Oracle Video Server™

Introducing Oracle Video Server

Release 3.0

February 1998

Part No. A53956-02

ORACLE®

Enabling the Information Age™

Introducing Oracle Video Server

Part No. A53956-02

Release 3.0

Printed in the U.S.A

Primary Authors: Brian Linden, Matt Prather

Contributors: Kim Bartlett, Chris Bedford, Frances Buran, John Dowden, Gordon Furbush, Jean Giarrusso, Dana Hausman Izenson, Lesley Kew, Liz Landreth, Ming Lee, Sue Lee, Alex Lind, Martin McKendrick, Mason Ng, Jeffrey Olkin, Dave Pawson, Mark Porter, Pat Ritto, Daryl Rosen, Andrew Samuels, Shawn Shyh, James Steel, Sean Trabosh, Daniel Weaver, John Zussman

Contents

Figures

Send Us Your Comments

Introducing Oracle Video Server, Release 3.0

Part No. A53956-02

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- electronic mail - omsdoc@us.oracle.com
- FAX - 650-506-7615. Attn: Oracle Video Server Documentation Manager
- postal service:
  Oracle Video Server Documentation Manager
  Oracle Corporation
  500 Oracle Parkway, Box 6OP5
  Redwood Shores, CA 94065
  U.S.A.

If you would like a reply, please give your name, address, and telephone number below.

Preface

Introducing Oracle Video Server provides basic conceptual information about the Oracle Video Server system. This tour is not a technical explanation of how the software works, but an overview of the basic concepts behind the Oracle Video Server system.

Read this book before you:

- install the Oracle Video Server as described in your *Oracle Video Server Installation Guide*, or
- start and operate it as described in your *Oracle Video Server Administrator's Guide and Command Reference*

The Oracle Video Server system includes these Oracle software products:

- Oracle Video Server
- Oracle Media Net
- Oracle Video Client
- Oracle Video Server Manager

This preface discusses:

- this document’s intended audience
- the document’s organization
- other documents related to the Oracle Video Server system

Audience and Organization of This Document

This document is divided into two chapters, intended for slightly different audiences:

The first chapter presents a general conceptual overview and discussion of video servers and the benefits they can offer you, as well as some of the advantages afforded by Oracle's unique video server implementation. This part is intended for anyone who would like to learn more about the Oracle Video Server and its capabilities.

The second chapter presents Oracle Video Server architectural details, and is intended for OVS system administrators and multimedia application developers.

Related Documents

See the *Oracle Video Server Road Map* for a list of related documents.

Conventions

In examples, an implied carriage return occurs at the end of each line, unless otherwise noted. You must press the *Return* key at the end of a line of input.

The following conventions are also used in this document:

Convention	Meaning
⋮	Vertical ellipsis points in an example mean that information not directly related to the example has been omitted.
. . .	Horizontal ellipsis points in statements or commands mean that parts of the statement or command not directly related to the example have been omitted
boldface text	Boldface type in text indicates a term defined in the text, the glossary, or in both locations.
italicized text	Italicized text indicates emphasis or a document title.
< >	Angle brackets enclose user-supplied names.
[ ]	Brackets enclose optional clauses from which you can choose one or none.
$	The dollar sign represents the DIGITAL CommandLanguage prompt in OpenVMS and the Bourne shell prompt in Digital UNIX

Your Comments Are Welcome

We value and appreciate your comments as an Oracle user and reader of the manuals. As we write, revise, and evaluate our documentation, your opinions are the most important input we receive. Our Send Us Your Comments form is at the front of the manual, at the end of the table of contents. We encourage you to use this form to tell us what you like and dislike about this manual or other Oracle manuals. If the form is not available, please use the following address or FAX number.

Oracle Video Server Documentation Manager
Oracle Corporation
500 Oracle Parkway
Box 6OP5
Redwood Shores, CA 94065
U.S.A.
FAX: 650-506-7615
E-mail: omsdoc@us.oracle.com

1

Concepts

This chapter is intended to introduce you to streamed digital video and video servers in general, as well as the advantages offered by Oracle Corporation's unique implementation of video server technology.

