Oreka TR 1.10 Administrator Manual

Revision 3771

Table of Contents

Chapter 1. Introduction

Table of Contents

What is Oreka

Oreka is a cross-platform system for recording and retrieval of audio streams. It supports VoIP, TDM and sound device based capture. It also includes features such as quality monitoring and screen recording. The Oreka user interface (OrkWeb) is web-based and provides features such as call live monitoring, recordings playback, extensive search and query capabilities, audit trail and many others.

Architecture

The Oreka system consists of a combination of the following services

- **OrkAudio** : this is the audio capture background service. It supports VoIP, TDM and Sound Device based recording.
- **OrkTrack** : this service centrally tracks activity on the entire system and logs recordings to any popular SQL database.

- **OrkWeb** : this service is the web interface accessible via any standard compliant web browser. It relies on the Tomcat web server.
- **OrkRfb** : this is the screen capture background service. It relies on the RFB protocol used in VNC.

The system supports multiple instances of OrkAudio and OrkRfb reporting to OrkTrack so that multiple recording servers can be seen as one recording system. OrkTrack and OrkWeb are installed as one package. They may reside on the same server as the recorder or on a different server.

For the sake of simplicity, Oreka TR will be refered to simply as Oreka in the rest of this document.

Chapter 2. Pre-requisites

Table of Contents

Server Specs

The most important parameter for performance is CPU L2 cache. Recording higher number of concurrent calls is facilitated by using big CPU L2 caches such as 8MB, 12 MB or even more.

Running Oreka in **Virtual Machines (VM)** is supported

Test Server (PC or Laptop)

- Pentium IV, 1 GHz CPU
- 512 MB RAM
- 500 MB Hard Drive

Production Server

- Dual processor/Dual Core CPU, 2.6 GHz CPU (0-100 concurrent conversations, Dual Core CPU; 100-200 concurrent conversations, Quad Core CPU)
- 2 GB RAM
- 4 MB L2 Cache
- 2 fast Hard Drives (SCSI 10K RPM), one for the OS, one for storing the recordings

Operating System

Oreka TR runs on Linux and Windows platforms.

OrecX's preferred platform is **Linux CentOS 6 64-bit** . However, Oreka can be deployed on other Linux flavors. All recent Windows versions are also supported. This includes Windows XP, 2003 server, Win7, 2008 server

For support of other operating systems, please inquire at <support@orecx.com> .

Database

MySQL is recommended as Orecx LLC's primary database environment. Oreka also supports most major database systems including IBM DB2, Oracle and PostgreSQL.

Getting VoIP traffic to the Oreka Server

Before Oreka can start recording, ensure that VoIP traffic is seen on a server interface. Use SPAN port mirroring to get the right traffic to the Oreka server. Two configurations are possible:

- SPAN monitoring the entire VoIP VLAN so that all traffic to and from phones is intercepted;
- SPAN monitoring the PSTN Gateways **and** the Signalling server (e.g. SIP proxy, Cisco Call Manager, Avaya Communication Manager, ...)

This is to ensure that both the media traffic (RTP) and signalling (SIP, Skinny, H.323, UNISTIM, ...) are intercepted by the recorder. Use a packet analyzer such as the free Wireshark tool to verify that both types of packets are appearing on the Oreka server's interface.

Once the VoIP traffic appears on the server, you are ready to start using the Oreka software.

Mechanisms to get VoIP traffic

In terms of insertion point, Oreka can intercept packets via several mechanisms:

- **Ethernet switch SPAN monitoring port:** for Ethernet switches that have this capability. For Cisco, see http://www.cisco.com/en/US/products/hw/switches/ps708/products_tech_note09186a008015c612.shtml . This is the most popular solution.
- **Ethernet tap:** requires additional hardware, this option has become fairly cheap lately. See Netoptics Teeny Tap or Barracuda Ethernet Tap products for example.
- Being installed **directly on the PBX** or media gateway server. Not ideal.
- Old style **Ethernet hub** inserted in the Ethernet path, i.e. all traffic copied to all ports. Not recommended, do this only for testing or low traffic sites.
- Setup the machine as an **Ethernet bridge** inserted in the Ethernet path. Not recommended, do this only for testing.

About Audio Files

Oreka uses the compressed GSM format to store audio recording files.

This format uses about 1.6 KBytes for 1 sec of recording (or 13 kbits/sec).

As an example, 2,000 hours of audio would require approximately 10 GBs of disk space.

Chapter 3. OrkAudio

Table of Contents

Overview

OrkAudio is the Oreka audio recorder component. It is a process that runs on Windows or Linux and records audio packets received on one of the server interfaces. It can record VoIP packets as well as TDM-based voice calls.

Typical Installation (Single Server)

On Linux

Here are the steps to install OrkAudio using installers on CentOS or Red Hat Enterprise Linux (RHEL).

Requirements

- You need a minimal installation of CentOS or RHEL. A graphical desktop is not required.
- You need to be logged in as "root" to install the Oreka software.

Installation

- Use the installer file provided to you by OrecX, e.g. `orkaudio-1.2-660-x1459-i386.centos5-installer.sh.tar`
- Untar it: `tar -xvf orkaudio-1.2-660-x1459-i386.centos5-installer.sh.tar`
- Run the installer: `./orkaudio-1.2-660-x1459-i386.centos5-installer.sh` (accept all required components)
- If the installer fails, please contact <support@orecx.com> . It is also possible to attempt the procedure in the section called “How do I manually install orkaudio?”

On Windows

Requirements

- You need to be logged in as Administrator before proceeding.
- Access to the internet is highly recommended for download of the Oreka software.

Installation

Use the installer file provided to you by OrecX, e.g. `orkaudio-1.2-657-x1463-win32-installer.zip` . Copy this file to a temporary folder on the target machine, unzip it and run the embedded executable. This will install WinPcap as well as OrkAudio.

Multiple Server Configuration (OrkAudio)

When installing Oreka on multiple servers, one or more OrkAudio recording servers are reporting to a single OrkWeb/OrkTrack central server. In this case, additional configuration in both OrkWeb and OrkAudio is required.

Communication with OrkTrack

The recorder needs to communicate to OrkTrack to report the recording metadata to be stored in the database. Thus, it needs to know where OrkTrack is running.

Make sure the `<TrackerHostname>` entry in the OrkAudio `config.xml` is properly set to the OrkWeb/OrkTrack hostname or IP address.

OrkWeb access to media files

For OrkWeb to be able to access the media files stored on the recorder's server, a web server application such as Apache httpd or Apache Tomcat needs to be installed and configured on the recorder's server. For a quick solution, use the OrkWeb installer and install only the Tomcat and Java Run-Time components. E.g., run ./orkweb-1.7-2586-x64-linux-installer.sh --nomysql --nooreka on Linux. In Windows, you can stop the installer after Java and Tomcat are installed.

OrkWeb also needs some special configuration, please refer to the section called “Multiple Server configuration (OrkWeb)”

For more details, contact <support@orecx.com> .

Applying OrkAudio License File

For OrkAudio to run properly, a license file must be applied. This file is provided to you by OrecX (e.g. `orkaudio-30-days-trial-license-20090320.txt`). Store this file in the folder where OrkAudio was installed, typically `/etc/orkaudio` on Linux and `C:\Program Files\OrkAudio` on Windows. Make sure to rename it to `license.txt` .

Warning: under Windows, you need to make sure file extensions are shown (go to My Computer/Explore/Tools/Folder Options/View and uncheck "Hide extensions for known file types"). Otherwise you risk naming the file licence.txt.txt without realizing it.

Whenever a new license file is applied, the orkaudio service must be restarted for the change to take effect.

Upgrade

Upgrade with RPM installer (Linux)

In this procedure, please replace version numbers with the relevant ones.

- Take a copy of /etc/orkaudio/config.xml and /etc/orkaudio/logging.properties if you had customized them
- # service orkaudio stop
- # rpm -e orkaudio-addons
- # rpm -e orkaudio
- # rpm -e orkbasecxx
- # tar xvf orkaudio-1.2-660-x1459-i386.centos5-installer.sh.tar
- # ./orkaudio-1.2-660-x1459-i386.centos5-installer.sh
- If any custom changes had been made to the old config.xml or logging.properties files such as IP filtering or NIC selection, apply those changes to the new files (do not simply overwrite the new files with the old ones, some configuration settings might have changed between the two versions)
- # service orkaudio start

Upgrade with individual RPM files (Linux)

In this procedure, please replace version numbers with the relevant ones.

- Take a copy of /etc/orkaudio/config.xml and /etc/orkaudio/logging.properties if you had customized them
- # service orkaudio stop
- # rpm -e orkaudio-addons
- # rpm -e orkaudio
- # rpm -e orkbasecxx
- # rpm -i orkbasecxx-1.2-691.i386.centos5.rpm
- # rpm -i orkaudio-1.2-691.i386.centos5.rpm
- # rpm -i **--nodeps** orkaudio-addons-1.2-1526.i386.centos5.rpm
- If any custom changes had been made to the old config.xml or logging.properties files such as IP filtering or NIC selection, apply those changes to the new files (do not simply overwrite the new files with the old ones, some configuration settings might have changed between the two versions)
- # service orkaudio start

Upgrade with Windows installer

- Take a backup copy of the orkaudio directory under Program Files
- Stop the orkaudio service
- Uninstall previous version OrkAudio
- Run the latest orkaudio installer (e.g. orkaudio-1.2-657-x1463-win32-installer.exe)
- If any custom changes had been made to the old config.xml or logging.properties files such as IP filtering or NIC selection, apply those changes to the new files (do not simply overwrite the new files with the old ones, some configuration settings might have changed between the two versions)

Upgrade with Windows binary archive

- Stop the OrkAudio service
- Unpack the orkaudio zip archive downloaded from your OrecX space (e.g orkaudio-1.2-688-x1537.zip)
- Select and copy all files inside the top directory. This should include orkaudio.exe, some .dll files as well as subdirectories
- Paste the files on top of your orkaudio install directory so that all .exe and .dll files are all replaced by the new ones in the archive
- When done, all replaced files are usually seen as selected in the Windows file explorer This can give you a good indication of the success of the operation
- Restart the OrkAudio service

Files Location

Audio Output Files

Audio output files are written to the `c:\oreka\audio` under Windows and in `/var/log/orkaudio` under Linux by default.

Audio files are classified according to the following default scheme (see also TapeFileNaming and TapePathNaming in the section called “File and Path Names in OrkAudio”):

yyyy/MM/dd/hh

Audio file themselves are named after the following scheme:

yyyyMMdd_hhmmss_trackingID.extension

You can modify the audio files location by editing the *<AudioOutputPath>* configuration parameter described in the section called “Configuration” . Note that if this parameter is changed, OrkWeb needs to be told where to look for the recordings. This requires modifying Tomcat's `$tomcat/conf/server.xml` file to update the context path docBase parameter accordingly:

```
<Context path="/audio" docBase="c:/oreka/audio/" ></Context>
```

If this parameter does not exist already, just add it under the <Host> section. Don't forget to restart Tomcat after this change.

