Dialogic® PowerMedia™ XMS

Diagnostics Guide

November 2016 05-2747-001

www.dialogic.com

Copyright and Legal Notice

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation and its affiliates or subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in certain safety-affecting situations. Please see http://www.dialogic.com/company/terms-of-use.aspx for more details.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at www.dialogic.com.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 6700 Cote-de-Liesse Road, Suite 100, Borough of Saint-Laurent, Montreal, Quebec, Canada H4T 2B5. **Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Dialogic, Dialogic Pro, Dialogic Blue, Veraz, Brooktrout, Diva, BorderNet, PowerMedia, PowerVille, PowerNova, MSaaS, ControlSwitch, I-Gate, Mobile Experience Matters, Network Fuel, Video is the New Voice, Making Innovation Thrive, Diastar, Cantata, TruFax, SwitchKit, Eiconcard, NMS Communications, SIPcontrol, Exnet, EXS, Vision, inCloud9, NaturalAccess and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation and its affiliates or subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 6700 Cote-de-Liesse Road, Suite 100, Borough of Saint-Laurent, Montreal, Quebec, Canada H4T 2B5. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Third party links were confirmed as live as of the publication date of this document.

Table of Contents

Revision History

Revision	Release Date	Notes
05-2747-001	November 2016	Initial release of this document.
Last modified: November 2016		

Refer to www.dialogic.com for product updates and for information about support policies, warranty information, and service offerings.

1. Welcome

This diagnostics guide provides high-level procedures for collecting data that is used by Dialogic Technical Support to aid in debugging the Dialogic® PowerMedia™ Extended Media Server (also referred to herein as "PowerMedia XMS" or "XMS").

The procedures in this document provide data that is commonly requested by Dialogic Technical Support. When contacting Dialogic Technical Support to resolve issues with PowerMedia XMS, it is recommended to provide as much diagnostic information as possible with each case or test run in order to minimize retesting or regathering in the future. If the collected logs are large, contact Dialogic Technical Support to use the Dialogic FTP site to transfer the files.

This guide is not meant to be an exhaustive list of all available debugging methods. When working with the Dialogic Technical Support, additional information may be needed to help understand, reproduce or replicate, and resolve an issue.

Intended Audience

This publication is intended for the following audience:

- System Integrators and Network Operation Engineers (NOEs)
- Toolkit Developers
- Independent Software Vendors (ISVs)
- Original Equipment Manufacturers (OEMs)

Note: This document discusses tools and utilities that may be available within the operating system or hypervisor environments. These utilities or settings may vary from vendor to vendor. Please contact the vendor for more information.

Related Information

See the following for additional information:

- PowerMedia XMS documentation at http://www.dialogic.com/manuals.

2. Overview

Data to debug the PowerMedia XMS can be retrieved from various areas and components of the XMS. The following diagram provides a high-level overview of the different functional areas of PowerMedia XMS and also of the adjunct third-party components such as the operating system or hypervisor technologies.