While this chapter deals primarily with higher-level conceptual information, it also presents enough details of video server architecture to understand the technology and its capabilities.

What Is Streamed Digital Video?

If you have purchased a digital versatile disk (DVD) of a favorite movie, you're already familiar with digital storage and delivery of multimedia content. When the producers record the laser disc, the information is interpreted and encoded digitally, then this digital representation is recorded onto the disc. When you play the laser disc, this digital representation is then decoded, and the result is the video display that you see.

Streamed digital video works in much the same way as the digital video you've already used: the video is initially recorded in either traditional analog (unbroken, continuously varying representation) formats or directly into a digital format. If the original video is in a traditional format such as videotape, it is then encoded into a digital format that can be stored on disks and decoded for playback.

Streamed video, however, is fed from a server computer with large storage and delivery capabilities to a client machine that decodes and displays the streamed video as it arrives. This eliminates the need for the video to physically reside either on a playback medium (such as the laser disc) or locally on the machine that displays the video.

How Is Streamed Video Better than Traditional Media?

Streamed video can present several advantages over such traditional media as film and videotape: like the video on a laser disc, streamed digital video does not degrade or “wear out” from repeated usage. It remains in original condition until it is explicitly deleted, modified, or overwritten.

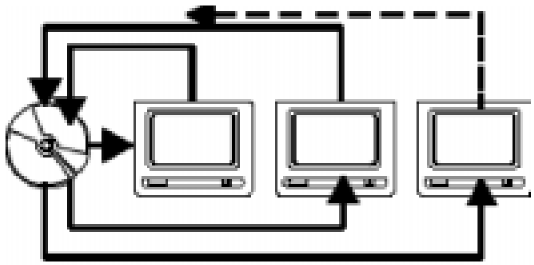
Streaming video over a network also affords a high degree of portability. Unlike the laser disc, streamed digital video does not require that you physically move the *storage* medium with the *display* mechanism. The client machine need only have network access to the server machine where the videos are stored.

Digital video is also available as downloaded files, but these files present many of the same disadvantages as traditional storage media:

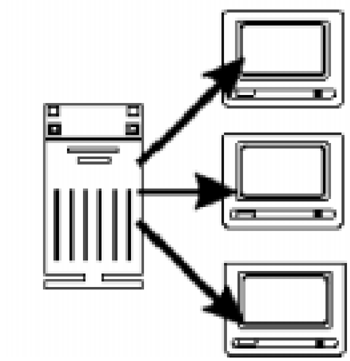
- You must download the *entire* file before you can view *any* of the file. In contrast, you can begin viewing streamed video as soon as the beginning of the stream reaches the client machine.
- The local machine must provide enough storage space to contain the entire downloaded video file. Streamed video requires that the client machine provide only enough capacity to store and display the video portion being delivered and viewed (and perhaps a small buffer to provide error-recovery capability).

More than this, though, streamed digital video is **scalable**. Many people can view the same video—not just multiple copies of the same movie title—at very nearly the same time. The difference in these delivery mechanisms is illustrated in Figure 1–1.

Figure 1–1 Traditional digital video vs. streamed digital video



Traditional digital-video media such as laser discs operate in a "spiral" fashion. The medium supports only one viewer at a time; use then "spirals" to the next viewer, back to disc storage, to the third viewer, and so on. Each viewer has the *entire* video during viewing, and it is unavailable to any other viewer.



Streamed digital-video media delivers the video as a series of small pieces. As soon one small piece of video has been sent to one viewer, that piece is available to be sent to another. Thus, streamed digital video can simultaneously serve many viewers.

This important feature of streamed digital video and video server systems is called **video-on-demand**, and it means that you can configure the video server to enable your clients to watch what they want to when they want to, choosing any title that has been digitized and stored for delivery.