Configuration Files

OrkAudio configuration files are located in the install directory under Windows and in `/etc/oreka` under Linux. The files are:

- *config.xml* : this is the main OrkAudio configuration file. Plugins also read their configration parameters from subsections of this file. Please see the section called “Configuration” for more details.
- *logging.properties* : this is the log4j logging configuration file which allows for great flexibility in logging scope and output format. Please see http://logging.apache.org/log4j/1.2/manual.html

Log Files

Log files are located in the install directory under Windows and in `/var/log/oreka` under Linux. By default, Oreka produces the following output:

- *orkaudio.log* : this is the main OrkAudio logfile.
- *tapelist.log* : this logfile contains the details (metadata) for each recording that was performed by OrkAudio.
- *messages.log* : : this log file contains a subset of details (metadata) for each recording that was performed by OrkAudio. Useful for re-creating database entries.

Plugins Files

Plugins exist as dll files under Windows and as DSO (Dynamic Shared Objects) with .so extensions under Linux. They are located in `{OrkAudioInstallDirectory}/audiocaptureplugins` under Windows and in `/usr/lib` under Linux.

- *VoIp.dll - libvoip.so* : VoIP recording plugin for SIP, Cisco Skinny and pure RTP protocols.
- *H323voip.dll* : VoIP recording plugin for H.323, Avaya, Nortel UNISTIM and MGCP protocols.
- *SoundDevice.dll - libsounddevice.so* : Sound Card based recording
- *Generator.dll - libgenerator.so* : Audio generator for faking audio capture (useful when testing)

Audio Encodings and Formats

Wire audio encodings

Wire audio encodings are detected automatically by OrkAudio. Audio is not usually stored in its original wire format. Audio is recorded in real time to mcf files in order to maximize capturing performance and is later transcoded to the final storage format as specified in the section called “Storage audio formats” . The following encodings are supported:

- G.711 ulaw
- G.711 alaw
- GSM 6.10
- iLBC
- G.729A
- G.723.1
- G.722

Storage audio formats

The storage format is the file format used to archive recordings to disk. Mcf capture files are transcoded from the wire encoding to the final storage format on a best effort basis. All possible storage encodings are currently wrapped into the wav container format. This means that all generated audio files have a .wav file extension and easily play on any existing Windows or Linux media player. The following formats are supported, please see the section called “Configuration” for more details.

- *GSM 6.10 wav* : default format
- *G.711 ulaw wav*
- *G.711 alaw wav*

- *PCM wav* : uncompressed audio, not recommended.

Configuration

Basic Configuration

Configuration of OrkAudio and its plugins is performed by modifying the config.xml file (see also the section called “Configuration Files”). Core OrkAudio configuring parameters are the following:

- *<AudioOutputPath>* : this parameter controls the root directory where capture and storage audio files are stored. It can be a relative or absolute path.
- *<CapturePlugin>* : this parameter controls which audio capture plugin should be used. Valid values are `VoIP.dll` and `libh323voip.dll` in Windows, and `libvoip.so` and `libh323voip.so` under Linux.
- *<TrackerHostname>* : the hostname or IP address of the server where OrkTrack (a component installed with OrkWeb) resides. If you use the OrkTrack hostname instead of the IP address (recommended), make sure that DNS is set up correctly and that you can ping that hostname from this OrkAudio server. Exemple:

  ```
  <TrackerHostname>my-orktrack-server1</TrackerHostname>
  ```

  You can also enter several orktrack engines, e.g. for redundancy. They must be comma separated. When using non-standard TCP ports for OrkWeb/OrkTrack (standard is port 8080), it is possible to specify them in the hostname:port format. For example, if one tracker is on port 8080 and the second tracker is on port 80:

  ```
  <TrackerHostname>my-orktrack-server1:8080, my-orktrack-server2:80</TrackerHostname>
  ```

- *<StorageAudioFormat>* : this parameter controls the final file format of the tapes. Valid values are the following: gsm, ulaw, alaw and pcmwav. "gsm" is the default value and is the best compression rate available. All values generate wav files with various degrees of compression.
- *<TapeProcessors>* : usually set to BatchProcessing, Reporting. Reporting ensures that the metadata about recordings is "reported" to the database through OrkTrack.
- *<DeleteNativeFile>* : this parameter allows you to keep the uncompressed .mcf file even after the transcoding to the .wav is complete. The default is "yes". Set it to "no" to keep the .mcf file.
- *<TapeDurationMinimumSec>* : minimum duration in seconds for a call to be recorded.
- *<AllowAutomaticRecording>* : if set to "yes" (default setting), calls will be recorded by default. "no" indicates that only explicitly requested recording will occur (e.g. from Live Monitoring in OrkWeb).
- *<LookBackRecording>* : if set to "yes" (default), will always record the call from the beginning regardless of when the request to start the recording was initiated, while a "no" setting will record only from the time the request was made.

File and Path Names in OrkAudio

It is possible to configure the path to which audio files are written as well as the audio file names through dynamic parameters

First, the *TapeFileNaming* tape processor needs to be added to the list of processors in the top node of the orkaudio config.xml file:

```
<TapeProcessors>BatchProcessing, TapeFileNaming, Reporting</TapeProcessors>
```

Secondly, the *<TapeFileNaming>* and/or *<TapePathNaming>* must also be added in the top node of the orkaudio config.xml file. They both contain a CSV list of elements. If an element appears between square brackets, it will be replaced by the value corresponding to the keyword. If an element appears without square brackets, it will be used verbatim in the file name. Example:

```
<TapeFileNaming>myrecordings-,[nativecallid]</TapeFileNaming>
```

with a native call ID of FDBCE@69.13.45.6 will result in the following file name: `myrecordings-FDBCE@69.13.45.6.wav` .

If the TapeFileNaming parameter is missing, the default naming scheme applies which is a timestamp plus the internal trackingID.

If the TapePathNaming parameter is missing, recordings are distributed to the default year/month/day/hour folder tree structure described in the section

called “Files Location” .

Note: TapePathNaming configuration is always relative to AudioOutputPath.

Here is a list of acceptable keywords for tape and path naming:

- [nativecallid]: this is the call ID extracted from the underlying protocol (SIP, Cisco Skinny, ...)
- [trackingid]: this is the internal Oreka tracking ID
- [direction]: in, out or unkn
- [shortdirection]: I, O or U
- [remoteparty]
- [localentrypoint]
- [localparty]
- [localip]
- [remoteip]
- [hostname]
- [year]
- [day]
- [month]
- [hour] in 24 hours format
- [min]
- [sec]

You can also use any tag or additional custom extracted field in file names by using its tag/field name as a key. e.g for a SIP field called X-Unique-ID, add [X-Unique-ID] to TapeFileNaming or TapePathNaming. See also the section called “Extracting arbitrary fields from SIP headers”

Additional example:

```
<TapeFileNaming>myrecording,[hour],[min],[sec],_,[shortdirection],_,[remoteparty],_,[localparty],_,[hostname]</TapeFileNaming>
<TapePathNaming>mypathprefix/,[year],[month],/,[day]</TapePathNaming>
```

Configuring the VoIP plugin

VoIP plugin specific configuration is found in the `config.xml` file under the *VoIpPlugin* tag. Many options are available for this plugin, such as limiting traffic, blocking traffic from/to a specific IP address, ... The default `config.xml` has some of the main options listed in it and commented out. If any of these parameters are not documented here, please contact <support@orecx.com> for more details.

Selecting Network Device(s) to intercept traffic from

It is possible to configure the network device to monitor for VoIP traffic using the <Devices> directive. While OrkAudio attempts to automatically select the server interface on which it detects VoIP traffic, you may need to configure the interface manually if orkaudio.log shows no sign of traffic. E.g.

In Windows:

```
<Devices>\Device\NPF_{E0E496FA-DABF-47C1-97C2-DD914DFD3354}</Devices>
```

In Linux:

```
<Devices>eth2</Devices>
```

Several comma-separated interfaces may be configured in the examples above

IP addresses based filtering

Filtering based on IP addresses or CIDR style subnets is available via the PcapFilter parameter, e.g.

```
<PcapFilter>host 192.168.0.34 or net 192.168.1.0/24</PcapFilter>
```

In the example above, only packets coming from or going to either IP address 192.168.0.34 or any address within the 192.168.1.x range will be retained. Any other packet will be ignored. The syntax is the standard tcpdump syntax as describe here: http://wiki.wireshark.org/CaptureFilters

Extracting arbitrary fields from SIP headers

It is possible to configure the VoIP plugin to extract any given standard or custom added SIP field by adding the following configuration parameter and specifying the wanted fields as a csv list (field names simply need to appear exactly as they appear in the SIP headers as viewed e.g via wireshark):

```
<SipExtractFields>contact, max-forwards, X-UNIQUE-ID</SipExtractFields>
```

After the change, the extracted fields will start appearing in OrkWeb as tags in the detailed recording view (when clicking on a recording's timestamp) and will become searchable via the tag name/tag text search boxes. They will also appear as tags in the tape messages that can be found in orkaudio.log

Live Monitoring

For Live Monitoring from OrkWeb to work, the 59120 port must be accessible on the server where OrkAudio is running. In Linux check the iptables settings. in Windows, the firewall settings.

See also the section called “Live Monitoring (On-Demand Recording)” for configuring OrkWeb.

Running OrkAudio

Make sure the OrkAudio license file is properly applied. Refer to the section called “Applying OrkAudio License File” .

Starting the OrkAudio service

Under Windows, start the OrkAudio service in Service Management (start/run/services.msc).

Under Linux, start the OrkAudio service by typing *service orkaudio start* on the command line.

Verifying that OrkAudio started

You might need to double-check that the orkaudio service was started correctly.