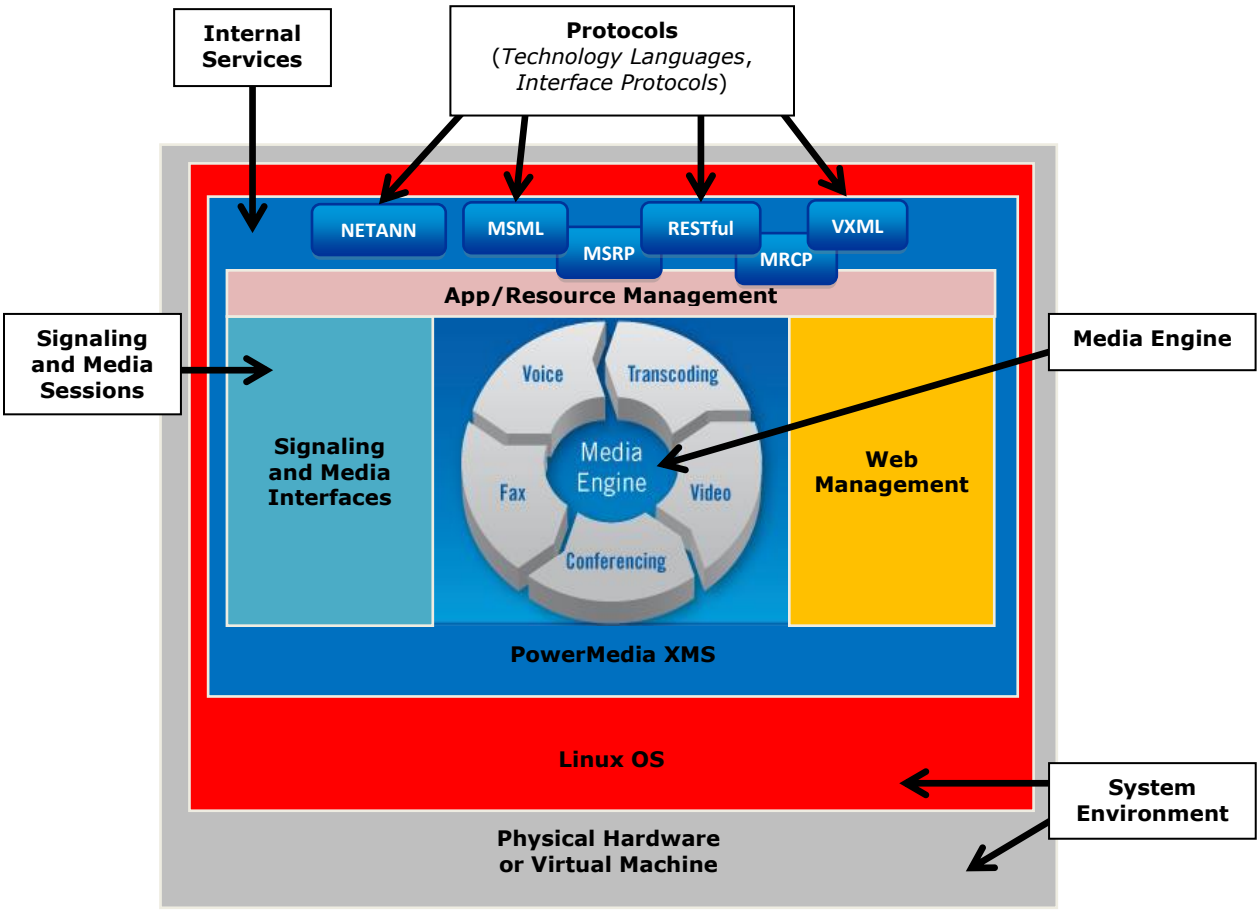


Figure 1: High-Level Architecture of XMS

The following sections in this document cover the areas and components shown in Figure 1 and provide a high-level overview of the area diagnostic capability that can aid in debugging the PowerMedia XMS:

- Signaling and Media Sessions
- Internal Services and Protocols
- Media Engine
- System Environment

3. Signaling and Media Sessions

Network Tracing and Logging

Example Use Case

Tools such as tcpdump or tshark can be used for network analysis.

Tcpdump is a common network protocol packet analyzer tool that allows the user to capture and display packet information from a live network. Tcpdump uses libpcap library to capture packet data on the network. Tcpdump is generally distributed on Linux systems.

Tshark is another network protocol packet analyzer tool that allows the user to capture packet data from a live network. It also provides the capability to read packets from a previously saved capture file and either prints a decoded form of those packets to the standard output or writes the packets to a file. Tshark's native capture file format is pcap format, which is also the format used by tcpdump and various other tools.

SIP call control messages, HTTP messages, audio/video quality issues, call rejections, disconnected calls, media sessions (RTP), and media quality (such as "no media" or "no DTMF") are issues that can be diagnosed using network tracing tools.

System Performance Impact

Log files may get large over time and consume disk space. Circular buffers and filtering are recommended for long-term tracing runs. Also, specify the correct interfaces if multiple NIC/Networks are enabled.

Considerations

Network security or network configuration may need to be addressed in order to collect this type of data. Consult with your network system administrator before enabling a network tracing tool. Under a heavy load, the Linux kernel may also drop packets destined for tcpdump or tshark.

References

For information on how to enable and collect logs for troubleshooting purposes, refer to the "Start or stop network capture (tcpdump)" section of the following article:

- http://www.dialogic.com/support/helpweb/helpweb.aspx/4209/how_to_enable_and_collect_logs_for_troubleshooting_purposes/PM_XMS

For more information on tcpdump, refer to the following information:

- http://www.tcpdump.org/tcpdump_man.html
- An example of the tcpdump command that would capture ten minutes of network traffic into six files is shown below:

```
tcpdump –i any –B 10240 –s 0 –G 600 –w 'networktrace_%Y-%m-%d_%H:%M:%S.pcap' –W 6
```

For more information on tshark, refer to the following article:

- http://www.wireshark.org/docs/man-pages/tshark.html
- An example of the tshark command that utilizes the RINGBUFFER option. This capture will have 100 files each limited to a maximum value of 20 MB:

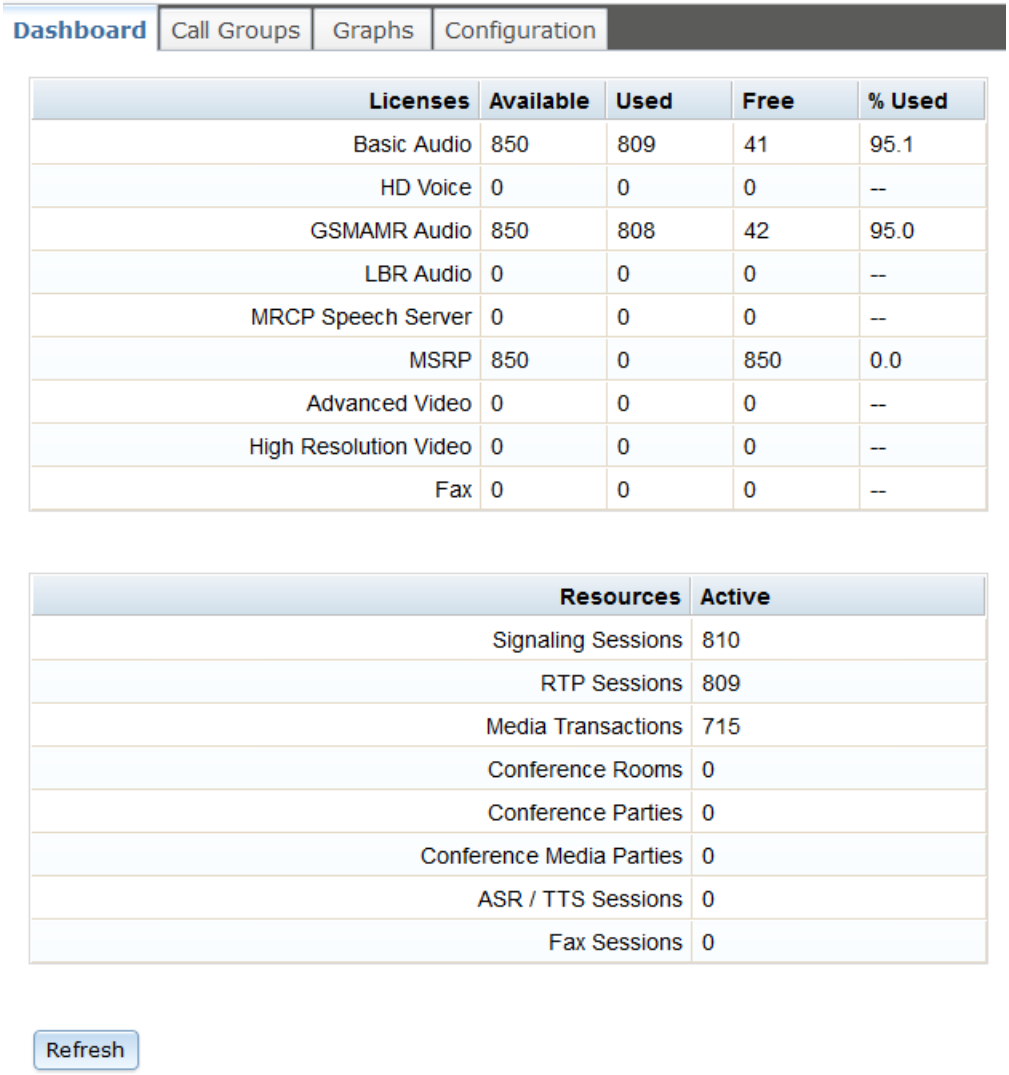
```
tshark -i eth0 -w networktrace_%Y-%m-%d_%H:%M:%S.pcap -s 0 -b filesize:20000 -b files:100
```

