

PAWZ AGENT V9.0A

Release Notes

June 2009

Revision/Update:

Version 9.0A is a feature and fix release.

PerfCap Corporation
Nashua, New Hampshire

Printed April 2009

© 2001- 2009 PerfCap Corporation

All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from PerfCap required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

PerfCap shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for PerfCap products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

PerfCap software, including associated documentation, is the property of and contains confidential technology of PerfCap Corporation. Service customer is hereby licensed to use the software only for activities directly relating to the delivery of, and only during the term of, the applicable services delivered by PerfCap or its authorized service provider. Customer may not modify or reverse engineer, remove, or transfer the software or make the software or any resultant diagnosis or system management data available to other parties without PerfCap's or its authorized service provider's consent. Upon termination of the services, customer will, at PerfCap's or its service provider's option, destroy or return the software and associated documentation in its possession.

PAWZ AGENT Release Notes – V9.0

Release Overview

These release notes give an overview of new features and of the installation for PAWZ Agent Version 9.0. Please refer to the *PAWZ User and Installation Guide* for more in-depth information.

Summary of Features and Changes

The following is a summary of new features and changes specific to this release

PAWZ Agent had the following changes in v9.0.

- File System data by Percentage, GB and MB
- I/O wait time for LINUX (2.6)
- Kitting changes for Solaris, AIX, HP and Linux (rpm)
- Fixes for AIX data spikes
- AIX LPAR processing changes

Release Contents

This release of the PAWZ Agent product set consists of :

- PAWZ Agent
- PAWZ Real-time Agent
- PAWZ Agent Release Notes

Version 9.0 is comprised of the following items:

- PAWZ Agent 9.0 software
- PAWZ User and Installation Guide

For more information on all the components, see PAWZ User and Installation guide and the Software Product Description.

Feature: PAWZ Agent Changes 9.0A

PAWZ Agent had the following changes in V9.0A

- Processing of Top Filesystem by Percentage usage
- Processing of Top Filesystem by Gigabyte usage
- Processing of Top Filesystem by Terrabyte usage
- IO/WAIT cpu mode for Linux
- Packaging changes for AIX, Linux, Solaris, HP-UX
- Solaris Service changes.
- AIX data processing changes for fixing data spikes

End of Life for VMS Alpha , 7.2-1 and 7.1-2

eCAP v9.0 will be the last version for Alpha VMS 7.2-1 and 7.1-2.

Feature: PAWZ Agent EMC support (EMC Console package)

PAWZ Agent EMC Console Kit V9.0: no changes

Feature: PAWZ Planner Support

PAWZ Agent V9.0 Includes PAWZ Planner Support. (included since V7.0)

On UNIX and OpenVMS, PAWZ Planner support is incorporated with the eCAP

Analyze software. During installation, the “eCAP Analyze / Reduce / PAWZ Planner support” should be selected.

On Windows, installing the PAWZ Agent will automatically install the PAWZ Planner Support software.

The PAWZ Planner Support software enables the PAWZ Agent to provide the PAWZ Server with a data file, for workload characterization, that is tailored for Capacity Planning. This file, called a .MERG file is transferred to the PAWZ Server after a MERG request.

Feature: PAWZ AGENT for HP-UX Measureware

The PAWZ Agent can be configured for either eCAP Monitor or Measureware data source on HP-UX Itanium systems.

The PAWZ Agent will use Measureware’s extract utility to extract performance data into CSV format. These files are then converted into eCAP Monitor format (CPC). At this point data data can be used by all PAWZ applications.

The Measureware settings file /opt/perf/newconfig/parm should be configured to monitor all disks and all processes.

Some data elements are not available from Measureware data source.

Feature: PAWZ Real-Time Agent for Windows

PAWZ Real-Time Agent (since V7.1) has been changed to have only one component. The dependency on JAVA as been removed. The PAWZ Agent Windows packaging no longer requires any java processes. This also removes the need for port 2102. The PAWZ realtime Agent only requires port 2101.

Feature: PAWZ Agent Application Data Transfer Support

PAWZ Agent (since V7.3) has been changed to support transfer of application data files to a remote server system.

These file transfer requests originate from PerfCap’s File Transfer Application (FTA) product.

The PAWZ Agent only allows data to be transferred from paths that have been permitted to be accessed. The are defined on installation. The path list is stored in the pawz_hostname.config file or in the system registry on Windows platforms.

If the application data transfer request tells the pawz agent that the data file has an embedded time format. The following list of time formats are matched against the filename.

Time Formats:

YYYYMmmDD YYYYmmmDD YYYYMMmDD YYYYMMDD	DDMmmYYYY DDmmmYYYY DDMMMYYYY DDMMYYYY	YYYY-mmm-DD YYYY-MMM-DD YYYY-Mmm-DD YYYY-MM-DD
DD-mmm-YYYY DD-Mmm-YYYY DD-MMM-YYYY DD-MM-YYYY	DD_mmm_YYYY DD_Mmm_YYYY DD_MMM_YYYY DD_MM_YYYY	YYYY_mmm_DD YYYY_MMM_DD YYYY_Mmm_DD YYYY_MM_DD

PAWZ AGENT for OpenVMS

PAWZ Agent for OpenVMS – Installation

Installation of PAWZ Agent on OpenVMS Systems does NOT require a system reboot. The PAWZ Agent is installed with the VMSINSTAL utility.

The PAWZ Agent on OpenVMS now has a directory hierarchy to match the UNIX platform.