Another way to make streamed digital video available to clients is to schedule it. An administrator can schedule delivery of a specific video at a specific time on a specific channel and customers can tune in and watch it. Near video-on-demand (**NVOD**) is one application of scheduling in which the administrator schedules delivery of a video to begin automatically every few minutes on a different channel. For example, you might schedule "Marvin's Room" to begin at 8:00 p.m., and every 15 minutes thereafter, until midnight. If customers miss the 8:00 show or have to pause in the middle of the movie, they can wait just a few minutes, and tune to the appropriate channel to see or resume the movie.

Video-on-demand and scheduled video are possible because the digital data itself is stored as a series of **stripes**, typically of 32k or 64k each, with the stripes distributed over multiple disks. Depending on the encoded bit rate, a single stripe on a disk might represent several seconds of video time. As soon as the information in the stripe has been sent to one client machine, the stripe can be sent to another, so many different parts of the file can be read simultaneously, and a single copy of a video can serve many users concurrently.

Striping also enables a sophisticated backup-and-recovery system known as **RAID (Redundant Arrays of Inexpensive Disks) Protection**. With RAID, video can still be delivered in the event of a disk failure. RAID only needs to recover a single stripe, rather than an entire video, so the server doesn't need to duplicate storage space for each video, and that means that more space can be devoted to the videos themselves.

You can learn more about striping and RAID in Chapter 2.

What Is a Digital Video Server?

Digital-video files can be very large. A 2-hour video in an uncompressed format may require several hundred gigabytes of storage space. So before files are stored, they are compressed. The compressed file size depends on the **codec** (compression / decompression software) and the compression rate you choose, however the same video encoded in the **MPEG (Motion Pictures Experts Group)** format at 1.5 Mbps still requires about 2 gigabytes.

Because the amount of digital information is so large, digital video generally demands large storage and delivery capacity. Several movie titles or customer presentations will require multiple gigabytes of storage. For this reason, it's not practical to store these videos in the same places they'll be viewed, such as:

- desktop computers or laptop computers connected to a network
- NC (network computing) devices that rely on a network connection and do not include their own storage space, or
- traditional television sets with a set-top box to enable viewing of digital video

This is where a **video server** comes in. A server machine is designed to offer the storage and throughput capacities needed for timely and reliable delivery of digital video. In turn, this means that you need not find storage for every video you want to view on your desktop computer --- which would leave little room for anything else!

What is the Oracle Video Server?

The Oracle Video Server (OVS) system is a unique implementation of video server technology for networked computers which store, manage, deliver, and display digital multimedia data (real-time, full-screen video and high-fidelity audio) on demand. OVS is supported on a variety of server platforms and can scale to serve many thousand concurrent users.

This section is not intended to present a detailed explanation of the OVS functionality, it provides a basic understanding of how the pieces of this system fit together and relate to one another.

OVS within the Network Computing Architecture (NCA)

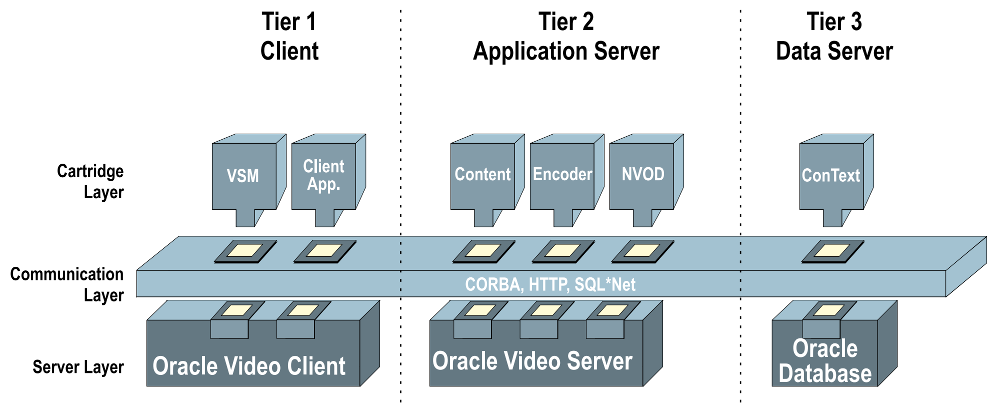
The OVS system is fully compliant with Oracle's Network Computing Architecture (NCA). NCA affords multiple computing advantages, including a "thin-client" architecture that requires minimal data storage space on the client machine.