Call Monitoring

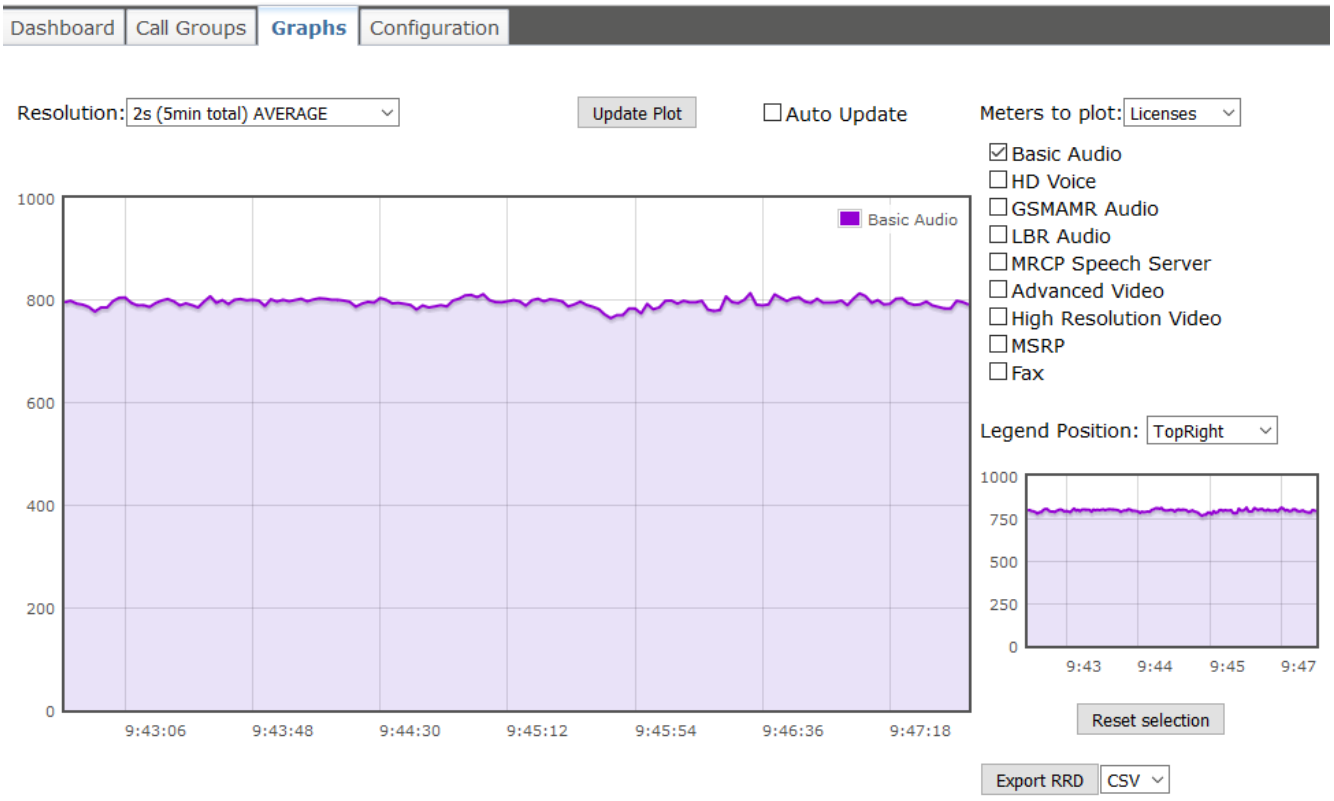
Example Use Case

In the PowerMedia XMS Admin Console, the **Monitor** menu provides information such as real-time active counts of resources and licenses being used by the PowerMedia XMS. Applications can use this data to monitor the system call, code, conferencing status, and usage.

The **Monitor > Dashboard** page in the PowerMedia XMS Admin Console contains real-time active counts of resources and licenses being used by the XMS. This data can be used to monitor system usage.



On the **Monitor > Graphs** page in the PowerMedia XMS Admin Console, graphs display meter values including license usage, resource usage, memory usage, CPU usage, network usage, SIP statistics, and HTTP statistics. The meter values are stored so they can be viewed within a particular time frame in seconds, minutes, hours, or days. The resolutions are 2 seconds, 60 seconds, 5 minutes, and 60 minutes. Those interval values are for AVERAGE or MAX time.



X Axis = Time
Y Axis = Count

System Performance Impact

None.

Considerations

None.

References

Refer to the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information on configuring and enabling the XMS Call Monitoring feature.

Call Detail Record (CDR)