Logical	Description	Default
PAWZ\$LIBRARY	Root	sys\$sysdevice:[pawz]
PAWZ\$DB	Database	sys\$sysdevice:[pawz.db]
PAWZ\$DOC	Documentation	sys\$sysdevice:[pawz.doc]
PAWZ\$LOG	Log Files	sys\$sysdevice:[pawz.log]
PAWZ\$TMP	Temporary Files	sys\$sysdevice:[pawz.tmp]
PAWZ\$SETTINGS	Settings Files	sys\$sysdevice:[pawz.settings]

PAWZ Agent for OpenVMS – Post Installation

When installing in a VMS cluster environment, the post installation will require update of DCLTABLES on each node. After the DCLTABLES are updated, the \$PAWZ CONFIGURE should be run on each cluster node. Alternatively, the \$PAWZ CONFIGURE can be run on one system and the resulting .CONFIG file copied for each node in the cluster, see the sample below. There needs to be a PAWZ\$SETTINGS: PAWZ_hostname.CONFIG file for each node in the cluster.

PAWZ Agent for OpenVMS – Settings file

The settings file is a file that contains user startup preferences for the pawzagent.

The settings file created with the PAWZ CONFIGURE command is located in the PAWZ\$SETTINGS directory. This file will be named PAWZ_hostname.CONFIG.

After this file is created and contains the user preferences, using the \$PAWZ AGENT/START command will first read the settings file and then parse out any command line options, which will override the settings file.

PAWZ Agent for OpenVMS – PAWZ CONFIGURE

PAWZ CONFIGURE is the command used to create a pawz agent settings file on OpenVMS.

The following is sample output from running PAWZ CONFIGURE.

\$ PAWZ CONFIGURE

PAWZ Agent configuration:

Welcome to the pawzagent configuration program.

For help press ? at any command prompt

Enter PAWZ Agent Port (nn, ?) [1661] : <CR>

Enter PAWZ Server IP Address (n.n.n.n, ?) [0.0.0.0] : <CR>

Verbose logging (y,n,?) [n]? : N

Enter AppData Path (n,?) []? : **data:[appdata.dat]**

Enter Compression Command (n,?) [n]? : **zip -j -q *.zip ***

Selected settings are as follows:

PAWZ Server:	0.0.0.0
Setting Server:	0.0.0.0
PAWZ Port:	1661
Verbose Logging	No
AppData Path:	data:[appdata.dat]
Compression Command:	zip -j -q *.zip *

Keep these settings? (y/n) [y]:

Your settings are stored in the following file:

SYSS\$SYSDEVICE:[PAWZ.SETTINGS]pawz_SYS001.config

To modify these settings, execute:

```
$ PAWZ CONFIGURE
  PAWZ Agent initialization completed.
$
$! if clustered, create a .config for each node in the cluster
```

```
$ COPY SYS$SYSDEVICE:[PAWZ.SETTINGS]pawz_SYS001.config -
  SYS$SYSDEVICE:[PAWZ.SETTINGS]pawz_SYS002.config
```

PAWZ Agent for OpenVMS – logging

The PAWZ agent creates and uses 1 log file per day. The log files are located in the PAWZ\$LOG directory. When Starting the PAWZagent, the /VERBOSE qualifier will turn on detailed logging.

Log files are retained for 5 days before being deleted.

PAWZ Agent for OpenVMS – PERF file

The PERF file is a performance information data file which is transferred between the PAWZ Agent and the PAWZ Server. The agent will create this file when requested to by the PAWZ server, on nightly or demand download requests. This .PERF file is created on VMS using the PAWZ\$PERF.EXE image. Creating the .PERF file generally is transparent to the end user; but at times it may be necessary to manually run the command to create it for troubleshooting purposes. The syntax is as follows

```
$ PAWZ PERF /PMD_NODE=nodename/BEGIN=10-APR-2004:00:00/END=10-
APR-2004:23:59 /PERF=nodename.perf
```

There is an optional qualifier /VERBOSE

There is an optional qualifier /VERSION

The PAWZ\$PERF.EXE image will NOT create a .PERF from data collectors other than eCAP Performance monitor V4.0 and newer.

PAWZ Agent for OpenVMS – system logicals

PAWZ Agent for OpenVMS has some system logicals to control its behavior. These can be added to the PAWZ Agent startup file, `SYS$STARTUP:PAWZ$STARTUP.COM`

PAWZ\$AGENT_PRIORITY (default 4)
PAWZ\$AGENT_PGFLQUO (default 300000)

These should be defined with `/SYSTEM` and `/EXECUTIVE`.

PAWZ Agent for OpenVMS – command line options

The PAWZ Agent is controlled on OpenVMS system with the PAWZ command and the AGENT component. The PAWZ AGENT command has the following command line qualifiers.

Qualifier	Description	Default
<code>/START</code>	Start the PAWZ Agent	
<code>/STOP</code>	Stop the PAWZ Agent	
<code>/STATUS</code>	Get The PAWZ Agent status	
<code>/PORT=[n]</code>	Set PAWZ Agent port	1661
<code>/GROWTH_INTERVAL=[n]</code>	Set PMD file size check interval rate in seconds.	900
<code>/VERSION</code>	Get PAWZ Agent version	
<code>/WRITE_CONFIG</code>	Write PAWZ Agent settings file to PAWZ\$SETTINGS folder.	

PAWZ Real-Time Agent for OpenVMS – command line options

The PAWZ Real time Agent is controlled on OpenVMS system with the PAWZ command and the REALTIME component. The PAWZ REALTIME command has the following command line qualifiers.