Similarly, the Network Computer (NC) client presents high levels of security and error recovery; users need only use any NC client machine to log onto the NC server. They can then restore access to any applications and data that may have been lost because of power outage, flood, or any other damage to the client machine.

Because of its thin-client nature, an NC client machine is also very affordable to purchase, deploy, and maintain.

Figure 1–2 shows OVS in the NC environment, a scalable, object-oriented architecture, which consists of three tiers and three layers.

Figure 1–2 Oracle Network Computing Architecture



Tier 1: Client

There are two client components provided as part of the OVS system:

Oracle Video Client: The Oracle Video Client (OVC) software allows you to develop interactive, video-based multimedia applications for such uses as computer-based training (CBT), interactive kiosks, corporate repositories, and Web sites. The OVC software provides several tools to help you build and view client video applications. For more information about the OVC, see the *Oracle Video Client Developer's Guide*.

Oracle Video Server Manager: The Oracle Video Server Manager (VSM) is a Java application that provides management-console client machines with point-and-click control over OVS services, clients, and content. (VSM need not be installed on every client machine.) For more information about the VSM, see *Beyond the Basics* on page 1–20 of this document, and *Getting Started with Oracle Video Server Manager*.

Tier 2: Application Server

The Application Server tier consists of the Oracle Video Server (OVS). The OVS functions as the application server in this environment, receiving and processing requests for digital video, then delivering the digital video content to the client device.

Tier 3: Data Server

The Data Server tier contains the target services which the client needs to access to get application-specific data. In the OVS system environment, the Data Server tier consists of the Oracle database (optional). A database is a reliable repository for persistent storage of structured data. The OVS system can use the Oracle database to write and query database tables associated with OVS processes, including logical content, clips, and schedules.

Server Layer

The Server Layer provides the basic functionality for each tier.

Cartridge Layer

The Cartridge Layer provides programmatic functionality for the Server Layer of each tier.

Communication Layer

The Communication Layer provides communication between servers, cartridges, and tiers.

What is a Digital Video Client?

If digital video is so large that it must be stored on, and delivered from, a video server, how do you actually obtain and view the video information in the stored digital video? The **video client** machine is responsible for obtaining, decoding, and displaying the video stored on the server.

Most commonly, the client machine requests the video from the server. In a video-on-demand configuration, video delivery is always a result of requests from the client; the server does not send digital video content until specifically requested to do so. The video client then uses either software or hardware to decode the digital video it requests from the server, and displays that video for the viewer. (In a scheduled video system, the client does not explicitly request video delivery, since the delivery schedule has already been established.)

The client does not store the video it displays. Storage and delivery are left to the server machines, which are large and fast enough to handle the demands of delivering large amounts of video data. The client requests, receives, and displays the digital video but does not retain that data—once the digital video data has been displayed, it no longer exists on the client machine. This means that the client machine needs only the relatively small amount of system resources necessary to store and operate the client software, not the digital video itself.

Customizing the Video Client

The Oracle Video Client can do more than simply request and display digital video. Its true power lies in its ability to act as a platform that enables you to develop completely customized video applications for use in computer-based training, information kiosks, Web sites, and corporate information repositories.

The OVC software comprises a variety of components, each extensible and customizable, including:

- Java native classes and a Java player applet, to afford complete consistency and portability across a variety of operating platforms, such as Microsoft Windows and various “flavors” of the UNIX operating system. Use of a Java-based video client also eliminates the need for an installed HTML browser on the target platform. Additionally, a Java-based client implementation enables smooth integration and use on Java-based devices such as NCs and set-top boxes.
- Web Plug-in, for use under a browser in HTML pages.
- ActiveX control, to integrate the video client into applications designed to operate in the Microsoft Windows 95 or Windows NT operating systems.

Developers can then deploy the base video client on various machines, using these “extensions” as the means of integrating the client’s functionality.