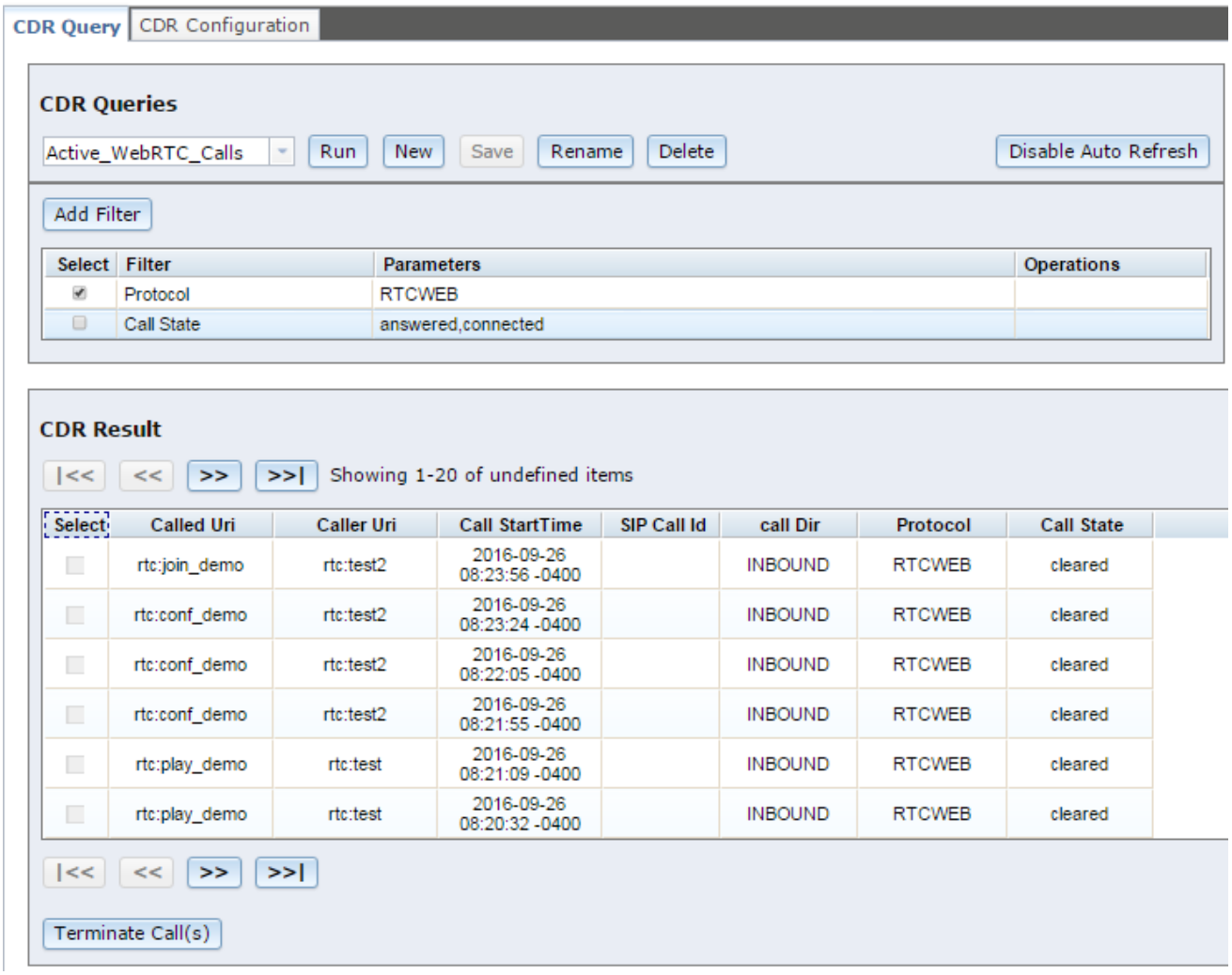
Example Use Case

In the PowerMedia XMS Admin Console, the **CDR** menu provides real-time, detailed information for all calls successfully arriving on the XMS. Detailed call information includes call start time, call duration, audio bit rate, DTMF mode, and RTP start and end time. System administrators can easily peer into an individual active call and non-active calls on the server to obtain more call information.

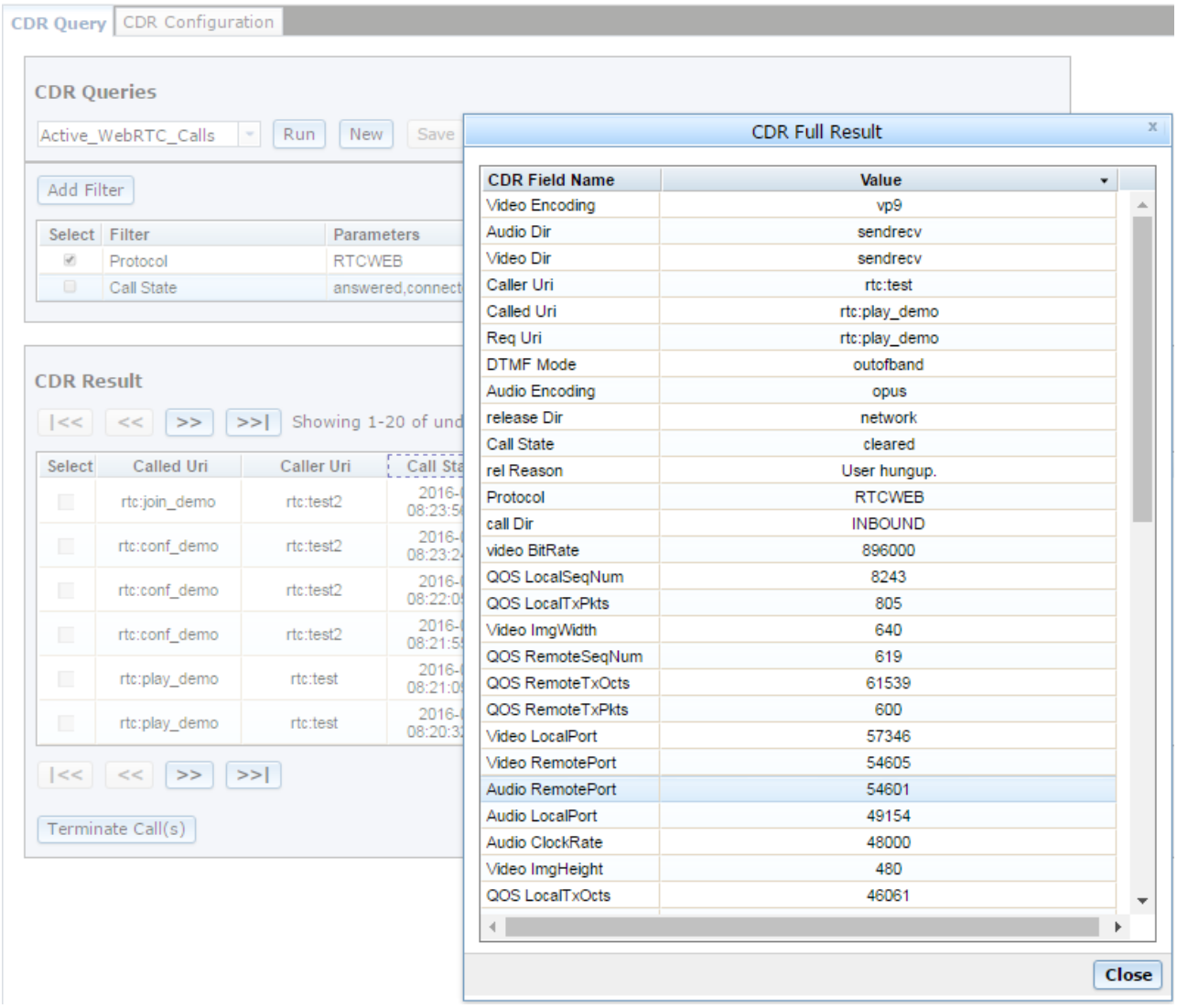
CDR services will need to be enabled via the PowerMedia XMS Admin Console on the **System > Services** page. Select **On Start Enabled** for **cdrserver** to enable CDR services.