Qualifier	Description	Default
-----------	-------------	---------

/START	Start the PAWZ Real-Time Agent	
/STOP	Stop the PAWZ Real-Time Agent	
/STATUS	Get The PAWZ Real-Time Agent status	
/SCAN=[n]	Set Scan Rate	2
/[NO]PROCESS	Turn On/Off collecton of per-process data	/NOPROCESS
/PORT=[n]	Set PAWZ Real-Time Agent port	2101
/PRIORITY=[n]	Set Real-Time Agent Priority. The range is from 0 to 23	18
/VERSION	Get PAWZ Real-Time Agent version	
/WRITE_CONFIG	Write PAWZ Real-Time Agent settings file to PAWZ\$SETTINGS folder.	

PAWZ Real-Time Agent for OpenVMS – settings file

The settings file is a file that contains user startup preferences for the PAWZ Real time Agent. This file, REALTIME_hostname.CONFIG is located in the settings directory of the PAWZ Agent install folder, PAWZ\$SETTINGS. It contains user preferences for starting pawz\$rta. After this file is created and contains the user preferences, issuing the \$PAWZ REALTIME command will first read the preferences file and then parse out any command line options, which will override the settings file.

At this time, the installation does not create the settings file and there is not a program to set the values. The settings file can be created by issuing:

```
$ PAWZ REALTIME /WRITE_CONFIG
```

This will create a PAWZ\$SETTINGS:RTA_`hostname`.config file. This is an ASCII file which can be edited.

PAWZ Agent for VMS PERF file option PAWZ\$PERF_INCLUDE_DSA

The PAWZ\$PERF processing will filter out DSA shadow set drives; only the physical drives are reported. In some cases, a user may wish to see the DSA master devices. To enable the PAWZ\$PERF to send the DSA information to the PAWZ Server, add the following line to the PAWZ\$STARTUP.COM file.

```
$ DEFINE/SYS/EXEC PAWZ$PERF_INCLUDE_DSA 1
```

PAWZ Agent for UNIX Platforms

PAWZ Agent for UNIX platforms– installation

The default installation directory for UNIX platforms depends on the platform. See table below.

The directory hierarchy contains all files for the pawzagent as well as eCAP Monitor, Analyze, Reduce and Planner. The logs subdirectory will contain all log files from the pawzagent and eCAP Monitor. The data sub-directory contains the data files (cpc) from the performance monitor. This directory can be changed by creating a soft link to a directory where the files are to be installed.

Each UNIX platform is installed with the install.sh script. This script will uninstall currently installed PerfCap software and then invoke the systems native installation facility with the user's selections.

PAWZ Agent for UNIX platforms– default installation path

PAWZ Agent software for UNIX platforms now has a default installation path that is more consistent with standards of each operating system. The new defaults are:

OS	Default
HP TRU64	/usr/opt/perfcap
HP-UX	/opt/perfcap
IBM AIX	/usr/perfcap
Linux	/usr/perfcap
Sun Solaris	/opt/perfcap

NOTE

At this time, on UNIX platforms, if the default installation path is changed a soft link must exist from the default location pointing to the new installation directory.

PAWZ Agent for UNIX platforms– Silent Installation

The silent_install.sh script that was introduced in V7.0 has been deprecated. Silent installs now are done with the install.sh using command line qualifiers.

- silent
- help
- data_dir <path>
- install_dir <path>
- license_dir <path>
- save_db (retain dba files)

- peak <ms value>
- poll <ms value>
- dump <dump value in seconds>
- retain <ndays>
- mapdisk <map disknames on Solaris>

- pawz_port <port>
- pawz_verbose
- pawz_server <ipaddress of pawzserver>
- pawz_appdata_dir <path>
- pawz_compression <compresson command>

- rta_port <rtaport>
- rta_scan <scan rate in seconds>
- rta_verbose
- rta_process

Samples

```
sh install.sh -silent
```

(this will install with default options or reinstall with currently set options)

```
sh install.sh -silent -license_dir /tmp -data_dir /data/ECP -poll 2000 -dump 120 \  
-rta_process
```

(this will install looking for the license files in /tmp and setting the directory for data

files in /data/ECP. The collector will scan at 2000 ms and write to disk every 120 seconds. The real time agent will collect process information)

PAWZ Agent for UNIX platforms– post installation

It is recommended that you use the system's initd mechanism for starting and stopping the eCAP monitor on system reboots

This may be done manually by copying the *install_path/settings/perfcap.initd* file to:

AIX

```
# cp /usr/opt/perfcap/settings/perfcap.initd /etc/rc.d/perfcap
#chown root:system /etc/rc.d/perfcap

#chmod +x /etc/rc.d/perfcap

# ln -s /etc/rc.d/perfcap /etc/rc.d/rc2.d/S90perfcap
# ln -s /etc/rc.d/perfcap /etc/rc.d/rc3.d/K90perfcap
# ln -s /etc/rc.d/perfcap /etc/rc.d/rc4.d/K90perfcap
# ln -s /etc/rc.d/perfcap /etc/rc.d/rc5.d/K90perfcap
# ln -s /etc/rc.d/perfcap /etc/rc.d/rc6.d/K90perfcap
# ln -s /etc/rc.d/perfcap /etc/rc.d/rc7.d/K90perfcap
# ln -s /etc/rc.d/perfcap /etc/rc.d/rc8.d/K90perfcap
# ln -s /etc/rc.d/perfcap /etc/rc.d/rc8.d/K90perfcap
```

HP-UX

```
# cp /opt/perfcap/settings/perfcap.initd /sbin/init.d/perfcap
# chown bin:bin /sbin/init.d/perfcap
# chmod 555 /sbin/init.d/perfcap
# ln -s /sbin/init.d/perfcap /sbin/rc3.d/S90perfcap
# ln -s /sbin/init.d/perfcap /sbin/rc0.d/K90perfcap
```

Linux (RedHat)

```
# cp /usr/local/perfcap/settings/perfcap.initd /etc/rc.d/init.d/perfcap
# chkconfig --add perfcap
# chkconfig --list perfcap
```

Linux (SLES Suse)

```
# cp /usr/local/perfcap/settings/perfcap.initd /etc/rc.d/perfcap
# chkconfig --add perfcap
# chkconfig --list perfcap
```

Solaris

svc enable perfcap

Tru64

```
# cp /usr/opt/perfcap/settings/perfcap.initd /sbin/init.d/perfcap
# ln -s /sbin/init.d/perfcap /sbin/rc3.d/S90perfcap
# ln -s /sbin/init.d/perfcap /sbin/rc0.d/K90perfcap
```

If you have modified the /etc/inittab file, please remove entries which start the cpcunix process.