What Are the Challenges of Digital Video?

Broadly speaking, the challenge of streamed of digital video is to

1. retrieve large encoded content files from storage,
2. stream these files at high delivery rates across a network that may be unreliable, and then
3. reassemble, decode and display the streamed content on a client device.

Although the particular challenges of implementing digital video will vary depending on the environment (broadband, enterprise, or Internet), some of these challenges are common to all environments. For example, latencies in delivering data over a network might be acceptable or even unnoticeable in applications such as word processors or spreadsheets, which are not **isochronous** or time-sensitive. Such latencies are extremely noticeable in time-dependent applications such as video, and show up in the form of **glitches** such as:

- misaligned or non-corresponding parts of the video screen
- some (or all) parts of the screen stop moving

- audio and video are not synchronized
- the screen is blank for short instances of time

The Oracle Video Server system anticipates these challenges and provides effective high-level tools to address them. For example, OVS uses several means of ensuring that your digital video delivery is reliable and won't be interrupted in the event of a failed disk or a bad video stripe. In addition to RAID (Redundant Arrays of Inexpensive Disks) Protection, OVS offers the ability to create **Spare Disks** (physical storage space into which you can temporarily transfer content in the event of a disk failure) and use **hot-swapping** (a technique to transfer digital-video data to a spare disk without bringing down the video server system or losing video service).

How Is Digital Video Used?

You can use Oracle Video Server to incorporate digital video into a variety of applications, such as interactive training-on-demand, product announcements, CEO messages, point-of-sale kiosks, web sites, corporate repositories, pay-per-view television and multimedia catalogs. These multimedia applications can provide your company with competitive advantages which dramatically improve productivity while reducing costs.

This section describes some of the interesting ways in which you can use OVS.

Clips and Logical Content

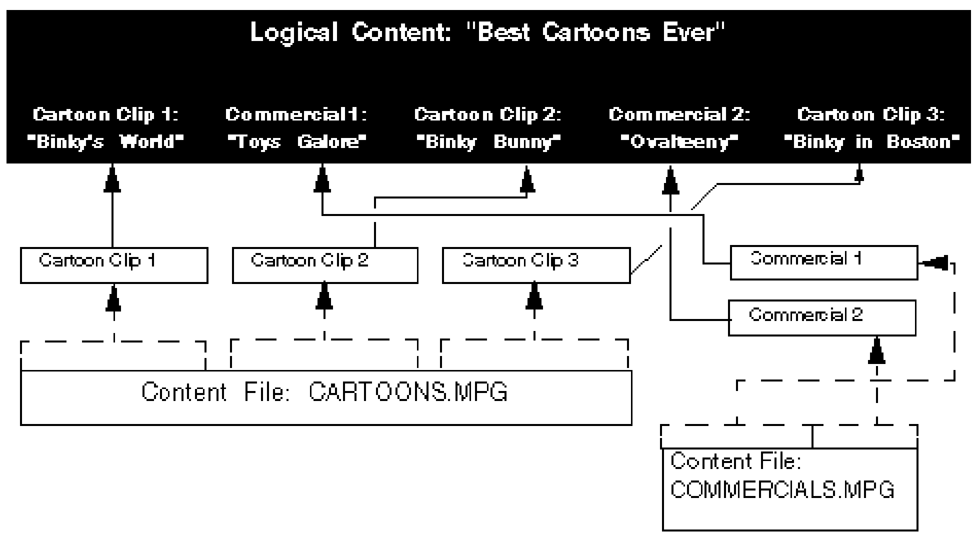
When commercial video subscribers request a "feature" video, the video they actually see will probably also contain advertisements for available products and services. There may also be previews of other movies, concerts, and events soon to be available on digital video.

Each of these short segments is known as a **clip**. The OVS enables system administrators to assemble these clips as they see fit, then label the assembled clips and videos with a single title, to make retrieval and delivery easier. Because these clips do not actually need to *physically* be assembled into a single file, but are accessed from their physical locations by OVS's internal logic, they are referred to collectively as **Logical Content**. Much as beads can be strung together in many different combinations to create a single unified piece of jewelry, logical content clips are "strung together" to create a single logical content title, or a single unified stream of digital video.