Once CDR services are enabled, go to the **CDR > CDR Query** page in the PowerMedia XMS Admin Console.

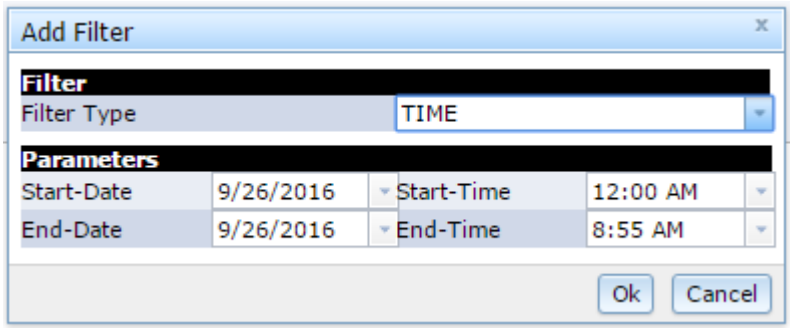
A quick view of CDR results when load is being applied to the system is shown.



Double-click on a listed call for more information about the call.



To view certain CDR results, add a filter.



System Performance Impact

None.

Considerations

None.

References

Refer to the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information on configuring and enabling the XMS CDR feature.

4. Internal Services and Protocols

Diagnostic Logging

Diagnostic logging provides debugging information for XMS internal services (e.g., xmserver, nodecontroller, appmanager, broker, cdserver, etc.) and protocols (MSML, NETANN, REST, VXML, mrcpclient, etc.). These logs are used to troubleshoot overall XMS functionality (e.g., call status, service status, etc.) and protocol-specific issues.

Example Use Case

In the PowerMedia XMS Admin Console (WebGUI), the **System > Diagnostics** page provides the option to set the logging level for PowerMedia XMS. There are five logging level values:

- **NOTICE**: Top logging level and provides references such as "System Started" type messages.
- **ERROR**: Includes **NOTICE** level prints and provides known error conditions (e.g., "Engine level API FAILURES"). This is the lowest logging level.
- **WARNING**: Includes **NOTICE+ERROR** prints and flags references that are not errors but could point to potential issues depending on their context.
- **INFO**: Includes **NOTICE+ERROR+WARNING** prints and provides informational level logging (e.g., new call notification prints).
- **DEBUG**: Includes **NOTICE+ERROR+WARNING+INFO** prints and provides lower level verbose prints that Dialogic Engineering uses to help trace a problem within the system. This is the highest logging level.

The default PowerMedia XMS log location is */var/log/xms*. The diagnostics can be downloaded to your system by clicking the **Download Diagnostics** button.

Note: Multiple log files are created. Each service/application has its own log file. By default, these log files are set to 10 MB each with 100 backups. Therefore, each service (excluding HMP service) should have a 1 GB scrolling window for logging. If necessary, change the default logging parameters within the WebGUI. Refer to the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for details on configuration.

System Performance Impact

When running XMS diagnostics logging at increased logging levels, there will be some additional CPU and hard drive access loading. The loading is dependent on the channel density of the system and also the level of logging that has been enabled. If any system performance issues are encountered, you may need to reduce the channel load or reduce some of the debugging prints. The **ERROR** setting provides the lightest logging level while the **DEBUG** setting provides the most intensive logging level.

Considerations

Consider CPU and MEM usage when the logging level is set to the highest level (**DEBUG**). At the highest level of logging, disk I/O tends to impact system performance. The XMS WebGUI does not need to be restarted when the logging levels are modified.

References

Refer to the following articles for information on how to enable and collect logs for troubleshooting purposes:

- http://www.dialogic.com/support/helpweb/helpweb.aspx/4764/useful_scripts_to_monitor_administer_and_debug_xms/PM_XMS
- http://www.dialogic.com/support/helpweb/helpweb.aspx/4209/how_to_enable_and_collect_logs_for_troubleshooting_purposes/PM_XMS

SNMP Management

Example Use Case

Simple Network Management Protocol (SNMP) is a standards-based IP network management mechanism for exchanging information between SNMP agents that typically reside on a managed device and SNMP management systems. SNMP allows the user to set thresholds for traps using different parameters such as license usage, service status changes, etc.

System Performance Impact

Minimal.

Considerations

None.

References

Refer to the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information on configuring and enabling XMS for SNMP.