Please review the install_path/settings/perfcap.initd file to ensure that startup command line is correct for your site.

PAWZ Agent for UNIX platforms– settings file

The settings file is a file that contains user startup preferences for the pawzagent. This file, pawz_hostname.config is located in the settings directory of the perfcap install folder /usr/opt/perfcap/settings. It contains user preferences for starting the pawzagent. After this file is created and contains the user preferences, issuing the pawzagent command will first read the preferences file and then parse out any command line options, which will override the settings file.

There are settings for each command line option; they are customizable by running the pawz_configure program. This file is run during installation and is also available post installation by running /usr/opt/perfcap/bin/pawz_configure.

PAWZ Agent for UNIX platforms– pawz configure

The following is sample output from running pawz_configure. Detailed information is available by entering a '?' at each prompt.

#/usr/opt/perfcap/bin/pawz_configure

PAWZ Agent configuration:

Welcome to the pawzagent configuration program.

For help press ? at any command prompt

Enter PAWZ Agent Port (nn, ?) [1661] : <CR>
Enter PAWZ Server IP Address (n.n.n.n, ?) [0.0.0.0] : **192.168.0.100** <CR>
Verbose logging (y,n,?) [y]? : **n**
Enter AppData Path (n,?) []? : **/data/appdata_dir**
Enter Compression Command (n,?) [n]? : **gzip -c * > *.gz**

Selected settings are as follows:

PAWZ Server:	192.168.0.100
Setting Server:	0.0.0.0
PAWZ Port:	1661
Verbose Logging	No
AppData Path:	/data/appdata_dir
Compression Command:	gzip -c * > *.gz

Keep these settings? (y/n) [y]: **Y**

Your settings are stored in the following file:

/usr/opt/perfcap/settings/pawz_sys01.config

To modify these settings, execute image:

/usr/opt/perfcap/bin/pawz_configure

PAWZ Agent initialization completed.

#

PAWZ Agent for UNIX platforms – command line options

The PAWZ Agent is activated with the *pawzagent* command on UNIX based systems. The command line qualifiers are as follows:

Qualifier	Description	Default
-start	Start the PAWZ Agent	(default)
-stop	Stop the PAWZ Agent	
-status	Get The PAWZ Agent status	
-port <nn>	Set PAWZ Agent port	1661
-priority <nn>	Set priority level from -20 (highest) to 20 (lowest).	0
-datadir <path>	Directory to find raw data files	<i>Install_path/data</i>
-logdir <path>	Directory for log files	<i>Install_path/logs</i>
-appdatadir <path>	Directory for application data files	""
-compression <cmd>	Compression command for compressing application data.	""
-server <server IP>	PAWZ Server IP	0.0.0.0
-[no]verbose	Verbose logging	-noverbose
-version	Get PAWZ Agent version	
-write_config	Write PAWZ Agent settings file to <i>install_dir/settings</i>	

PAWZ Agent for UNIX platforms – logging

The PAWZ agent now creates and uses 1 log file per day. The log files are located in the logs directory under the installation directory. When starting the pawzagent, the -verbose qualifier will turn on detailed logging.

PAWZ Agent for UNIX platforms – PERF file

The PERF file is a performance information data file which is transferred between the PAWZ Agent and the PAWZ Server. The agent will create this file when requested to by the PAWZ server, on nightly or demand download requests. This .PERF file is created on UNIX using the pawzperf image. Creating the .PERF file generally is transparent to the end user; but at times it may be necessary to manually run the command to create it for troubleshooting purposes. The syntax is

as follows:

```
# on TRU64
```

```
# /usr/opt/perfcap/bin/pawzperf -cpcunix  
/usr/opt/perfcap/data/ecp_hostname_2004Apr14.cpc-1 \  
-begin 14-APR-2004:00:00 -end 14-APR-2004:23:59 -perf ./nodename.perf
```

Using the pawzperf command with no command line qualifiers will process the previous 24 hours data with the existing .memory file (in settings folder). It will create a .PERF file (in the tmp folder) named hostname_yyyyMMMdd.perf

```
# /usr/opt/perfcap/bin/pawzperf
```

There is an optional qualifier `-verbose`

PAWZ Real-Time Agent for UNIX platforms – command line options

The PAWZ Real time Agent is activated with the `pawzrta` command on UNIX based systems. The command line qualifiers are as follows:

Qualifier	Description	Default
<code>-start</code>	Start the PAWZ Real-Time	(default)
<code>-stop</code>	Stop the PAWZ Real-Time	
<code>-status</code>	Get The PAWZ RealTime Status	
<code>-[no]collectproc</code>	Turn on/off collection of per-process data	<code>-nocollectproc</code>
<code>-[no]alert</code>	Turn on/off real-time alerting	<code>-alert</code>
<code>-scan <nn></code>	Set scan rate in seconds	2
<code>-port <nn></code>	Set TCPIP port	2101
<code>-priority <nn></code>	Set the process priority from -20 (highest) to 20 (lowest)	0
<code>-logdir <path></code>	Log file directory	<i>Install_path/logs</i>
<code>-[no]verbose</code>	Enable/disable verbose logging	<code>-noverbose</code>
<code>-write_config</code>	Write config file to the settings folder	

PAWZ Real-Time Agent for UNIX platforms – settings file

The settings file is a file that contains user startup preferences for the PAWZ Real-Time Agent. The file `rta_hostname.config` is located in the settings directory of the perfcap installation folder. (e.g. `/usr/local/perfcap/settings`). It contains user preferences for starting `pawzrta`. After this file is created, it contains the user's preferences. Issuing the `pawzrta` command will first read the preferences file and then parse out any command line options, which will override values from the settings file.