In Linux, use the following command:

```
# ps -ef | grep orkaudio
```

A line showing that orkaudio is running must appear:

```
root     32071     1  0 10:02 ?        00:00:00
/usr/sbin/orkaudio
```

In Windows' Service Management tool (start/run/services.msc), ensure that the orkaudio service is started

OrkAudio Auto-Start

During installation using the official installers, OrkAudio will install itself as an automatic service, i.e. a service that restarts automatically if the server is rebooted. This is important in the case of power failure, maintenance or other unpredictable events that may cause the system to fail or be restarted. Ensure that the service is configured properly as follows:

- In Windows' Service Management tool (start/run/services.msc), right click on the orkaudio service and ensure that the Startup type is set to Automatic
- In Linux, the command "chkconfig -- list orkaudio" should yield a response such as below:

```
# orkaudio        0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

If not, `chkconfig orkaudio on` will need to be executed.

Migrating OrkAudio to Another Server

To move OrkAudio to a different server with the same operating system follow the procedure below:

- Stop the orkaudio service on the old server
- Install OrkAudio (refer to the section called “Typical Installation (Single Server)”) on the new server
- Copy the OrkAudio configuration files and license file from the old server to the new server: `config.xml` , `logging.properties` and `license.txt` , in the OrkAudio installation folder
- Copy all the audio files and folders to the new server
- Start the OrkAudio service on the new server

Load balancing - orkbalancer

The Oreka load balancer (orkbalancer) is a software service that can take a high amount of VoIP traffic and share it across multiple core recorders (orkaudio). Traffic input can come from one or more network interfaces and be distributed to any number of recorders. Recorders are specified by their IP addresses and ports, meaning that:

- It is possible to run recorders on the same host as the balancer or on separate servers.
- It is possible to run multiple recorders on a given server, thereby reaping the benefits of using servers with lots of cores.

The load balancer ensures that the RTP media traffic is well balanced between the recording targets and uses a round robin algorithm. The two directions of a given RTP exchange are always sent to the same recorder so that bidirectional conversations can be recorded. Any non-RTP traffic is sent to all recorders, meaning that signalling traffic is automatically distributed to all recorders.

orkbalancer currently only runs on Linux CentOS targets.

The configuration file for orkbalancer is the following file: /etc/orkbalancer/config.xml

In order to control from what network interfaces the orkbalancer should listen from (port mirroring NICs):

```
<HostDevicesList>eth0, eth1</HostDevicesList>
```

In order to control between which orkaudio targets the traffic should be balanced:

```
<LoadBalancingTargets>127.0.0.1:20000,192.168.0.10:20000,192.168.0.11:20002</LoadBalancingTargets>
```

The LoadBalancingTargets ports should be spaced by two units, e.g. if one target is on port 20000, the next target should be at least on port 20002. This is because signalling and media traffic are sent to two consecutive ports.

Any participating orkaudio target should be configured correctly for load balancing by adding the following in the orkaudio config.xml under the VoIpPlugin node:

```
<OrekaEncapsulationMode>true</OrekaEncapsulationMode>
<OrekaEncapsulationPort>20000</OrekaEncapsulationPort>
```

Troubleshooting

OrkAudio does not record any VoIP traffic

If no recordings appear in the <AudioOutputPath> directory, Here is the checklist:

- Windows users only: Make sure winpcap3.1 or above is properly installed (start/programs should contain a Winpcap entry)
- Make sure that the VoIP plugin records from the right network interface(s) specified in the <Devices> xml tag of the config.xml file. A list of all network interfaces is shown in orkaudio.log at startup.
- Make sure that RTP traffic is actually present on the specified network interface(s). A packet sniffer such as Ethereal can be useful for that.
- Refer to the section called “Getting VoIP traffic to the Oreka Server”

Not possible to replay recorded files

Recoded wav files should all be replayable by a media player such as Windows Media Player. Here is the checklist

- Make sure that a sound device is installed on your system.
- Make sure the file is not being processed by OrkAudio. Try again later.
- Make sure that if the RTP traffic is encoded as G.729A or G.723.1, you are properly licensed for those codecs. OrkAudio logs error messages at startup if there is a licensing problem.

Only one side of the conversation is recorded

Make sure that RTP traffic for both sides is actually seen on the considered network interface. A packet sniffer such as Ethereal can be used for that.

Metadata is not detected

Use wireshark to see if signalling packets (e.g. SIP or Skinny) can be seen on your chosen sniffing NIC. Simply capture some traffic for 1 minute and type "sip" or "skinny" as a wireshark display filter in order to see only the relevant packets. If resultset is empty, it might mean that the packet interception strategy is missing some traffic, see the section called “Getting VoIP traffic to the Oreka Server” .

Chapter 4. OrkWeb - OrkTrack

Table of Contents

Overview

OrkWeb and OrkTrack are two components of the Oreka package that are deployed and installed together and are often referred to simply as OrkWeb.

The actual OrkWeb component is the user interface that is accessible through any browser, while the OrkTrack component is mainly responsible for receiving metadata about recordings from OrkAudio and storing them in the database. See diagram below for a usage example.

Installation

On Linux

Here are the steps to install OrkWeb on CentOS or Red Hat Enterprise Linux (RHEL). For assistance with other Linux distributions, contact <support@orecx.com> .

Requirements

- You need to be logged in as "root"
- You need an internet connection to download MySQL, Java and Tomcat.

As mentioned earlier, OrkWeb and OrkTrack require a database engine (preferably MySQL), Java and Tomcat. Java and Tomcat are downloaded and installed by the OrkWeb installer provided to you by OrecX. MySQL, on the other hand, needs to be downloaded and installed separately.

Installation

- If the database server (MySQL) is not already installed, install it: e.g. `yum install mysql-server`
- If the MySQL service is not already running, start it: e.g. `service mysqld start`
- Make sure that MySQL will restart automatically after a system reboot: e.g. `chkconfig mysqld on`
- Untar the OrkWeb installer provided to you by OrecX: e.g. `tar -xvf orkweb-1.7-2586-x64-linux-installer.sh.tar`
- If the Apache Tomcat service is already installed and is running, stop the service: e.g. `service tomcat stop`
- Run the installer: `./orkweb-1.7-2586-x64-linux-installer.sh`

The OrkWeb installer will prompt you for the MySQL "root" user password. By default, MySQL is installed with no default password. If you had set one, enter it here.

The installer will then prompt you for the installation of Java and Tomcat. Accept the default directories. It will then install OrkWeb and OrkTrack under Tomcat, and configure the Tomcat service to be automatically started after a reboot. However, it will *not* run the Tomcat service at the end of the installation. You will need to start it yourself after you apply the license file, as described in a later section.

On Windows

Requirements

- You need to be logged in as Administrator before proceeding.
- Access to the internet is highly recommended for download of the Oreka software, and MySQL.

As mentioned earlier, OrkWeb and OrkTrack require a database engine (preferably MySQL), Java and Tomcat. Java and Tomcat are downloaded and installed by the OrkWeb installer provided to you by OrecX. MySQL, on the other hand, needs to be downloaded and installed separately.

Installation

- Install MySQL: you can download the MySQL Windows Essentials for your platform at http://dev.mysql.com/downloads/mysql/5.6.html#win32 . Install it accepting all the defaults. Take note of the "root" password if you assign one. It will be needed when you install OrkWeb.
- If Apache Tomcat service is already installed and is running, stop the service
- Install OrkWeb: unzip the OrkWeb installer provided to you by OrecX, `e.g. orkweb-1.7-2586-win32-installer.zip` .
- Run the embedded executable. This will first install Java, then Tomcat (accept all default options) and finally OrkWeb and OrkTrack.
- Make sure to enter the correct MySQL "root" password when installing OrkWeb. You can always update it later in the database.hbm.xml file in the OrkWeb installation folder, `C:\Program Files\OrkWeb` by default.

The installer configures the Tomcat service to be automatically started after a reboot. However, it will not run the Tomcat service at the end of the installation. You will need to start it yourself after you apply the license file, as described in a later section.

Upgrade

Upgrade using installer

- Back up all files under the configuration folder (typically /etc/orkweb on Linux, and C:\Program Files\OrkWeb in Windows).
- Back up the database, e.g.: `mysqldump -uroot -p<password> oreka > orekaDB.sql`
- Download the OrkWeb installer, e.g. in Linux: `wget http://orecx.com/mycompany/orkweb-1.7-2586-x64-linux-installer.sh.tar`
- Stop the Tomcat service, e.g. in Linux: `service tomcat stop`
- Untar or unzip OrkWeb the installer
- Install OrkWeb, e.g. in Linux: `./orkweb-1.7-2586-x64-linux-installer.sh.tar --nomysql --notomcat --nojava` . In Windows, simply ignore the installation of Java and Tomcat.
- If any custom changes were made to the old `database.hbm.xml`, `logging.properties` or other configuration files, apply those changes to the new files. Ensure that the database credentials are updated correctly in `database.hbm.xml`. Do not simply overwrite the new files with the old ones, some configuration settings might have changed between the two versions.
- Re-start the Tomcat, e.g. in Linux: `service tomcat start`

Upgrade using WAR files

To upgrade using .war files use the procedure below. The version of the .war file made available to you by OrecX may differ from the example below, but the procedure still applies.

- Get the orkweb.war and orktrack.war files. Sometimes libraries may be required too. E.g. `orkweb-1.8-2588.war` , `orkweb-1.8-2588.war` and `oreka-tomcat-java-deps-1.7-2550.zip`
- Backup the database, e.g.: `mysqldump -uroot -p<password> oreka > orekaDB.sql`
- Stop the Tomcat service, e.g. in Linux: `service tomcat stop`
- If libraries need to be upgraded, move the existing `$tomcat/shared/lib` folder to a safe location, then unzip the libraries zip file provided to you by OrecX (e.g. `oreka-tomcat-java-deps-1.7-2550.zip`) into the `$tomcat/` folder. This will re-create the `$tomcat/shared/lib` folder with the new library files.
- Move the existing `$tomcat/webapps/orkweb` and `$tomcat/webapps/orktrack` folders to a safe place **somewhere else than under `$tomcat/webapps`** , you will need them in a later step.
- Copy the orkweb war file to `$tomcat/webapps/` **`orkweb.war`**
- Copy the orktrack war file to `$tomcat/webapps/` **`orktrack.war`**
- Restart the tomcat service. It should create new `orkweb` and `orktrack` folders under `$tomcat/webapps/`
- Edit the new `web.xml` files in the `$tomcat/orkweb/WEB-INF` and `$tomcat/orktrack/WEB-INF` folders and update the ConfigDirectory and TomcatHome parameters to point to the appropriate folders (in Windows, the defaults are `C:\Program Files\OrkWeb` and `C:\Program Files\Apache Software Foundation\Tomcat 7.0` respectively; on Linux, `/etc/orkweb/` and `/opt/tomcat7/`)
- Restart the tomcat service

If you prefer a more automated way of running this procedure, you can download and use the upweb.sh script available at http://files.orecx.com/tools/upweb.sh (usage example: ./upweb.sh 1.8-2588). Please read the header of the file for instructions on the steps you need to

execute manually before and after running the script. For the Windows platform, you need to have Cygwin installed to use this script.

Applying OrkWeb License File

To apply the OrkWeb license file, e.g. `orkweb-30-days-trial-license-20131005.txt` , copy and paste its content into the Input License box in OrkWeb, accessible from the login page the first time the software is accessed, or from the Account page at subsequent tries.

Trial versus Production license

Note that there are two types of license files: trial and production. The trial type is sent to you by OrecX as a first license and allows you to record *everything* on the wire. It is good to start with this type of license to uncover any configuration tweaks that may be necessary.

For example, you may be seeing only the RTP streams but not the control packets, thus no phone number or extensions can be associated to calls. With a trial license, you will be able to quickly detect such a situation and correct it since all calls would be recorded and would appear in the OrkWeb Browse page with the local and remote party typically showing as IP addresses.

With a production license, users *must* be configured in OrkWeb with their phone numbers or extensions (or other) as a login string, for Oreka to be able to associate the local party in the VoIP packets with the configured user and thus recording the call. If no users have been configured in OrkWeb, a production license will inhibit all recording. See also the section called “Login Strings”

Before migrating to a production license, make sure all extensions, phone numbers, SIP URIs, ... that need to be recorded, are configured as login strings for defined users in OrkWeb. The login string field must match what you see in the local party field in the Browse page.

Files Location

OrkWeb/OrkTrack Configuration files

OrkWeb and OrkTrack have a set of configuration files that allow them to know where and how to access the database, and what information to write to log files. The main configuration files are shared between OrkWeb and OrkTrack and can be found under `/etc/orkweb` in Linux, and typically under `C:\Program Files\OrkWeb` in Windows. These folders contain 5 main files:

- `database.hbm.xml` : for application database access information.
- `logging.properties` : for application logging configuration.
- `localpartyMap.csv` : for mapping the local party seen by orkaudio to the one configured in orkweb (e.g. IP address to an extension).
- `remotepartyMap.csv` : for mapping the local party seen by orkaudio to the one configured in orkweb.

The OrkWeb and OrkTrack web applications' web.xml files have two important parameters in them: ConfigDirectory and TomcatHome. The former one must point to the installation folder (e.g. `/etc/orkweb`).while the latter points to the container folder (e.g. `/opt/tomcat7`).

OrkWeb/OrkTrack Log files

Both the OrkWeb-OrkTrack applications, and the Tomcat web server have their own logging mechanisms. Below are the files of interest for both cases:

- `orkweb.log` : this file contains messages logged by OrkWeb and OrkTrack. the level of logging is defined in the `logging.properties` file. It can be found in `/var/log/orkweb` under Linux and typically in `C:\Program Files\OrkWeb` in Windows.
- `catalina.out` : this is the file where Apache Tomcat logs its own messages. It resides in the `$tomcat/logs/` folder.

Running OrkWeb - OrkTrack

By default, OrkWeb and OrkTrack use port 8080. Thus, ensure that port 8080 is open on the server. In Linux, you need to look at iptables, while in Windows, you can check your Firewall settings from the Control Panel.

Before starting OrkWeb and OrkTrack, ensure that the database server, typically MySQL, is running. Once done, start the Apache Tomcat service which will launch OrkWeb and OrkTrack.

On Linux

You can use `ps -ef | grep mysqld` to verify if mysqld is running, and `service mysqld start` to start it if it is not.

Once the MySQL is running, start the Apache Tomcat web server, e.g. `service tomcat start` .

On Windows

Go to Start/Run... and type services.msc. This opens the Services Manager application. Ensure that the MySQL service is running. If not, start it.

Once the MySQL service is running, start the Apache Tomcat web server, by right-clicking on the Apache Tomcat service and selecting Start.

Accessing OrkWeb

Open any standard web Browser and type the following URL: http://localhost:8080/orkweb . If you are accessing from a location other than the server on which Oreka was installed, replace *localhost* with the hostname or IP address of the Oreka server.

This will bring up a login screen as the one shown below. Login as admin/admin.

If this is the first time you attempt to login after the installation, you will be presented with a license input screen. Copy the content of the license text file issued to you by Orecx and paste it into the text box.

Managing Users

OrkWeb users are managed in the Admin/users page. It is possible to create, edit, disable and delete users. You can also configure users to be recordable or not. Non-recordable users are typically users who have administrative privileges, and do not count against the licensed user limit.

Your license key limits the number of active users you can have at any point in time. To free a few licenses, you can delete users or simply disable them. Being disabled, they will still be visible but no new recordings will be associated to them and those disabled users won't be able to log into OrkWeb.

There is one pre-defined and non-editable user in OrkWeb: "admin". This user has all the possible privileges or access policies enabled, and is reserved for the main admininstrator.

Login Strings

Each user can have multiple login strings entered as a cvs list of text strings. Those login strings serve two purposes:

- They act as unique identifiers for OrkWeb users.
- They specify what phone numbers or extensions are to be recorded.

When a new call occurs, if the local party field, as would be seen in the OrkWeb Browse page, matches one of the user's login strings, the recording will be associated to that user. **The login string matching is case sensitive** .

For example, Tom's login strings might be " *tom, 5312* ". Any call to extension 5312 will be associated to Tom, and Tom will be able to log into OrkWeb using either " *tom* " or "*5312*" as an identifier (assuming Tom has a password defined. OrkWeb login is denied for users with blank passwords).

Configuration of users and login strings is very important for correct behavior when using production licenses. Refer to the section called “Trial versus Production license”

Importing CSV List of Users

When you have a great number of users to define in OrkWeb, it could be a daunting task to create them all, one at a time, from the Admin/users page. In this case, you can use the Import Users feature available on that page. This feature allows you to import a comma-separate list of users directly into OrkWeb. Below are the fields that define the file format:

```
firstname, lastname, login string, password, recordable, force password change, email, group
```

Only the firstname and lastname fields are mandatory. The rest can be left blank if not required. Also, only one login string can be entered on one line. The "recordable" and "force password change" fields can be set to "false"/"true", "0"/"1" or "no"/"yes". The "force password change" field, if set, forces the corresponding user to change their password the next time they login. This is mainly useful when defining new users.

There are two other global settings that can be configured when importing users: "Keep old login strings" and "Recordable". "Keep old login strings" allows you to select whether a user in the csv list that already exists in OrkWeb should have their login string(s) replaced by the login string in the imported file, or whether the new login string is to be simply added to that user's existing login string(s).

If a login string already exists in the database with different firstname/lastname, no action is executed.

Local Party Mapping

Sometimes, the local party reported by the OrkAudio appears in OrkWeb in a format that does not meet your requirements, e.g. MAC address, IP address, ... In cases where this cannot be modified at a configuration level neither at the recorder level nor at the telephony platform level, Oreka provides you with a special tool to circumvent the issue: local party mapping.

To map the local party to an extension (or other) to meet your requirements, create or edit a `localpartyMap.csv` file in the OrkWeb installation folder, and add the entries following the example below:

```
10.10.1.1, 1540
          10.10.1.2, 1541
          00:08:5d:13:19:a0, 3523
          SIP/SOFTHONE034, 3681
          ...
```

The first entry is what you want to replace, the second, is the target output.

Once this file is completed, restart Tomcat (or at least OrkTrack), for the new changes to take effect.

Remote Party Mapping

This is the same as the local party mapping above, but for the remote party. The file name is `remotepartyMap.csv` in this case.

User Auto-Provisioning (UAP)

Oreka may be configured to automatically provision users based on recordings, i.e. whenever a new recording is detected, its local party can be automatically provisioned in OrkWeb. This feature is mainly useful in active recording setups using SIPREC, whereby the decision of what users to record is pre-configured in the telephony platform, hence ensuring that auto-provisioning will only occur for required users.

To configure UAP, first the OrkAudio config.xml configuration file needs to be updated as follows (OrkAudio needs to be restarted for this change to take effect):

```
<SipUAPlugin>
             <SipRecReportLocalNameAsTag>yes</SipRecReportLocalNameAsTag>
            </SipUAPlugin>
```

Then, the following setting must be enabled in the OrkWeb database:

```
orkuserconfig.userAutoProvisioning = true
orkuserconfig.uapNameSplitChars    = "_. "  or " "
```

The orkuserconfig.uapNameSplitChars field is usually required to contain a blank space.

The user first and last name will be obtained from the OrkAudio recorder typically in the format "firstname lastname", and will be parsed accordingly to provision the user.

Some important notes on how UAP works:

- With UAP, the recording's local party (i.e. user's login string) is key. If the local party is not found as a configured login string in orkweb, it is provisioned and associated to a new **recordable and external** user with the corresponding first and last name. This is regardless of whether another user by the same first and last name already exists (the login string is unique in Oreka, but the not the user name.)
- UAP is subject to user and login string license limits. Once reached, no new users are auto-provisioned.
- UAP not only adds new users, it can also update existing ones. This is true only if the login string already exists and is associated to a user who is marked as **external** .

If you need assistance, please contact <support@orecx.com> .

User Authentication with Single Sign On (SSO)

OrkWeb can be configured to act as a single sign on consumer, hence allowing user login based on authentication from an external platform. This may be complemented with user auto-provisioning (SSO UAP), which would automatically add/update the user to the OrkWeb database if SSO authentication succeeds.

When SSO is configured, users attempting to log into OrkWeb will be authenticated by the external platform instead of by OrkWeb. This occurs if the user already exists in the OrkWeb database as an "external" user and is not disabled, or if it is not in the database and SSO UAP is used (see section below). If the user exists as "external" but is disabled, login will be denied.

Two SSO sources (producers) may be configured for better reliability if needed. See ssoURL and ssoURLSecondary parameters below.

Here is a list of parameters to configure in the database to enable SSO functionality for Broadworks and LDAP:

Broadworks SSO configuration example

```
orkuserconfig.singleSignOn = true
orkuserconfig.ssoType = "Broadworks"
orkuserconfig.ssoURL = "https://xsp.bwvoip.net/com.broadsoft.xsi-actions/v2.0/user/$userId/profile"
orkuserconfig.ssoURLSecondary = "https://secondary.source.com/v2.0/user/$userId/profile"
```

LDAP SSO configuration example

```
orkuserconfig.singleSignOn = true
orkuserconfig.ssoType = "LDAP"
orkuserconfig.ssoURL = "ldap://ldap.forumsys.com:389/"
orkuserconfig.ssoAuth = "simple"
```

```
orkuserconfig.ssoPrincipal = "cn=$cn,dc=example,dc=com"
```

- The ssoType currently supports "Broadworks" and "LDAP". The field is not case sensitive.
- The ssoURL is the URL that OrkWeb will use to request authentication from the SSO producer. The above examples are typical URL used for Broadworks and LDAP. $userId is a parameter that will host the login user name for Broadworks.
- The ssoPrincipal field can accept one of two LDAP parameters to host the user name: $cn or $uid.

Single Sign On User Auto-Provisioning (SSO UAP)

SSO may be complemented with user auto-provisioning functionality. In that case, if the user attempting to log into OrkWeb does not exist, it is auto-provisioned with basic access policies ("Users" security group), and is set to be "external". Otherwise, if it already exists as an "external" user, its attributes (mainly first name, last name and email) are updated if necessary.