This concept is also the way you construct a playlist on your home CD player. For example, if you have three CDs loaded into your player, you may find that you like the first four tracks of the first disc (but not the fifth track), all of the tracks on the second disc, and only the eighth track of the third disc. You can program your player to play only these tracks, in any sequence you choose, and give this assemblage a single name to be stored in the player's memory for easy retrieval. Logical content within digital video files works in nearly the same way, except that a database is required to track all of the many possible combinations of video clips.

Figure 1–3 shows one possible way in which logical content might be used to assemble a children's cartoon show; the basic process will work the same way when any video subscriber chooses a video using logical content capabilities.

Figure 1–3 Logical content assembled to create a single cartoon show



You can also write OVS applications that define logical content titles dynamically. These applications can customize logical content based on users' interests and viewing history. For example, the application could program the advertisements that show during a movie based on the interests of the viewers.

You can learn more about logical content, clips, and the differences between the video server computer (where OVS resides) and the database server computer (where an Oracle database stores logical content definitions) in Chapter 2.

Real-Time Feeds

Some commercial video providers also offer **Real-Time Feeds**. In this situation, special encoders are used to record, encode, and begin streaming digital video of an event as that event occurs—in so-called "real time." News channels, stock updates, sports, corporate events, or other services that depend on up-to-the-minute accuracy and timeliness of information can use **One-Step Encoding** (digital video is encoded as it is recorded) and **Continuous Real-Time Feeds** to ensure their customers get the latest information via digital video, because the viewer sees the event nearly as it happens, the only delay being the minimal time necessary to encode and deliver each moment of the event. Viewers can also pause, restart, rewind, and fast forward the video up to the current "live" point.

Where is Digital Video Used?

Digital video can be used in many environments. This section describes some example, divided into the broad categories broadband, enterprise, and Internet.

Broadband

The broadband environment is characterized by high-speed delivery of complex video data, such as would be required by full-screen, full-motion video. This data need not necessarily represent long videos (for example, a home-shopping segment might be 2-3 minutes or less), but will include a great deal of digital-video information.

Typically, this environment requires the use of a dedicated **set-top box**, a special receiving and decoding device attached to a display such as a television monitor. Examples of broadband usage would include:

- Subscription-service delivery of a digitally recorded event such as a blacked-out athletic game
- Movies, concerts, Broadway shows, or other events delivered on demand
- News on demand
- Product and service directories
- Distance learning

- Home shopping
- Home financial services
- Medical diagnostics

Asynchronous Transfer Mode (ATM)

Broadband deployment of streamed digital video often uses the **Asynchronous Transfer Mode** (ATM) networking protocol for video delivery. ATM provides a reliable mechanism for high-speed delivery of large amounts of highly complex data, using fixed-size delivery packets called **cells**. Because of its switched, connection-oriented nature and its fixed cell size of 53 bytes, ATM is an inherently reliable, scalable networking technology.

ATM can offer several advantages over other networking protocols, including:

- scalability, to work at different speeds and on different media
- open-ended growth paths, since ATM is not tied to any physical medium or speed
- a single network for delivery of voice, data, and video, thereby improving efficiency and manageability
- compatibility with existing physical networks (because ATM is not dependent on a specific type of physical transport)

In some cases the ATM network is used only for delivery of the video data itself, and another network is used to deliver system messages between the client and server. The second network often has less bandwidth than the video network; for example, using **Internet Protocol** (IP) over a telephone line.

Digital Subscriber Line (DSL) Transport Systems

One emerging way of delivering video is through **DSL** (Digital Subscriber Line) transport systems. DSL was originally intended by telephone companies as a way to deliver both television and telephone service over existing copper telephone lines. Using the same copper wires on which much existing telephone service is based, DSL can provide data access and downloading up to 50 times faster than a 28.8 Kbps modem.