At this time, the installation does not create the settings file and there is not a program to set the values. The settings file can be created by issuing the "`pawzrta - write_config`" command. This will create `rta_`hostname`.config` file in the settings folder. This is an ASCII file which can be edited.

PAWZ Agent for Windows Platforms

PAWZ Real-Time Agent for Windows – installation

The PAWZ Agent is a system service which will run continuously on WINDOWS systems.

PAWZ Real-Time Agent for Windows – post installation

After completion of the PAWZ Agent installation, the PAWZ Agent and PAWZ Real time Agent services should have been started.

PAWZ Real-Time Agent for Windows – java

Java is no longer required. Port 2102 is no longer used.

PAWZ Real-Time Agent for Windows – logging

The PAWZ agent now creates and uses 1 log file per day. The log files are located in the log folder under the installation directory (C:\Program Files\PerfCap\PAWZ\Agent\log).

Verbose logging for the Agent can be enabled by editing the system registry entry HKLM\SOFTWARE\PerfCap\PAWZagent\Log Details. The value should be “1” (for detailed logging on) or “0” (for detailed logging off).

PAWZ Agent for Windows – registry entries

The following table contains each PAWZ Agent registry entry, default value and a description.

The root for PAWZ Agent is:

HKEY_LOCAL_MACHINE \ SOFTWARE \ PERFCAP \ PAWZagent

<u>Name</u>	<u>Default</u>
Port	1661

Port contains the PAWZ Agent to PAWZ Server TCP socket port.

InstallDirectory **C:\program files\PerfCap\PAWZ\Agent**

InstallDirectory contains the PAWZ Agent root directory path.

TmpDirectory **C:\program files\PerfCap\PAWZ\Agent\Tmp**

TmpDirectory contains the path where the PAWZ Agent/RealTime Agent temporary files will be created.

BinDirectory **C:\program files\PerfCap\PAWZ\Agent\Bin**

BinDirectory contains the path where the PAWZ Agent/RealTime Agent executable and binary files will be installed.

LogDirectory **C:\program files\PerfCap\PAWZ\Agent\Log**

LogDirectory contains the path where the PAWZ Agent/RealTime Agent log files will be written.

SettingsDirectory **C:\program files\PerfCap\PAWZ\Agent\Settings**

SettingsDirectory contains the path where the PAWZ Agent/RealTime Agent settings files will be written.

OutgoingDirectory **C:\program files\PerfCap\PAWZ\Agent\Outgoing**

OutgoingDirectory is not used.

DeviceDirectory **C:\program files\PerfCap\PAWZ\Agent\Device**

DeviceDirectory contains the path where the PAWZ Agent/RealTime Agent device files will be installed. These files are for PAWZ Planner support.

LoopDelay **“5”**

LoopDelay contains a value between “0” and “20”. This sets a “n” millisecond delay in each CPU consuming loop of a PAWZPERF execution. This allows throttling of the PAWZPERF.exe image. This will spread the CPU consumed by the image over a longer time period.

AppDataDirectory “”

AppDataDirectory contains a comma separated list of paths that the pawzagent is allowed to look into for application data.

CompressionCommand “”

CompressionCommand contains simple expression value that the PAWZ Agent to construct a command for compressing application data file(s) before transfer. The CompressionCommand should contain a valid command with asterisks in place of the data file name. For Example: `zip -q -j *.zip *`

LowPriority “0”

LowPriority contains a value of “1” or “0” . If set to “1” the PAWZ Agent will run data processing requests at BELOW_NORMAL priority.

LogRetention “5”

LogRetention contains a value for how long the PAWZ Agent should retain log files.

KeepPerf 0

KeepPerf contains a value (“0” or “1”) for enabling/disabling keeping of .PERF files.

ServerIPAddress “0.0.0.0”

ServerIPAddress contains a comma separated list of IP Address of PAWZ Server(s) that the Agent will respond to.

Log Details “0”

LogDetails contains a value (“0” or “1”) for enabling/Disabling verbose logging for the PAWZ Agent.

Log Details RTA “0”

Log Details RTA contains a value (“0” or “1”) for enabling/Disabling verbose logging for the PAWZ Real-Time Agent.

RTA_Port “2101”

RTA_Port contains the port for PAWZ Real-Time Agent to PAWZ Real-Time Server communications.

RTA_Process **“0”**

RTA_Process contains a value (“0” or “1”) for dnabling/disabling Real-Time collection of per-process data: NOT USED

Version **“version string”**

Version is an output registry entry. The pawzagent will set the value to the version and build date string.

Restrictions and Known Problems

The following is a summary of currently known restrictions and potential problems.

[OpenVMS, reference 1692] PAWZ PERF does not send extended disk

The pawz perf processing on OpenVMS is not returning Disk utilization, Disk queue length, Disk data rate or disk response time for the top disk graphs.

