



## Design guidance

---

This section contains information intended to help plan for MediaSense deployment.

- [Proactive storage management, page 1](#)
- [Media storage space provisioning, page 2](#)
- [SIP configuration, page 2](#)
- [Codec configuration for phones, page 2](#)
- [Network provisioning, page 2](#)
- [Using scalable queries, page 3](#)
- [Distribute HTTP download requests over time, page 3](#)
- [Alarm monitoring, page 3](#)

## Proactive storage management

MediaSense offers both retention priority and recording priority storage retention modes. Under retention priority, clients are required to manage the space available for recordings because the system will not perform any automatic pruning. Under recording priority, the system will automatically prune old recordings but the pruning operation does not necessarily delete metadata or generated mp4 files that were created using the deprecated `convertSession` API. MediaSense can be configured to automatically clean up these elements or clients can take proactive responsibility for managing their disk space.

When using recording priority and you have not configured MediaSense to automatically delete pruned metadata, the client application must actively delete sessions that have been automatically pruned. Clients may either periodically issue an API request for pruned sessions or they may elect to receive session pruned events and explicitly delete those it no longer needs.

When using retention priority, there is no automatic pruning and the client is fully responsible for guaranteeing that enough disk space is available for new recordings.

Only if the system is configured for recording priority and automatic deletion of pruned recordings is turned on can the client avoid taking part in storage management.

These session management activities are invoked using the MediaSense API. (For more information, see the *MediaSense Developer Guide*.) If pruning activities are going to be performed regularly, schedule them for low usage periods in order to minimize impact on normal operations.

## Media storage space provisioning

At the application level, MediaSense contains two kinds of media storage--each in its own directory location. Recording storage is located in a partition known as **/recordedMedia** and uploaded videos are located in a partition known as **/uploadedMedia**. These are logical partitions that are each made up of 1 to 16 virtual disks at the VM level. The virtual disks are mapped (using VMWare host configuration) to physical disks on the server or on a SAN. The physical disks are configured and managed using a RAID controller.

The physical disks must meet certain speed and throughput requirements that are described in the "Storage" section in the Compatibility matrix that follows. ESXi "thin provisioning" is not supported on any disk.

## SIP configuration

The following guidance applies to CUBE deployments

- In CUBE deployments, use 'SIP early offer' on dial peers that go to MediaSense. This is the default setting. Only Unified Communications Manager implements 'delayed offer' with no option.
- Use 'SIP over TCP' on dial peers or trunks that go to MediaSense.
- Configure 'SIP options ping support' for dial peers or trunks that go to MediaSense (except in single-node deployments). This feature greatly improves failover support for multi-server MediaSense deployments as well as for MediaSense cluster failover.

## Codec configuration for phones

For Unified Communications Manager recording, some of the newer Cisco IP phones support iLBC or iSAC codecs and Unified Communications Manager may negotiate for them. However, since MediaSense does not yet accept these codecs, they must be disabled for recording enabled devices in the Unified Communications Manager service parameter settings.

## Network provisioning

CUBE interfaces that carry RTP media *must* be configured to

- a fixed 1 gigabit speed, or higher
- be fully duplexed
- not rely on auto-negotiation

Sometimes auto-negotiation fails when 100 megabit speeds are available. Even when 100 megabit speeds are properly negotiated, they are not fast enough to handle a heavy call load.

Recording servers like MediaSense receive a lot of network traffic but generate relatively little of their own traffic. The asymmetric nature of such traffic can lead to the expiration of MAC address forwarding table entries on network switches, which may result in the network being flooded. Network administrators must take this possibility into consideration when configuring network switching equipment.

## Using scalable queries

MediaSense offers an API for searching the metadata in a very flexible manner. While many queries will execute with little or no impact on the normal operation of the MediaSense servers, it is possible to formulate queries that have a significant impact. MediaSense limits the number of simultaneous queries it will process but does not consider the relative cost of each individual query. Customers who use the query APIs must read and adhere to the guidelines for writing scalable queries, which can be found in the *MediaSense Developer Guide*.

## Distribute HTTP download requests over time

Some customers will use the HTTP Download facility to create copies of all recordings, using MediaSense more as a temporary location for these files than as a long term archive. Customers may also batch these requests and issue them once a day or on some other periodic basis. It is better, from a resource usage perspective, to distribute these requests more evenly over time. For example, use the session ENDED event to trigger a download as soon as a call recording terminates.

## Alarm monitoring

Various situations that require administrator attention cause alarms to be raised (system conditions). These conditions are observed in the system logs as well as in RTMT's alarms page. Using RTMT, you can configure these alarms to be sent to a SYSLOG server and to send email messages to a designated email address. MediaSense does not currently support SNMP alarms.

At least one of these methods must be used to actively monitor the state of the MediaSense servers.