5. Media Engine

Run Time Function (RTF) Tracing and Logging

RTF logging traces the Dialogic Engine library function calls (APIs) that are being issued or called by the XMS processes or services. By default, RTF logging levels are enabled to MOST "critical" errors ONLY. Depending on the functional area of where the issue is being seen, additional client component modules can be enabled to capture additional engine layer information. To enable additional information in the RTF logs, the RTF configuration XML file needs to be modified to enable tracing on selected client component modules.

The XMS logs are located here: */var/log/dialogic*.

Example Use Case

Logging is specific to media operations within the media engine, such as streaming, low-level device connections, etc.

System Performance Impact

When running RTF logging with increased logging levels, there are some additional CPU and hard drive access loading. The loading is dependent on the channel density of the system and also the level of logging that has been enabled. If any system performance issues are encountered, you may need to reduce the channel load or reduce some of the debugging levels to reduce the debug prints.

Considerations

It is recommended to monitor how long it takes for the RTF logs to roll over. The size and number of logs to keep during rotation can be adjusted if necessary. Enabling other component modules should be done with the recommendation of Dialogic Support.

References

- "PowerMedia XMS Troubleshooting: RemoteRtfTool" in *Dialogic® PowerMedia™ XMS Installation and Configuration Guide*
- "How to enable and collect logs for troubleshooting purposes" in http://www.dialogic.com/support/helpweb/helpweb.aspx/4209/how_to_enable_and_collect_logs_for_troubleshooting_purposes/PM_XMS

6. System Environment

System Message Logs

Example Use Case

Logging at the system environment level is used to focus on operating systems and system processes type errors. The */var/log/message* file stores valuable non-debug, non-critical, and critical messages. This log is the general system activity log for the server. Additionally, some of the XMS server (AppManger) and media engine logging are stored within this system log file.

The default PowerMedia XMS log location is */var/log*.

When using Red Hat or CentOS, it may be necessary to turn off the rate/burst limiting. Disable rate/burst limiting in the messages file by doing the following.

Red Hat and CentOS 6.x

1. Add the following lines to the end of */etc/rsyslog.conf*:

```
$SystemLogRateLimitInterval 0
$SystemLogRateLimitBurst 0
```

2. Run the following:

```
/etc/init.d/rsyslog restart
```

Red Hat and CentOS 7.x

Note: RH7.x uses "journald" to set the rate limit.

1. Adjust the behavior of the systemd-journald service by modifying */etc/systemd/journald.conf* (or *journald.conf.d*).

   Note: To turn off any kind of rate limiting, set either value to 0.

2. Restart journald services for the changes to take effect.

```
# systemctl restart systemd-journald.service
```

System Performance Impact

Minimal.

Considerations

None.

References

Refer to the operating system vendor for additional information on server level messaging.

For more information on configuring RH/CentOS 7.x rate-limiting via journald, refer to the following:

- http://www.freedesktop.org/software/systemd/man/journald.conf.html
- http://www.suse.com/documentation/sled-12/book_sle_admin/data/journalctl_config.html

Runtime System Performance Tools

Example Use Case

Linux utilities such as *top*, *sar*, and *netstat* are used to focus on system-level performance troubleshooting. The *top* processes on a system shows trends in areas like CPU utilization and memory usage. The *sar* processes on a system monitors performance of various sub-systems in real-time. Raw CPU percentage is used to rank the processes.

System Performance Impact

Minimal.

Considerations

It is useful to provide a view of the processor by each processor's utilization instead of just the whole system average.

References

For more information on Unix top, refer to the following article:

- http://www.unixtop.org/man.shtml

For more information on sar, refer to the following articles:

- http://linux.die.net/man/1/sar
- http://www.thegeekstuff.com/2011/03/sar-examples

For more information on monitoring network status via netstat, refer to the following article:

- http://docs.oracle.com/cd/E23824_01/html/821-1453/ipconfig-142.html
- http://linux.die.net/man/8/netstat

For more information on using the Linux command line tools to monitor performance, refer to the following:

- http://www.tecmint.com/command-line-tools-to-monitor-linux-performance - An example of using the Linux top command to take a snapshot of server performance and add it to a log file every ten seconds is shown below:

```
top -b -n 10 >> top.txt
sar -n DEV 1 1 >> sar.txt
netstat -anope
```

Services and Processes Core Dumps

Example Use Case

PowerMedia XMS runs applications and services in order to execute the media server functionality. A core dump is a generated file that contains diagnostic information about a service or process and can be used for debugging purposes. Core dumps can be created on-demand or are generated by the operating system when a service or process ends abnormally.

The abrt daemon is used to collect information about the cores. The default location is in */var/spool/abrt*.

Note: The entire abrt directory should be collected and provided to Dialogic support team, NOT just the core file.

System Performance Impact

The system will be down until recovery of XMS has completed.

Considerations

Coring is enabled and set to ON for all XMS services and processes by default.

