  `version` VARCHAR(32) NULL ,  
  `customer` VARCHAR(32) NULL ,  
  `technician` VARCHAR(32) NULL ,  
  PRIMARY KEY (`session_id`));
```

```
CREATE TABLE `passmark_mgtconsole`.`sysinfo_details` (  
  `session_id` int(11) unsigned NOT NULL,  
  `title` VARCHAR(32) NULL ,  
  `description` VARCHAR(150) NULL ,  
  KEY (`session_id`));
```

```
CREATE TABLE `passmark_mgtconsole`.`hdd_details` (  
  `session_id` int(11) unsigned NOT NULL,  
  `model` VARCHAR(32) NULL ,  
  `serial` VARCHAR(120) NULL ,  
  `size` VARCHAR(32) NULL ,  
  KEY (`session_id`));
```

```
CREATE TABLE `passmark_mgtconsole`.`ram_details` (  
  `session_id` int(11) unsigned NOT NULL,  
  `name` VARCHAR(32) NULL ,  
  `model` VARCHAR(64) NULL ,  
  `serial` VARCHAR(64) NULL ,  
  `size` VARCHAR(16) NULL ,  
  `date` VARCHAR(32) NULL ,  
  `manufacturer_specific` VARCHAR(64) DEFAULT NULL,  
  KEY (`session_id`));
```

```
CREATE TABLE `passmark_mgtconsole`.`bit_results` (  
  `result_id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `session_id` int(11) unsigned NOT NULL,  
  `timestart` INTEGER NOT NULL ,  
  `timeupdate` int(11) DEFAULT NULL ,  
  `timestop` int(11) DEFAULT NULL ,  
  `duration` int(10) unsigned DEFAULT NULL,  
  `result` VARCHAR(16) NULL ,  
  PRIMARY KEY (`result_id`),  
  KEY (`session_id`, `result_id`));
```



```
CREATE TABLE `passmark_mgtconsole`.`memtest_results` (
  `result_id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `session_id` int(11) unsigned NOT NULL,
  `timestart` INTEGER NOT NULL ,
  `timeupdate` int(11) DEFAULT NULL ,
  `timestop` int(11) DEFAULT NULL ,
  `duration` int(10) unsigned DEFAULT NULL,
  `result` VARCHAR(16) NULL ,
  PRIMARY KEY (`result_id`),
  KEY (`session_id`,`result_id`));
```

```
CREATE TABLE `passmark_mgtconsole`.`events` (
  `session_id` int(11) unsigned NOT NULL,
  `result_id` int(11) unsigned NOT NULL,
  `timestamp` INTEGER NULL ,
  `level` VARCHAR(16) NULL ,
  `type` VARCHAR(16) NULL ,
  `description` VARCHAR(256) NULL ,
  KEY (`session_id`,`result_id`));
```

```
CREATE TABLE `passmark_mgtconsole`.`testlist` (
  `session_id` int(11) unsigned NOT NULL,
  `result_id` int(11) unsigned NOT NULL,
  `test` varchar(50) NOT NULL DEFAULT '',
  `cycles` int(11) DEFAULT NULL,
  `operations` bigint(20) DEFAULT NULL,
  `result` varchar(16) DEFAULT NULL,
  `errors` int(11) DEFAULT NULL,
  `lasterror` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`result_id`,`session_id`,`test`));
```

```
CREATE TABLE `passmark_mgtconsole`.`configuration` (
  `setting` VARCHAR(16) NOT NULL ,
  `value` INTEGER NULL ,
  `string` VARCHAR(256) NULL ,
  KEY (`setting`));
```

```
CREATE TABLE `passmark_mgtconsole`.`memtest_reports` (
  `result_id` int(11) NOT NULL,
  `session_id` int(11) NOT NULL,
  `memory_range` varchar(45),
  `cpu_selection` varchar(40),
  `cpu_temp` varchar(20),
  `ram_temp` varchar(20) ,
  `tests_passed` varchar(20),
  `erroraddr_low` varchar(45),
  `erroraddr_high` varchar(45),
  `error_bitmask` varchar(20) ,
  `error_numbits` varchar(20) ,
  `errors_maxcont` varchar(20) ,
  `err_cpus` varchar(20) ,
  PRIMARY KEY (`result_id`),
  KEY (`session_id`));
```

```
#You may need to grant privileges to your user
#GRANT all on passmark_mgtconsole.* to user;
```