```
orkuserconfig.ssoUap = true
```

If you need assistance, please contact <support@orecx.com> .

Managing Groups

Oreka has two types of groups:

- Security groups: This represents a privilege level and might be called a *role* in other software systems.
- Regular groups: This represents a logical group of users (e.g. sales, marketing)

The combination of these two types of groups enables fine-grained access to the different features and collected data.

Security groups

Every user in Oreka belongs to one *single* security group which determines the access privileges for that user. Newly created users are implicitely associated to the default "Users" security group. There are 4 pre-defined groups in OrkWeb. More security groups can be created but it is typically much easier to adapt the existing security groups for your needs than to create completely new ones. The pre-defined security groups have sensible access policies defaults that are fine for most needs. Also, the pre-defined security groups cannot (and should not) be deleted:

Pre-existing security groups are:

- Users
- Administrators
- Group Administrators
- Supervisors

Regular groups

Regular (non-security) groups can be useful e.g. for filtering or creating specific recording rules on a group of people. Users can belong to multiple regular groups. Groups can also be part of groups, thereby creating a group hierarchy of any depth.

Access Policies

When you click on the "view" link next to a group in the Admin/groups page, a "Managed Access Policies" button appears. Clicking on it displays the Access Policies definition page. You can create your own security groups and give them the access policies that you wish but it is easier to tweak an existing security group.

Users associated to a particular Security Group inherit that group's access policies.

Example 1: Call center with Agents and Supervisors

In this example, call center agents shall be able to see their own recordings, supervisorA shall be able to see recordings for groupA and groupB, supervisorB shall be able to see recordings for groupB only. Here are the steps to configure this:

- Create one recordable user per recorded extension
- Create one non-recordable user per supervisor
- Create regular group A
- Create regular group B
- Add every agent user to either groupA or groupB (it is possible to add certain agents to both groups if wanted)
- Add supervisorA to groupA and groupB
- Add supervisorB to groupB only
- Add both supervisors to the Supervisors security group

Example 2: Business hosted telephony provider

In this example, each end-customer is a company with several users and each company shall be mapped to a group. CompanyA shall be mapped to groupA, CompanyB shall be mapped to groupB. Within each company, one person shall be administering the system. Only this person might be abe to access the live monitoring system. It shall not be possible for any company user to see data for another company, or even know that it exists. Here are the steps to configure this:

- Create one recordable user per recorded extension for both companies
- Create one non-recordable user per supervisor, e.g. supervisorA1 and supervisorA2 and supervisorB1
- Create one non-recordable user per company administrator, e.g. groupadminA and groupadminB
- Create regular groupA for companyA
- Create regular groupB for companyB
- Add every recordable user to either groupA or groupB
- Add supervisorA1, supervisorA2 and groupadminA to groupA
- Add supervisorB1 and groupadminB to groupB
- Add all supervisors to the Supervisors security group
- Add groupadminA and groupadminB to the Group Administrators security group
- Edit the access policies of the Supervisors and Users security group in order to remove the live monitoring privilege altogether. The Group Administrators can access live monitoring by default, so no need to change access policies there.

Programs

Overview

In order to restrict what is recorded by Oreka, it is possible to create so-called recording programs from the Programs page. Those programs let OrkWeb administrators specifiy recording schedules as well as filtering criteria. Any number of these programs may be created to achieve high complexity recording rules.

Very important note: when at least one audio or screen recording program is active, recordings become subject to Programs and are no longer retained by default.

A program is initiated by a *trigger* event, which causes a set of *criteria* to be evaluated. If the *criteria* are met, a corresponding *action* is taken. Hence, the Programs feature's usefulness is not limited to selective recording. The criteria-based functionalities allowed by programs are:

- Selective audio recording
- Selective audio and screen recording
- Selective screen recording (scheduled)
- Selective copying, moving, deleting or emailing of recordings (Media Manager)
- More to come in the future

The *Copy, Move, Delete and Email recordings* programs collectively define what we call the **Media Manager** feature.

Triggers

Triggers are events such as a pre-configured time or the completion of a recording. *Scheduled* (time-based) triggers may be one-time events or repeatable triggers that restart at a given frequency.

One-shot triggers are configured by setting the "Trigger first execution" field. Recurring triggers also set the "Trigger repeat period (in days)" field which establishes the frequency of the trigger.

Only Media Manager programs currently use those explicit triggers. *Audio recording* programs have implicit triggers based on the completion of the recording process.

Criteria

Criteria are conditions that are evaluated when the trigger event occurs. If met, they cause the program action to be executed. Examples of criteria are target user, target group, local party, remote party, ... The available criteria depend on the type of program in question. The "Handle as exclusion criteria" option ensures that the action is taken only if the criteria are NOT met.

The criteria list may vary between different programs. Local and remote party criteria where available, support regular expressions (as opposed to wildcards). For example, in order to retain all recordings made to or received from phone numbers starting by the digit 5,6 or 7, enter "[5-7].*" in the remote party field of your program.

Be careful with the Target group criteria: it applies to the recording's user's parent groups as well as all their subgroups.

Scheduled Media Manager programs include among other, the following criteria:

- **Retention period (in days):** this specifies the upper boundary of (latest) recordings that may be copied, moved or deleted. Recordings newer than that boundary are kept intact.
- **Earliest timestamp:** earliest timestamp is updated automatically by the software every time a new batch of recordings is processed. It sets the earliest recording that should be considered, and ensures that recordings previously evaluated don't get re-processed the next time the program is executed.

Actions

Actions are executed when the program is evaluated and the criteria are met. Actions may be one of the following:

- Keep an audio recording.
- Keep an audio/screen recording pair.
- Star/stop a screen recording.
- Copy recordings (Media Manager).
- Move recordings (Media Manager).
- Delete recordings (Media Manager).
- Email recordings (Media Manager).

Selective Recording (Audio and/or Screen)

Selective recording can be achieved with the *Audio recording* and *Screen recording (scheduled)* types of programs.

The basic *Audio recording* program will simply keep or discard a recording based on the defined criteria. The decision is made when the recording completes. Almost any recording metadata may be used as a program criteria including local party, remote party, call direction, duration, user associated to the recording, time of the recording, ... There is also a "recording percentage" criteria that ensures that only a certain random percentage of recordings is kept.

The *Audio recording* program may also be used to trigger screen recordings that are to be associated to the audio recordings with the "Trigger Screen Recording?" option. This option is only available when Screen Recording is licensed. In this scenario, when an audio recording starts, the program in question automatically triggers a screen recording. It also sends a message to the screen recorder to stop the recording when the audio recording completes, and associates it to the audio recording counterpart.

The other type of selective recording is *Screen recording (scheduled)* . This is for screen recordings only, and provides the ability to configure time-based trigger events for starting and stopping the screen recording (e.g. record only between 10AM and 2PM, or only on Thursdays, ...).

Media Manager (Copy, Move, Delete, Email recordings)

The Media Manager (MM) functionality provides a mechanism to apply certain actions to **existing** recordings. The Media Manager is essentially an extension of the File Manager and Auto-Delete functionalities. The File Manager and Auto-Delete are global in scope and do not permit the definition of criteria. The Media Manager on the other hand provides finer granularity functionality, with the addition of criteria-based execution.

These MM actions may be scheduled (time-based), or triggered when a recording completes. The latter is available for the copy and move programs, but not for the delete program (since it is equivalent to an *Audio recording* program that would not keep such a recording).

For copy and move programs, the actions are defined by the following parameters:

- **Target service:** just like the Target service in the File Manager, this field must point to a File Server type service. It designates the target location where the file is to be copied or moved. Refer to the section called “Services” for more details.
- **Path name (string substitution):** Edit the secondary "Pathname" field if the target file's path is to be different from the original file's. Note that this refers only to the *Secondary pathname* not the *Primary pathname* . The primary path refers to the "File absolute path" of the service in question, e.g. `/var/log/orkaudio/audio` , while the secondary path refers to the <TapePathNaming> path configured in the recorder's config.xml. The default being a date-based hierarchy e.g. 2009/11/12/09/20091110_154500_ABC.wav. You can heance use this field to customize your path name based on recording metadata. See the section called “File and Path Names in OrkWeb” for the list of string substitutions available for path naming in OrkWeb.
- **File name (string substitution):** Edit the "Filename" field if you wish to customize the name of the target file. See the section called “File and Path Names in OrkWeb” for the list of string substitutions available for file naming in OrkWeb.

Delete programs have the following action parameters:

- **Delete files in file system:** enable if actual media files need to be deleted.
- **Delete entries in database:** enable if deleting the recordings' metadata in the database is required.

Below are but a few examples of the many Media Manager uses:

- Configuring different delete retention periods for different groups.
- Moving recordings to different location depending on underlying customer (group).
- Backing up recordings for only a handful of groups.
- Renaming media files differently based on different criteria.
- Emailing media files immediately after they are written to the associated user.
- ...

Programs Priority

Since multiple programs may be defined, a legitimate question that often comes up is: what happens when a given recording meets the criteria of two or more programs? Which program(s) gets priority and which one(s) decides on the fate of the recording?

The answer is that it depends on the priority configuration. By default all deployments give priority to "negative" programs, i.e. to programs that would cause the rejection of a recording. Hence, even if many programs report that a recording should be kept, all it takes is for one program to reject the

recording for it to be discarded.

This behavior may be reversed in the database configuration to ensure that one "positive" program guarantees that the recording gets kept.

All recording programs get evaluated for every new recording. When multiple programs agree that a recording should be kept, they all get associated to that recording.

Live Monitoring (On-Demand Recording)

OrkWeb offers a Live Monitoring feature that allows you to see what calls are occurring at any given time. You can listen to these calls live and opt to record them or not (keep and discard options in the Live Monitoring page.)

In order to configure OrkWeb for Live Monitoring, it is necessary to create users for all phone extensions you want to monitor and at least one group. All wanted users must then be added to the group(s). Please refer to the section called “Managing Users” . Once done, the live monitoring page will list all groups for selection.

See also the section called “Live Monitoring” for configuring OrkAudio

Note that by default, OrkAudio records all calls that appear in Live Monitoring, unless programs are defined and exclude such recordings. If you wish to operate in a purely on-demand recording fashion based on the Live keep/discard options, contact <support@orecx.com> for how to set up OrkAudio to default to a non-recording mode.

Multiple Server configuration (OrkWeb)

Before you start, check that you have prepared all your orkaudio servers as described in the section called “Multiple Server Configuration (OrkAudio)”

Replay modes

When running multiple servers, it is necessary to use the **standard replay** mode or the **centralized replay** mode. This setting can be configured in the OrkWeb config page.

Standard Replay: use this if you want the client media players to stream the audio data directly from the OrkAudio server where it has been recorded.

Centralized Replay: use this if you need all audio to be relayed by the OrkWeb server. This can be useful when allowing access to OrkWeb on a public IP address where you need everything to go through a single TCP port (the Tomcat TCP port).

Simple Replay: This mode is only for single server deployments. This is the default setting.

Changing the ports used to serve media files

If media files are served from a recorder (via tomcat, or apache httpd, or equivalent) using a port that is different from the default port 8080, make sure that the "File serve port" of the service associated with the recorder in question in config/services has the correct value. For more information on services, please see the section called “Services” .

Auto-Delete

The Auto-Delete feature may be used to automatically delete files and their metadata in the database after a certain period of time as defined by the "retention period" parameter. When activated, it runs as a background task, at the frequency set by the "wake-up period" parameter.

This feature is global in nature and applies to ALL media files in the system. This is true even for setups with multiple recorders installed on different servers. To delete recordings based on specific criteria (e.g. group, service, ...), refer to the section called “Media Manager (Copy, Move, Delete, Email recordings)” , available as of OrkWeb version 1.8.

The Auto-Delete feature runs as a background task. When it wakes up, it searches in the database for all recordings that are older than the retention period. It then deletes recording files (e.g. .wav for audio and .fbs for screen recordings), in batches of one day, starting from the oldest.

Configuration Example

Below are the instructions for how to configure Oreka to delete all media files in the system and their database info, if they are older than 180 days (approximately 6 months):

- Set "Delete files in filesystem" to Enabled
- Set "Delete entries in database" to Enabled
- Set "Wake-up period" to the frequency at which this task is to run. Typically, 1 day (86400 secs) is sufficient.
- Set "Date/time of first execution" as desired. This is mainly useful for postdating the start of execution of this feature.
- Set "Retention period" to the number of days recordings are to be kept in the system before being deleted.