References

For more information on core dumps, refer to the following article:

- http://wiki.archlinux.org/index.php/Core_dump

Hypervisor Monitoring

Example Use Case

When using hypervisors (e.g., VMware), the guest operating system or virtual machines use virtual hardware that emulates service components. The hypervisor manages the physical resources for each virtual operating system image. Each hypervisor has its own set of tools to monitor CPU performance, memory usage, etc. The vendor should be consulted to determine what tools are available.

Monitor the hypervisor to determine any potential bottlenecks or oversubscription of system resources. Bottlenecks and oversubscription can have an impact on virtual operating system images that are running, which impacts XMS performance by introducing latencies.

System Performance Impact

The hypervisor vendor documentation should be consulted to determine performance impact of using tools that the vendor supplies. It is recommended to monitor the system for a short time after enabling these tools to assess the impact.

Considerations

Look for any potential bottlenecks of CPU usage, memory usage, and I/O system contention. Check the hypervisor vendor documentation to assess what performance tools the vendor has at a hypervisor level. The vendor may allow configurable alarms that can be monitored.

References

For information on latency-sensitive software in VMware as well as optimizing the VM host, refer to the following articles:

- http://www.dialogic.com/webhelp/XMS/3.2/XMS_VMOptimizingAppNote.pdf
- http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/latency-sensitive-perf-vsphere55-white-paper.pdf

For information on virtual machine CPU usage alarm for VMWare (2057830), refer to the following article:

- http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2057830

For information on virtual machine memory usage alarm for VMware (2057846), refer to the following article:

- http://kb.vmware.com/selfservice/microsites/search.do?cmd=displayKC&docType=kc&externalId=2057846&sliceId=1&docTypeID=DT_KB_1_1&dialogID=702228918&stateId=0%200%20702244328

System and Software Installation Specific

Example Use Case

There are various log files generated during the operating system installation and PowerMedia XMS installation. These log files can be useful when troubleshooting issues with starting PowerMedia XMS.

Refer to the following log files for information:

- */root/install.log* - Operating system installation log
- */var/log/yum.log* - YUM (Yellowdog Updater, Modifier) information logging
- *xms_install.log* - XMS installation log (located in the */tmp* directory by default of an ISO installation or can be located in the installation directory of an RPM installation)
- System specification information (see References)
- Package management information (see References)
- xmsinfo script (see References)

The default locations of the system installation logs are */root* and */var/log* directory.

System Performance Impact

None.

Considerations

None.

References

For information on system specification, refer to the following article:

- http://www.binarytides.com/linux-commands-hardware-info

For information on package management with RPM and on how to use RPM commands, refer to the following articles:

- http://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/3/html/System_Administration_Guide/s1-rpm-using.html
- http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap3sec20.html
- The following sample xmsinfo script collects different types of system information such as the active running processes, netstat information, memory information, installed packages, and sar, top, and XMS WebGUI information. Copy the script below and save it to a file on the system. Change the file permission to be an executable file (chmod a+x).

```
#!/bin/bash

OUTFILE=/home/powermedia/xmsinfo.tgz

if [ $# -eq 1 ]; then
         OUTFILE=$1
fi

ls -lR /usr/dialogic > /var/log/xms/dirlisting.out



touch /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "ps -fe" > /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
ps -fe >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "rpm -qa" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
rpm -qa >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "netstat -anope" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
netstat -anope >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "df -h" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
df -h >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "free -ml" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
free -ml >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "uptime" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
uptime >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "top -b -n 1" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
top -b -n 1 >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "sar -A" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
sar -A >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "ifconfig" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
ifconfig >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "iptables --list" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
iptables --list >> /var/log/xms/additionalinfo.out
```

```
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "hostname" >> /var/log/xms/additionalinfo.out


echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
hostname >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "ping hostname" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
ping `hostname` -c 1 >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "env" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
env >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "proc/meminfo" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
cat /proc/meminfo >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "proc/cpuinfo" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
cat /proc/cpuinfo >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "/etc/system-release" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
cat /etc/system-release >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "lspci" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
lspci >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "sysctl -A" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
sysctl -A >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
echo "dmidecode | egrep -i 'manufacturer|product|vendor'" >> /var/log/xms/additionalinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/additionalinfo.out
dmidecode | egrep -i 'manufacturer|product|vendor' >> /var/log/xms/additionalinfo.out

touch /var/log/xms/webuiinfo.out
echo "################################################################################" >> /var/log/xms/webuiinfo.out
echo "WebUI" >> /var/log/xms/webuiinfo.out
echo "################################################################################" >> /var/log/xms/webuiinfo.out
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/system >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/system/network >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/system/network/eth0 >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/license >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
```

```
curl http://127.0.0.1:10080/codecs >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/services >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/sip >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/routing >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/rtp >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/msml >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out
curl http://127.0.0.1:10080/mrcpclient >> /var/log/xms/webuiinfo.out 2> /dev/null
echo "--------------------------------------------------------------------------------" >> /var/log/xms/webuiinfo.out

tar cvzf $OUTFILE --exclude='*.tgz' /var/log/xms /var/log/dialogic /var/log/messages /etc/profile
```