[All, reference 1287] PAWZ PERF top images by memory does not add “others”

The pawz perf computation for Top Images by memory does not add in an others class. So the memory that is displayed in the PAWZ Server graphs does not add up to the total amount of memory on a system.

[Windows, reference 1162] PAWZ Real-Time Agent Hang.

PAWZ Real-Time agent on Windows has been seen to hang with the following error written to the log file

RTDC-F-RECV : 997 Overlapped I/O will complete later

The workaround is to restart the PAWZ real-time agent.

[Linux, reference 1104] PAWZ Agent realtime exception error on Linux

Defining real-time exception on Linux sometimes causes the PAWZ Realtime to exit.

[ALL] PAWZ Perf CPU offline

The PAWZ Agent through pawzperf may misreport CPU Utilization if Processors go on and off line. CPU will be charged to <noproc>

[ALL] PAWZ Perf Interval < 60

The PAWZ Agent through pawzperf is not able to process performance data files (CPC or PMD) that have a collection dump rate of less than 60 seconds. These will be reported as *Invalid Perf File*.

[ALL] PAWZ Perf Processing Data where DC dump rate changed

The PAWZ Agent through pawzperf is not able to process performance data files (CPC or PMD) after a data collector dump rate interval change. If the system changes from 300 to 120 seconds on the dump rate, the PAWZ perf will only process up to the point where the interval has changed.

[AIX] PAWZ Real-Time Memory Utilization reporting

The PAWZ Real-Time Agent on AIX does not collect memory utilization.

[TRU64] PAWZ Real-Time Agent hang

The PAWZ Real-Time Agent may hang if 100+ processes are created within 1 second of each other.

[UNIX] PAWZ Real-Time -collectproc

The PAWZ Real-Time Agent on qualifier -collectproc usage has changed. The new usage is -collectproc or -nocollectproc.

[VMS] PAWZ Real-Time Agent on VMS 8.2

The PAWZ Real-Time Agent does not work with OpenVMS Alpha 8.2.

[VMS] PAWZ Real-Time on VAX

The PAWZ Real-Time Agent is not supported on VAX VMS.

[VMS] PAWZ Real-Time Processes

The PAWZ Real-Time Agent on OpenVMS does not add new processes to the Real-Time applet tree. The list of processes can become stale over time.

[WINDOWS] PAWZ port conflict

The PAWZ Real-Time Agent on Windows has a conflict of socket ports with the *Microsoft Queue Server Service*. Each use port 2101.

PAWZ Agent Problem Resolution

Problems that have been fixed are listed in the next sections. The first information on each line within the brackets is the eCAP version the fix refers to and the reference number (bug tracking ticket number).

PAWZ Agent problems fixed: OpenVMS

[V9.0A, reference 3106] PAWZAgent VMS check PMD size change

A problem where the PAWZ Agent could shut down the eCAP Monitor when checking for PMD File growth, when the eCAP Monitor caches writes, has been fixed. The PAWZ Agent now uses a settable threshold for determining if the PMD file has not grown. The default is 15 minutes.

A command line qualifier has been added /GROWTH_INTERVAL=nn

[V9.0, reference 2990] PAWZ Realtime Agent Scan Rate

A problem where the PAWZ realtime agent would only collect data using the default 2 second scan rate has been fixed. Any user specified scan rate will work.

[V9.0, ALL, Reference 2963] PAWZ real time exceptions

A problem where the PAWZ Realtime agent was not processing the requests for real time exception data properly has been fixed.

[V8.4, reference 2722] PAWZ Agent Connection errors on IA64

A problem with PAWZ Agent socket connections has been fixed.

[V8.0, reference 2043] /PRIORITY option added

The PAWZ Real-Time Monitor now supports /PRIORITY to alter the base priority of the process. The PAWZ Real-Time Monitor now starts with base priority 18.

[V8.0, reference 2043] Real-Time Support for Alpha V8.n and IA64 V8.n

The PAWZ Real-Time Monitor now supports OpenVMS Alpha V8.n systems and OpenVMS Itanium V8.n systems.

[V7.1, reference 1388] favorite disk matching

A problem where disks may not be seen as a favorite has been fixed. This was due to placement of a space character in between the device name and the device label.

[V7.1, reference 1249] \$pawz perf /VERSION added

A `--version` qualifier has been added to the `pawzperf` command. This is an aid in troubleshooting.

[V7.1, reference 1189] PAWZ Agent tracks PMD data file growth.

The PAWZ Agent now tracks the growth rate of the PMD data file when it is checking the data collector status. If the PMD file is not growing, the PAWZ Agent will issue a data collector STOP command. The PAWZ Server will then issue the data collector START command. This change is due to a rare occurrence where the PERFCAP MONITOR continued to run but did not write data.

[V7.1, reference 1046] PAWZ Agent PERF file erroneous tag removed

An erroneous tag, ISSUEDTO, has been removed from the .PERF file. This sometimes caused the PAWZ Server to be unable to load the .PERF datafile.

[V7.1, reference 1041] PAWZ Agent PERF file logging

The PAWZ Agent now will always log the command used to generate a .PERF file. This aids in troubleshooting.

[V7.1, reference 1026] PAWZ PERF file error fixed

A problem where the `ecap-monitor.txt` license file was not properly added to the .PERF file (causing the PAWZ Server to be unable to load the .PERF data) has been fixed.

[V7.1, reference 1010] PAWZ Agent PGFLQUO changes

PAWZ Agent will now get its page file quota value from the SYSTEM account. If the value specified is greater than the default (300000) or greater than a defined `PAWZ$AGENT_PGFLQUO` value.