The "Stop Auto-Delete" button provides a quick mean for turning off this feature. It resets the "Delete files in filesystem" and "Delete entries in database" to Disabled, and "Wake-up period" to 0.

Deleting Remote Files

For deletion of remote files, an ssh server needs to be running on the remote server. Also, the corresponding service must be configured with the ssh parameters in OrkWeb, in the Config/services page. Refer to the section called “Services”

for more details on configuring services.

Notes

It is recommended to delete both files and database info with this feature. Deleting only database info leaves orphaned recordings that are neither easily accessible nor identifiable, while deleting files only and leaving the corresponding data in the database means that the Auto-Delete feature will unnecessarily re-attempt to remove already deleted files every time it wakes up.

Services

A service is a logical entity that represents an audio or a screen recorder, a media file storage server or an SMTP server for sending emails. Recorder services are created automatically by Oreka, while file storage and SMTP services need to be manually created by the administrator.

When a recorder initially starts, it sends its information parameters to OrkTrack which creates the corresponding service in the database, if it does not already exist. A recorder service may be later edited by the user, e.g. for configuring SSH access information.

File storage services are typically used with the File Manager or Media Manager features to designate a target location where to move or copy files. File storage services must be created by the user. The main attributes to configure for file storage services are the service name, service type (File Storage), hostname, file absolute path and SSH access parameters if applicable.

The following is a list of attributes that describe a service.

- Service ID: unique ID describing the service, and defined internally by Oreka.
- Service name: arbitrary name of the service. Audio and screen recorders automatically assign this field.
- Service type: audio recorder, screen recorder or file server.
- Description: text field for optional description of service.
- File server protocol: protocol used to access files on the underlying server, typically "http".
- Host name: host name or IP address of the server.
- Application port: assigned by the recorders, typically 59140 for audio recorder, 59170 for screen recorder. This field is not required for file storage services.
- File serve port: port to access files on the server. Typically 8080.
- Context path: this field should match a corresponding entry in Tomcat's server.xml file on the server described by this service and is used to construct the URLs for accessing files. The defaults are "/audio" for the audio recorder and "/screen" for the screen recorder.
- File serve path: this field is appended to the context path parameter to provide more flexibility in URL construction.
- File absolute path: used by features such as Delete, Auto-Delete and File Manager to tell OrkWeb the exact location on the server where the files reside. Examples are C:\Oreka\Audio in Windows and /var/log/orkaudio/audio in Linux.
- Streaming TCP port: indicates port on which streaming will occur. Used mainly by OrkAudio for the Live Monitoring feature.
- Local: indicates whether the service resides on the same server as OrkWeb (local), or on a remote server. If Local is unchecked, i.e. the service resides on a remote server, the SSH parameters that follow are required for accessing files associated to this service.
- SSH user name: the user name to access the server using SSH.
- SSH password: the password of the user name for accessing the server using SSH.
- Re-type SSH password: validation of the password of the user name for accessing the server using SSH.
- SSH port: the SSH port for accessing the server, typically 22.

File Manager

The File Manager feature allows copying, moving and renaming media files after they are created by the recorder. Copying media files is also referred to as "archiving".

The File Manager may be used in different scenarios. Here are some examples:

- Multiple recorders are being used and all media files need to be centralized at one location;
- All recordings are to be copied (archived) to another location for backup.

To activate the File Manager, select the archiving method and running mode as described below, and uncheck the Disable box, then "Submit" the changes.

This feature is global in nature and applies to ALL media files in the system. This is true even for setups with multiple recorders installed on different servers. To copy or move recordings based on specific criteria (e.g. group, user, ...), refer to the section called “Media Manager (Copy, Move, Delete, Email recordings)” , available as of OrkWeb version 1.8.

Running Modes

The File Manager can run in one of two modes: real-time or scheduled. For now, these two modes are mutually exclusive. The sections below give more details.

Real-Time (Immediate) Mode

In real-time or immediate mode, all media files that appear *after* the File Manager is activated will be archived or moved, almost immediately after the recording completes. What actually happens is that every time a new recording occurs, a corresponding entry is added to a queue to be processed by a background task. The processing of the queue performs the necessary task(s) such as archiving, moving and/or renaming the target file.

Since the processing occurs almost in real-time, **the destination storage location needs to be accessible at all times.**

To configure this mode, check the box next to "Immediately after recording completes".

Scheduled Mode

The scheduled mode provides a tool to archive or move files on a regular basis, at a pre-configured time of day and with a pre-defined criterion based on the age of the recording. Unlike in the real-time mode, the destination storage location in this case only needs to be accessible at the time when the File Manager is scheduled to wake-up.

To configure the scheduled mode, check the box next to "Every X day(s) ...". Make sure to fill in are fields on that line.

Configuration

- Uncheck the "Disable" checkbox to activate the feature.
- "Moving files" is used to move files, thus deleting them from their original location, while "Archiving" copies files to a new location, keeping the original file intact.
- Check/uncheck the "Access files from original location" checkbox, depending on whether playback of the recordings in OrkWeb is to occur based on the original file location or its new copy - valid only for archiving option.
- Select the running mode by checking the box next to "Immediately after recording completes" for the real-time mode, or "Every X day(s) ..." for the scheduled mode. The two modes are mutually exclusive. If you select the scheduled mode, you need to specify the time of day when the File Manager is to run, as well as a retention period criterion (0 means all files).
- Select the target service where the archiving is to occur (see the section called “Services” for adding services). **The target service MUST be different from any recording service in use.** It is best to define a new service of type File Server and properly define its attributes especially its File Absolute Path and its ssh parameters.
- Edit the secondary "Pathname" field if the target file's path is to be different from the original file's. Note that this refers only to the *Secondary pathname* not the *Primary pathname* . The primary path refers to the "File absolute path" of the service in question, e.g. `/var/log/orkaudio/audio` , while the secondary path refers to the <TapePathNaming> path configured in the recorder's config.xml. The default being a date-based hierarchy e.g. 2009/11/12/09/20091110_154500_ABC.wav. You can hence use this field to customize your path name based on recording metadata. See the section called “File and Path Names in OrkWeb” for the list of string substitutions available for path naming in OrkWeb.
- Edit the "Filename" field if you wish to customize the name of the target file. See the section called “File and Path Names in OrkWeb” for the list of string substitutions available for file naming in OrkWeb.

File and Path Names in OrkWeb

It is possible to configure the media files path and filename as a combination of the following dynamic parameters. Note that the pathname refers only to the secondary part (*Secondary pathname*) of the pathname that typically defaults to `year/month/day/hour/` .

- [orkuid]: the unique tape UID
- [segid]: the segment ID number
- [direction]: the direction, i.e. IN, OUT, UNKNOWN
- [shortdirection]: the short representation of the direction, i.e. I, O, U
- [localparty]: the local party field, typically containing a phone number or extension
- [remoteparty]: the remote party field, typically containing a phone number or extension
- [hostname]: the hostname obtained from the service associated to the recording
- [year]: the year in 4 digit format (e.g. 2009)
- [yy]: the year in 2 digit format (e.g. 09)
- [month]: the month, in 2 digit format (e.g. 02)
- [mm]: the month, in 2 digit format (e.g. 02)
- [day]: the day, in 2 digit format (e.g. 08)
- [dd]: the day, in 2 digit format (e.g. 08)
- [hour]: the hour, in 2 digit, 24 hour format (e.g. 18)
- [min]; the minutes, in 2 digit, (e.g. 06)
- [sec]: seconds, in 2 digit, (e.g. 06)
- [group]: the parent group of the user associated to the recording. *NOTE: this assumes the user has only one parent group, otherwise the results are unpredictable*.
- [email]: email address of the user associated to the recording.
- [firstname]: first name of the user associated to the recording.
- [lastname]: last name of the user associated to the recording.
- [<tagname>]: a tag type associated to the recording (e.g. [nativecallid])
- [ucid]: UCID associated to the recording, if CTI is used and yields this metadata.
- [trunkgroup]: trunk group associated to the recording, if CTI is used and yields this metadata.
- [trunkmember]: trunk member associated to the recording, if CTI is used and yields this metadata.
- [nativecallid]: native call id associated to the recording, if CTI is used and yields this metadata.

Quality Monitoring (QM)

The Quality Monitoring feature in Oreka allows managers or supervisors with proper privileges to evaluate calls by “scoring” or “marking” them based on a set of predefined criteria called **Scorecards** . The QM feature requires a separate license. A typical usage of QM scorecards is in call centers, where agent calls are often reviewed by their supervisors to ensure quality customer support and guide the agent to improve on weaker areas.

The guidelines or criteria are defined within scorecards. A scorecard may be associated to only one group in OrkWeb. Groups may represent company departments or any other logical entity. They typically include one or several users.

The typical steps involved in setting up and using QM are the following:

- Creating scorecards
- Importing them into OrkWeb
- Scoring calls based on those scorecards and saving the results
- Issuing statitistics reports on scorecard results

We will explore each of those aspects in the sub-sections below.

Creating Scorecards

The first step in setting up QM is to create one or more scorecards. Scorecards establish the criteria that will be used to evaluate a recording. Scorecards may include one or more sections which in turn may host one or more questions. Each question may allow several answers with a different score associated to each. Sections and question scores may also be weighted differently.

Other scorecard features are:

- Auto-fail: some answers may be designed to automatically fail the scorecard results. This allows flexibility in giving great weight to some very important criteria.
- Exclusion: some answers may be designed to have no effect on the final score. For example, a not applicable (N/A) answer typically indicates that the corresponding question is moot, and should be omitted from the scorecard results calculation.
- Comments: there are multiple level of comments that can be added to a scorecard including the scorecard itself, its sections as well as the answers.

Creating a scorecard requires filling a CSV scorecard template file. Please refer to the section called “How do I configure a scorecard for QM?” for more details on the format of the scorecard CSV file.

If a scorecard needs to apply to more than one group, one of two approaches may be used:

- Create two separate identical scorecards, one for each group, with a slightly different name, and associate them separately to each group,
- Create a parent group and make both groups in question its member or children groups, then associate the scorecard to the parent group. The two subgroups will automatically inherit access to that scorecard from their common parent group.

Importing Scorecards

Once a scorecard is created and configured to be associated to a group, it may be imported in OrkWeb in the Admin/scorecards page. Errors in the scorecard are detected and reported by the import function, and will cause the entire scorecard to be rejected.

Scoring calls based on scorecards

To "score" a recording, click on the scorecard icon in the OrkWeb "Browse" page or go to the recording details page and select the "Scorecards" tab. OrkWeb looks for the group(s) associated to recording's user, and makes any scorecard template associated to that user's parent groups available for the evaluation. Once the call is “scored”, the scorecard results may be saved for later review as well as for generating QM statistical reports.

Generating QM reports

Reporting on QM is possible through several reports with the ability to define a slew of filters for greater flexibility. In fact, all search filters that are available in the Browse are applicable to the QM statistics reports below. Reports may be generated in 3 formats: PDF, HTML and CSV unless explicitely specified otherwise. Here is a summary of the types of reports available:

- **Scorecard results report (PDF only)**
- **Group report:** a summary report displaying the number of scored calls, minimum/average/maximum score for each group.
- **Group Detailed report:** a report detailing the number of scored calls, minimum/average/maximum score for each question in each scorecard grouped by groups.
- **User report:** a summary report displaying the number of scored calls, minimum/average/maximum score for each user.
- **User Detailed report:** a report detailing the minimum/average/maximum score for each question in each scorecard grouped by users.
- **Scorer report:** a report listing the score and summary info of every recording grouped by scorers.

Security

Oreka addresses security issues at many different levels. Below is a summary:

- Secure access to the recordings, i.e. access by simple URL can be prohibited in general and allowed only for valid users who are logged into OrkWeb. See below for details.
- Encryption: : OrkAudio may be configured to encrypt files, and OrkWeb configured to decrypted them for playback. Files would thus be played back only through OrkWeb.
- Secure access to the application using SSL (https access).
- Authentication Rules for user login access such as locking a user after a given number of unsuccessful login attempts, and password rules for ensuring a minimum level of difficulty in passwords.

For more information, please contact <support@orecx.com> .

Securing access to your media files

Securing access to media files in Oreka ensures that only users legitimately logged in to OrkWeb are allowed accessing the files. To configure this protection level, two configuration actions are required:

- Set "Secure access to media files" option to "yes" in OrkWeb's Config/settings page. This will automatically set the Playback mode to "Centralized".
- For each server that is storing recordings, configure the web server (e.g. tomcat or httpd) to allow access to the files only from the server where orkweb is installed. Direct URL access to the files from user client PCs will no longer be allowed.
  - For single-server deployments, modify tomcat's server.xml file as follows:

```
<Context path="/audio" docBase="/var/log/orkaudio/audio" >

    <Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="127.0.0.1" deny=""/>

</Context>
<Context path="/screen" docBase="/var/log/orkaudio/screen" >

    <Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="127.0.0.1" deny=""/>

</Context>
```

  - For multi-server deployments, modify every recorder's tomcat's server.xml as above, but change the "allow" field content to include the hostname or ip address of the server where orkweb is running.

Securing access to the application (OrkWeb) using SSL

Secure access to OrkWeb uses SSL, and requires that the URL to the application rely on https instead of http. To configure this functionality, simply edit the `web.xml` file stored under `$tomcat/webapps/orkweb/WEB-INF` , and modify the line:

```
<transport-guarantee>NONE</transport-guarantee>
```

to

```
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
```

After this change, access to OrkWeb becomes https://localhost:8443/orkweb . If you continue using http://localhost:8080/orkweb , you will be automatically re-directed to the new https URL.

Note: the change above assumes that Tomcat was installed by the OrecX OrkWeb installer. The installer performs some customizations to the Tomcat `server.xml` file to make the functionality above accessible. Contact <support@orecx.com> if your `server.xml` file does not contain these customizations.

Foreign Language Support

Oreka's architecture was designed with foreign language support in mind. It is thus very modular and requires minimal effort - translation of a couple of files - to integrate a new language. The localization files are available in OrkWeb application's folders (e.g. `orkweb.properties` for English, `orkweb_fr.properties` for French).

Languages Included

The following foreign languages are currently included in OrkWeb:

- English
- French
- Japanese
- Portuguese
- Spanish
- Korean
- Russian
- Croatian
- Romanian
- German
- Dutch
- Swedish
- Italian

For other languages, please contact <support@orecx.com> .

Changing your Browser's Language

The selection of the OrkWeb language usually occurs automatically by default, and depends on your Operating System language. For example, if you are running the Spanish version of Windows, OrkWeb will appear in Spanish by default.

If you would like to change the displayed language, you would have to use the Language Options in your Browser as described below:

In Internet Explorer:

- Go to Tools/Internet Options
- Select Language/Add...
- Choose the language you want, and put it on top of the list
- Reload the OrkWeb page

In FireFox:

- Go to Tools/Options/Content
- Select Languages/Choose...
- Choose the language and move it to the top of the list
- Reload the OrkWeb page

Migrating OrkWeb to Another Server

To move OrkWeb functionality to a different server with the same operating system follow the procedure below.

Very important: make sure you are operating out of production hours.

On the old server:

- Stop the Tomcat service
- Backup the database, e.g.: `mysqldump -u root -p oreka > orekaDB.sql` . Note that the default database name is "test" with older OrkWeb Windows versions and "oreka" for all OrkWeb Linux versions as well as newer OrkWeb Windows versions
- Save the OrkWeb and Tomcat configuration files, as well as the license file: `database.hbm.xml` , `logging.properties` , `localpartyMap.csv` and `orkl.txt` , and any other file in the OrkWeb installation folder, as well as `server.xml` , in the tomcat configuration folder (`$tomcat/conf/` .)

On the new server:

- Install and start the MySQL service.
- Create the MySQL database on the new server (e.g. "create database oreka;"). **It is very important that the database name created here and imported in a later step below is the same as the one configured in the database.hbm.xml file from the old server.** If the old database is called "test", this would be a good opportunity to rename it to "oreka" in the database.hbm.xml file and in this step.
- Install OrkWeb (refer to the section called “Installation”).
- Restore the OrkWeb and Tomcat configuration files saved from the old server. Double-check that the database name in the database.hbm.xml file matches the one created above.
- Make sure MySQL and Tomcat are both services that restart automatically after reboot.
- Restore the database from the old server, e.g.: `mysql -u root -p oreka < orekaDB.sql` . Again, the name of the database specified MUST match the one in the database.hbm.xml file.
- Start the Tomcat service.
- Important: Make sure that at this point that you can browse the older recordings *before applying new traffic to the server* . If not the case, verify that your database name is correct and that the new database actually contains the same records as the old database.

If you are uncertain about any aspect of the migration process, contact <support@orecx.com> .

Customizing

Customizing the layout, style and colors

If you wish to brand your application by simply changing the logos that appear in orkweb, all you need to do is modify the 3 files under $tomcat/webapps/orkweb/images: `application-logo.png` , `company-logo.png` and `company-logo.jpg` . Make sure to keep copies of those files in case you upgrade orkweb in the future.

Otherwise, customizations of layout, style and color are a simple matter of tweaking html/css/icon files by a web designer. css files are found in $tomcat/webapps/orkweb/css. icon files are found in $tomcat/webapps/orkweb/images and can be modified with a tools such as Phostoshop. It is also possible to modify application html files in $tomcat/webapps/orkweb/WEB-INF e.g. for inserting proprietary content or tweaking the layout. If you do this, you will need to ensure that such changes are re-applied to any future upgrade of orkweb (refer to the next section).

Customizations and software upgrades

Any modification to OrkWeb will need to be re-applied every time a software upgrade is performed. Tools such as diff and patch might help for automating the application of small changes to html and css files, and this, even if the upgraded html and css files have been altered by OrecX since the software version you had customized.

Changing the application name

For example, to access the application as http://server:8080/myrecorder instead, here is the procedure:

- Change folder name under $Tomcat/webapps from orkweb to myrecorder
- Change references to orkweb in $Tomcat/webapps/orkweb/WEB-INF/web.xml as follows:

```
<display-name>myrecorder</display-name>
<servlet-name>myrecorder</servlet-name>

                                                    (This appears in two places in web.xml, change both)
```

- Rename the orkweb.application and orkweb*.properties files in $Tomcat/webapps/orkweb/WEB-INF to myrecorder.application and myrecorder*.properties respectively
- Edit the relevant orkweb*.properties files depending on the languages you want to support and search and replace all references to *OrkWeb* with your own application name.
- Restart tomcat

Accessing OrkWeb without specifying a port number

To access the OrkWeb application without specifying a port number in the URL, such as http://servername/orkweb instead of the typical http://servername:8080/orkweb , you will need to either replace the Connector entry in Tomcat's `server.xml` configuration file for port 8080 with port 80, or simply copy and paste that connector to a new one for port 80, the default Tomcat port. If there are any other references in that file to port 8080, they also need to be modified (or duplicated) to use port 80 instead.

Troubleshooting

Cannot login as admin/admin

Most likely, the database server is down or there is something wrong in the database URL and credentials in the `database.hbm.xml` configuration file.

No recordings are appearing in the Browse page

- Make sure the *end date* in the multi-criteria search form is set correctly.
- Make sure that the recorder (orkaudio) is running correctly and is creating new recordings.
- If you are using a production license, make sure that users are properly defined in the Admin/users page. Refer to the section called “Applying OrkWeb License File” .
- If you have defined any programs in the Admin/programs page, verify that those programs are configured properly to allow the recordings that you are expecting to see.

Cannot replay recordings

If you fail to replay recordings through OrkWeb, here is the checklist:

- Test that the sound device is working properly by trying to play a sound or music file using any available application.
- If you are using Microsoft Internet Explorer, make sure your copy of Windows Media Player works well by opening it from the start menu and trying to replay a sample audio file.
- If you are using Mozilla Firefox, make sure you have installed Java. You can test this using the following link: http://www.java.com/en/download/help/testvm.xml
- Verify that the file you are attempting to replay is actually present in the OrkAudio storage folder as configured in the recorder's `config.xml` file.