[V7.1, reference 1020] /VERBOSE changes

PAWZ Agent will now create separate .LOG file for each PAWZ\$PERF run when /VERBOSE is specified. These log files reside in PAWZ\$LOG and have the form PAWZ\$AGENT_PERF_HOSTNAME_DATE.LOG

[V7.0, reference 586] PAWZ Agent purge sysconfig

PAWZ Agent is now purging the PAWZ\$TMP: pawz_sysconf*.dat files. These are temporary files used by the PAWZ Agent for PAWZ Planner requests

[V7.0, reference 439] PAWZ\$PERF ACCVIO with large number of favorite Disks

A problem where if a large number of favorite disks (or processes) are defined, the PAWZ PERF command will crash and no performance data will download has been fixed.

[V7.0, reference 516] PAWZ Realtime ACCVIO

A problem where the PAWZ Real-time Agent was crashing on systems with large process count has been fixed.

[V7.0, reference 362] PAWZ Agent Install on Alpha VMS 8.2

The PAWZ Agent will now install on Alpha VMS 8.2.

[V7.0, reference 346] PAWZ Agent verbose logging

Verbose logging has been fixed for the PAWZ Agent.

[V6.0A, reference 100] PAWZ Realtime /SCAN added

PAWZ Real-Time Agent now has /SCAN=nn command line option.

[V6.0A] Per-Processor CPU reporting

A problem where CPU after the first processor was incorrect has been fixed.

PAWZ Agent problems fixed: UNIX

[V9.0A, Solaris, reference 3187] PAWZAgent reported negative memory size

A problem where the PAWZ Agent on Solaris 32 bit Intel was reporting a negative memory size has been fixed.

[V9.0A, Linux, reference 3146] PAWZAgent dmidecode error

A problem where the PAWZ Agent was printing an error to the terminal when image dmidecode was not found has been fixed.

[V9.0, reference 2990] PAWZ Realtime Agent Scan Rate

A problem where the PAWZ realtime agent would only collect data using the default 2 second scan rate has been fixed. Any user specified scan rate will work.

[V9.0, ALL, Reference 2963] PAWZ real time exceptions

A problem where the PAWZ Realtime agent was not processing the requests for real time exception data properly has been fixed.

[V9.0, AIX, reference 2230] AIX Process Page Fault data spike

A problem where Process Page Faults would be reported as a very large number has been fixed

[V9.0, AIX, reference 2836] AIX Process NIC data spike

A problem where NIC packet and transfer rates would be reported as a very large number has been fixed

[V9.0, AIX, reference 2836] AIX Process TCP data spike

A problem where TCP packet and transfer rates would be reported as a very large number has been fixed

[V9.0, AIX, reference 2884] AIX Per-Processor CPU

A problem where double the processors were being reported has been fixed.

[V9.0, AIX, eference 2971] AIX CPU Mode IO/WAIT

A problem where CPU IO/Wait time was incorrectly reported has been fixed.

[V8.4, AIX, Reference 2802] PAWZ Agent AIX serial number fix

A problem where the system serial number was being mis-reported for AIX systems has been corrected.

[V8.4, TRU64, Reference 2796] PAWZ PERF on TRU64 Process CPU

A problem where the process CPU was being incorrectly summed has resulting in large <NOPROC> has been fixed.

[V8.4, HP-UX, Reference 2708] PAWZ PERF on HP-UX Timezone

A problem where pawzperf on HP-UX didn't recognize time zone changes has been fixed

[V8.1, UNIX, Reference 2310] PAWZ Agent allows longer server IP list.

The PAWZ Agent settings file value for ServerIP has increased sized. Allowing for a longer list of valid Servers.

[V8.0, HP-UX, Reference 1887] PAWZ Agent on HP-UX packet origin fix.

A problem where on HP-UX systems the PAWZ Agent saw each packet as originating from 0.0.0.0 has been fixed.

[V7.2, Reference 1691] pawzperf with no qualifiers

The pawzperf command will now produce a .PERF file in the tmp folder of the installation for the previous 24 hours when no command line qualifiers are supplied.

[V7.2, Reference 1690] pawzperf data processing

pawzperf when processing data from multiple cpc data files, would sometimes incorrectly put the last interval for a data file as the last interval in the processing period. This would appear as a spike at midnight. This has been corrected.

[V7.1, reference 1249] pawzperf -version added

A -version qualifier has been added to the pawzperf command. This is an aid in troubleshooting.

[V7.1, reference 1052] PAWZ Realtime: Negative Memory Utilization on Solaris

An problem where memory utilization on Solaris 10 systems sometimes when negative has been fixed.

[V7.1, reference 1046] PAWZ Agent PERF file erroneous tag removed

An erroneous tag, ISSUEDTO has been removed from the .PERF file. This sometimes caused the PAWZ Server to be unable to load the .PERF datafile.

[V7.1, reference 1041] PAWZ Agent PERF file logging

The PAWZ Agent now will always log the command used to generate a .PERF file. This aids in troubleshooting.

[V7.1, reference 1023] PAWZ PERF file error fixed

A problem where the ecap-monitor.txt license file was not properly added to the .PERF file (causing the PAWZ Server to be unable to load the .PERF data) has been fixed.

[V7.0, UNIX, reference 863] No license reported

A problem with the V60AL pawzagent reporting data collector license not found has been fixed.

[V7.0, Linux, reference 674] Real-Time Agent missing process RSS

A problem where the PAWZ real-time agent was missing Process RSS has been fixed

[V7.0, UNIX, reference 647] Real Time Process RSS now KB

Process RSS values have been standardized across unix platforms to report in Kilobytes.