```
<AudioOutputPath>c:/oreka/audio/</AudioOutputPath>
```

- Make sure that a correct context path is configured in Tomcat's server.xml for the storage location of the media files. Make this context path matches the OrkAudio AudioOutputPath. Following the example above, ensure that Tomcat's `$tomcat/conf/server.xml` contains an entry such as:

```
<Context path="/audio" docBase="c:/oreka/audio/" ></Context>
```

  If this parameter does not exist already, just add it under the <Host> section. Make sure Tomcat is restarted after such a change.
- If the file is present, try to replay it locally with a media player such as Windows Media Player. If it does not replay, please refer to the section called “Not possible to replay recorded files”
- Make sure URLs are valid: retrieve the full URL to an audio file by doing a "view source" on the browse page and search for ".wav". Copy the complete URL and paste is directly into a browser's address bar to check if the file can be downloaded manually.
- If you are running Oreka on multiple servers, check the section called “Multiple Server Configuration (OrkAudio)”

Cannot login as a user I have just created

You need to add at least one login string for the user. Users can log into OrkWeb using any of the login strings they own. For more details, please refer to the section called “Login Strings”

I don't get the latest recordings

Make sure the *end date* in the multi-criteria search form is not in the past.

OrkWeb's performance has slowed down. What do I do?

Verify that the database indexes were created properly. For a list of indexes, download the script at create_indexes.sql . If the indexes are missing in your database, you will need to run that script. Usage syntax is explained in the header of the file.

Indexing on a large database may take serveral minutes. Hence, it is highly recommended it be done during off hours, when Tomcat is not running.

Chapter 5. FAQ

Table of Contents

How do I manually install orkaudio?

This is not the recommended installation procedure.

Orkaudio comes in two different packagings under Linux : automatic installer (.sh file extension) and RPM archive (.tar file extension). The automatic installer is the recommended way of installing the software. It comes as a single file named e.g. `orkaudio-1.2-6560-x1459-i386.centos5-installer.sh` . To install it, refer to the section called “On Linux”

While the automatic installer works well on CentOS and RHEL, it may sometimes fail. If you run into errors with it, you can always proceed to a manual installation by extracting the .rpm files from it, and installing them manually. The procedure is described below.

To extract the .rpm files, run the installer. At the first question, exit by pressing CTRL-C. This will create a subdirectory under /tmp with all the required rpm files. You can then proceed as follows:

- `yum install boost-devel`
- `yum install libpcap`
- `rpm -i xercesc-2.7.0-1.i386.rpm`
- `rpm -i ace-5.5.8-1.centos5.i386.rpm`
- `rpm -i log4cxx-0.9.7-1.i386.rpm`
- `rpm -i libsndfile-1.0.13-1.i386.rpm`
- `rpm -i orkbasecxx-1.2-660.i386.centos5.rpm`
- `rpm -i intel-ipp_rti-5.0p.x32.rpm`

- `rpm -i orkaudio-1.2-660.i386.centos5.rpm`
- `rpm -i --nodeps orkaudio-addons-1.2-1459.i386.centos5.rpm`

Copy the `orkaudio-startup-script` to the `/etc/init.d` directory as orkaudio. This will allow you to start and stop orkaudio either using `service orkaudio stop` and `service orkaudio start` or `/etc/init.d/orkaudio start` and `/etc/init.d/orkaudio stop`

How do I backup oreka?

This section is intended as a guideline for backing up your Oreka server. It lists all the entities or components that are involved.

- **Media files:** the recordings are your main data. They are stored under the <AudioOutputPath> folder as configured in the recorder's `config.xml` file.
- **Database:** contains the metadata about the media files. For mysql, you can back up using

  ```
  mysqldump -root -p<password> <database_name> > orekadb.sql
  ```

  where database_name is the name of the database, usually "oreka" in Linux, and "test" in Windows.
- **Configuration files:** both OrkAudio's and OrkWeb's. Refer to the section called “Configuration Files” and the section called “OrkWeb/OrkTrack Configuration files” for the files' locations.
- **Log files (if necessary):** both OrkAudio's and OrkWeb's. Refer to the section called “Log Files” and the section called “OrkWeb/OrkTrack Log files”
- **Customizations (if necessary):** if you have made any customizations, make sure to back them up. These may include Tomcat configuration changes, web interface-related tweaks, etc.
- **Oreka software:** ensure that you have a copy of the Oreka software or access to the website where the Oreka software was made available to you.

How do I configure Microsoft SQL Server (MS-SQL)?

MS-SQL Driver

Make sure that the Microsoft SQL server driver file (sqljdbc4.jar) is present in the $tomcat/shared/lib folder. If not, download it from Microsoft's web site and store it there.

MS-SQL configuration

- Download the sample MS-SQL configuration file available at database-mssql-example.hbm.xml .
- Modify the url, username and password as needed.
- Move the file to the OrkWeb installation folder (typically `C:\Program Files\OrkWeb`) as `database.hbm.xml` , replacing the existing file.
- Start (or re-start) the Tomcat service, and verify in the `orkweb.log` file that all started normally.

Important notes:

- When using a database created manually as opposed to by the first start of the orkweb application, it is necessary to apply the database indexes manually, to avoid performance issues down the line. Download create_indexes.sql nd execute it as recommended in the header of the file.
- In the database-mssql-example.hbm.xml file, the recommended hibernate dialect to be used is net.sf.oreka.dialect.SQLServerUnicodeDialect. This dialect supports unicode and helps eliminate some potential performance issues related to queries based on indexed character columns. This dialect is officially available only as of OrkWeb version 1.9-3003. If needed with an earlier version of OrkWeb, please contact <support@orecx.com> .

OrkWeb does not recognize non latin-based characters?!

If you are using MySQL, it may have defaulted to using the latin1 character set at installation. To use UTF8 instead, you need to perform the following steps:

- Configure MySQL to use UTF8 by adding or modifying the following entries to in /etc/my.cnf file (Linux) or my.ini file (Windows) - make sure to keep a backup of the original files:

  ```
  [client]
            default-character-set=utf8
            ...
            [mysql]
            default-character-set=utf8

            [mysqld]
  ```

```
default-character-set=utf8
```

- Restart the MySQL service.
- In MySQL, verify that the character sets are all UTF8 with the following command:

  ```
  SHOW VARIABLES LIKE 'character_set_%';
  ```

- Convert all the oreka database tables to UTF8. You can download the `mysql2utf8.sql` script at http://files.orecx.com/tools/mysql2utf8.sql

  and execute it as follows from the command line:

  ```
  mysql -uroot -p<password> oreka < mysql2utf8.sql
  ```

  Replace <password> with your MySQL root password.

- Edit the `database.hbm.xml` file (in /etc/okrweb on Linux, and typically in c:\program files\orkweb in Windows) and add the following suffix **"?characterEncoding=UTF-8"** to the "hibernate.connection.url" entry as in the example below:

  ```
  <property name="hibernate.connection.url">jdbc:mysql://localhost/oreka?characterEncoding=UTF-8</property>
  ```

- Ensure that Tomcat is configured to run with the -Dfile.encoding=UTF-8 Java option (for the static text).
- Restart the Tomcat service.

How do I configure a scorecard for QM?

Scorecards are often first written in spreadsheet files. Therefore a CSV format is the most natural extension for importing scorecards into OrkWeb. This means that a spreadsheet file may simply be saved in a CSV format and imported as is into OrkWeb, as long as it follows the guidelines and example below.

Scorecard CSV File Format

- A line starting with the # symbol is considered to be a comment.
- A non-comment line must contain comma-separated fields. The number of fields depends on the type of entry. There are 4 types of entries, distinguishable based on the first field:
  - Scorecard, name, comments
  - Group, name (optional)
  - Section, name, weight, comments
  - Question, name, weight, comments
  - Answer, description, value, excluded, autofail, comments
- Fields that are empty may be left blank. If one or more fields at the end of the line are empty, they may be simply omitted. OrkWeb will assign to them default values. Extra trailing fields that should not be there are simply ignored.
- Entries are hierarchical and at least one entry is required at each level, i.e. a scorecard that does not have separate sections must define a section anyway. The only entry that is optional is the Group. If not specified, the scorecard is not associated to any group (not recommended).
- A file may include multiple scorecards.
- About the fields:
  - "name", "description" and "comments" fields are of type String. The string fields must be surrounded by double-quotes, e.g. "Group Name", and should not include any double quotes, commas, etc within them (for now)
  - "weight" fields are decimal values (stored as float). Default weight value: 1.0
  - "value" field is a number (stored as integer).
  - "excluded" and "autofail" are booleans, and take values: true or false, yes or no, 0 or 1. Defaults: false for both.
  - The keywords (Group, Scorcard, Question, Answer) are case-insensitive. They are mandatory at the beginning of the line.
  - name, description and value fields are mandatory. The rest are optional but must be represented (i.e. by a comma separator). See above for the default values for optional fields.

Scorecard CSV File Example

```
# Oreka sample scorecard CSV file: Scorecard_Generic_Example2_Sales.csv
# Below
is a summary description of the fields required:
#
#   Scorecard,name,comments
#   Group,name
#   Section,name,weight,comments
#
Question,name,weight,comments
#   Answer,text,value,excluded,autofail,comments
```

```
#
Scorecard,"Sales QA
Form",
Group,"Sales"
#
Section,"Opening Call",,""
Question,"Professional Greeting - Intro",,""
Answer,"No",0,,,
Answer,"Yes",1,false,false,""
#
Section,"Information
Verification / Data Collection",,""
Question,"Verify Zip Code",,
Answer,"No",0,,,
Answer,"Yes",1,,,
Question,"Collect
Name",1.0,""
Answer,"No",0,,,
Answer,"Yes",2,,,
Question,"Collect Phone Number",1.0,""
Answer,"No",0,,,
Answer,"Yes",2,,,
Question,"Verify
if Existing Customer and if Residence or Business",1.0,""
Answer,"No",0,,,
Answer,"Yes",2,,,
#
Section,"Establishing
Purpose of Call",1.0,""
Question,"Reason for Call",1.0,""
Answer,"No",0,,,
Answer,"Yes",1,,,
Question,"Lead
Source",1.0,""
Answer,"No",0,,,
Answer,"Yes",4,,,
#
...
```

Chapter 6. Glossary

Table of Contents

Glossary

GSM 6.10 Codec

> GSM 6.10 is an audio codec optimized for voice. It is the default storage codec for Oreka (wrapped into a wav file). It is used in the majority of the cellular networks worldwide and has a compression rate of 13 Kbit/s. When wrapped into a wav file it uses roughly 1.6 KByte of disk space per second of recorded audio. This means it's almost ten times more compact than MP3 format at standard compression rate. The advantage of this format is its ubiquity. It is possible to replay it in almost any existing Windows or Linux media player without installing any extra software or codec.

mcf file

> Media Capture File format. It contains raw dumps of voice buffers in their original wire encoding. The file extension is ".mcf". This is an intermediate capture file format used before sessions are transcoded to their final storage format.

Primary pathname

> The "primary" pathname for a media file is the one configured in the recorder's `config.xml` , for example in <AudioOutputPath> for OrkAudio.

Secondary pathname

> The "secondary" pathname for a media file typically defaults to the YYYY/MM/DD/HH format and gets appended to the primary path. It may be configured separately in `config.xml` (e.g. <TapePathNaming> for OrkAudio).

Wildcard character

> A wildcard character can be used to substitute for any other character or characters in a string. The asterisk character (*) substitutes for any zero or more characters. The question mark (?) substitutes for any one character.