[V7.0, Solaris TRU64, AIX , reference 621] Real Time Memory Leak

Memory leaks have been fixed

[V7.0, reference 439] PAWZ\$PERF crash with large number of favorite Disks

A problem where having a large number of favorite disks (or processes) caused the PAWZ PERF command to crash and no performance data will download has been fixed.

[V7.0, reference 402] PAWZ Agent reports invalid agent

A problem where PAWZ Agent was report DC DOWN for an invalid agent has been fixed.

[V7.0, UNIX, reference 372] PAWZ Perf crash

A problem where the pawzperf would crash when processing data from a system where the data collection interval had changed has been fixed.

[V7.0, UNIX, reference 360] pawzrta –stop qualifier

The stop command has been enhanced to ensure that the pawzrta stops.

[V7.0, UNIX, reference 352] pawzagent –stop qualifier

The stop command has been enhanced to ensure that the pawzagent stops.

[V6.0A, AIX] <noproc> computation

A problem where excessive <noproc> was seen on AIX has been fixed.

[V6.0A, AIX] Gaps in Data

A problem where there were gaps in the data was being reported was fixed.

[V6.0a, AIX] Process Priority Reporting

Process priority reporting was fixed.

[V6.0, UNIX] PAWZ Real-Time log purge

PAWZ Real-Time Agent log files are now purged.

[V6.0, ALL] Real-Time not installed reported

The PAWZ Agent now reports Real-Time not installed rather than Real-time down.

[V6.0, VMS] PAWZ\$PERF ACCVIO

A PAWZ\$PERF crash processing data from systems with Fibre Channel controllers has been fixed.

[V6.0, Solaris] Undefined symbols

A problem where pawzagent and pawzrta images were reporting unresolved symbols has been fixed.

[V6.0, UNIX] pawzrta priority

The default priority of the pawzrta process has changed from 0 to -2

PAWZ Agent problems fixed: Windows

[V9.0, reference 2990] PAWZ Realtime Agent Scan Rate

A problem where the PAWZ realtime agent would only collect data using the default 2 second scan rate has been fixed. Any user specified scan rate will work.

[V8.1, Reference 2267] PAWZ Agent has application data request time formats

The PAWZ Agent now matches a broader range of time formats within the application data file. These formats are as follows

YYYYMmmDD YYYYmmmDD YYYYMMMD YYYYMMDD	DDMmmYYYY DDmmmYYYY DDMMMYYYY DDMMYYYY	YYYY-mmm-DD YYYY-MMM-DD YYYY-Mmm-DD YYYY-MM-DD
DD-mmm-YYYY DD-Mmm-YYYY DD-MMM-YYYY DD-MM-YYYY	DD_mmm_YYYY DD_Mmm_YYYY DD_MMM_YYYY DD_MM_YYYY	YYYY_mmm_DD YYYY_MMM_DD YYYY_Mmm_DD YYYY_MM_DD

[V8.0, reference] Verbose logging on OpenVMS change

Verbose logging on OpenVMS has been change to log the results of an individual PERF generation or MERG generation command. These results go into their own log file. Log files with `_P_` in their name are PERF request outputs. Log files with `_M_` within their name are MERG outputs.

[V8.0, Reference 2017] PAWZ Agent has locked .MEMORY File

A problem where the PAWZ Agent could leave the .MEMORY file in a locked state, thereby preventing additional data requests to fail has been fixed.

[V8.0, Reference 2012] PAWZperf scales CPU utilization

On slower systems, pentium iii, the pawzperf could improperly scale CPU utilization upwards resulting in CPU being reported as greater then 100%. This has been fixed.

[V8.0, reference] Log File name format change

The format of the log file name has changed to

pawzagent_HOSTNAME_yyyyMMMdd_hhmm.txt

[V7.2, Reference 1691] pawzperf.exe with no command line arguments

A problem where the pawzperf.exe required command line qualifiers to be specified has been fixed. The pawzperf.exe can be run with out command line qualifiers to process the previous days data. The resulting .PERF file will be written to the tmp directory under the installation folder.

[V7.2, Reference 1687] pawzrta crashes on windows 2003 server

A problem where the PAWZrta would crash, usually a few seconds after startup, on Windows 2003 servers has been fixed. This was a memory corruption that would unpredictably crash the image.

[V7.2, Reference 1682] PAWZ RTA detailed logging fixed.

A problem where it was not possible to enable detailed logging for the PAWZRTA has been fixed. This is now possible with the "Log Details RTA" registry entry. The "Log Details RTC" registry entry has been removed.

[V7.1, Reference 1475] pawz perf data file processing only first file

A problem where pawzperf was only processing the first .CPC file has been fixed. The pawzperf image will now process all CPC files.

[V7.1, Reference 1054, 1102] PAWZ Agent create process error

A problem where some systems would not create the pawzperf process due to a space in the path name has been fixed. This problem resulted in the inability to create a .PERF file. And could be seen by the following error in the log file:

PAWZAGENT-E-UPDATE, CreateProcess failed, %1 is not a valid Win32 application.

[V7.0, reference 851] CPU Utilization for Workloads

A problem where CPU was overcharged to workloads with many short lived processes has been fixed.

[V7.0, reference 794] PAWZ Agent Real-time Service

The PAWZ Real-time agent service controls a java process which processes real-time requests. The service now ensures that this java process is actually running. If the java process stops running the system service will restart it.

[V7.0, reference 730] PAWZ Agent circuit reset

The PAWZ Agent now reports *Unregistered Server* and not *Circuit Reset Error*.

[V7.0, reference 310] Unique Processes

pawzperf on windows now allows the option of not averaging processes; Allowing for the analysis of individual (unique) processes.