DSL is actually a continuum of a variety of transport systems that can carry about 1 to 6 Mbps. In general, the faster the DSL, the shorter the distance it can reliably cover. The fastest DSLs can cover only a few miles; the slowest can cover farther.

DSLs include:

- **ADSL** (Asymmetric DSL) — called “asymmetric” because the connection from service provider to client delivers large amounts of data at high rates, but the connection from client to service provider operates at a much lower rate.
- **HDSL** (High-bitrate DSL) — currently the only widely deployed DSL and essentially a replacement for traditional T1 service. T1 lines carry 1.5 Mbps and have been available for many years, but require technicians to tune them to optimum performance. HDSL modems can handle marginal connections with little problem, so they are often much less expensive to install and operate.

Telephone companies are developing new DSLs for future use.

Enterprise

In an enterprise environment, needs are geared toward the particular company (or “enterprise”) implementing the video solution. Content may still be as large as in broadband environments, but viewers are served over a corporate LAN (using **switched Ethernet** networking), WAN, or intranet.

Switched Ethernet, like ATM, is a highly scalable networking protocol capable of delivering very large amounts of data at high speeds. Additionally, an Ethernet network can, when needed, be migrated to an ATM network topography.

Switched Ethernet, as opposed to unswitched or **shared Ethernet**, helps prevent “bottlenecks” or slow-downs in network operation caused by multiple users needing to read every data packet to determine which packets apply to which clients. Likewise, switched Ethernet is a **full-duplex protocol**, meaning that each hub can simultaneously send and receive, unlike other **half-duplex protocols** that do only one or the other at one time. Switched Ethernet allocates and reserves necessary bandwidth for each user connecting to the network, so latencies in delivery of data over the network are minimized or eliminated.

Examples of enterprise usage of Oracle Video Server, over switched Ethernet, would include:

- Delivery of corporate training videos at a time convenient to the employee or least disruptive to work schedules, rather than as a single scheduled event with many employees in a single room at a time.
- Corporate announcements from upper management or the Board of Directors.

- Information stored in a “corporate repository” and available for retrieval as needed—for example, detailed step-by-step instructions for enrolling in your company’s 401(k) program, instructions for installing a piece of equipment, or an introductory tour of the new cafeteria.

Internet

The Internet environment is typically oriented toward delivery of digital video over public Internet pages. Most people still connect to the Internet via modem, and this environment is often referred to as “low-bitrate” because of the constraints imposed by modems with a relatively slow rate of delivery—a “low bit rate.” Although many newer modems boast speeds that seem very fast in comparison to older modems, they are still extremely slow compared to the delivery rates of a broadband environment, and this slower delivery rate must be taken into account when planning digital-video delivery for this environment.

Internet users who connect via a modem will connect with their **Internet Service Provider** (ISP), who provides access to the OVS through the Internet. Through a single connection, the client communicates with the ISP using PPP (Point-to-Point Protocol) and with the OVS using either **TCP/IP** or **UDP/IP** (Transmission Control Protocol or User Datagram Protocol/Internet Protocol).

Because of its increasing availability and affordability, DSL has also become a popular way to access the Internet. Although high-bitrate transports like DSL are avialable, as the administrator of a video-based Web site, you should plan for the slowest transport system that is likely to access your site, rather than relying on your clients to have a faster one.

There are many innovative uses for Internet-based delivery of video, including:

- Advertising. Many commercial locations now have Web sites that feature informative digital video to acquaint you with their product.
- News. You can subscribe to news-delivery Web sites that offer not only still photos (as in an “online newspaper”) but full-motion digital video of breaking stories.
- Site tours. Some sites, especially those for large academic institutions, are large and complex enough that they offer video orientation to the Internet site, the institution itself, or both.

Putting It Together: The Video Round Trip

So how does a request from the client machine for digital video result in that video being delivered and displayed to the viewer? This section illustrates that process, known as the **video round trip**, by showing typical configurations in each of three video environments: broadband, enterprise, and Internet.

You can learn more about streamed-video round trips in the online *Oracle Video Server Manager Quick Tour.*

All Environments

At its most fundamental, a streamed-video round trip functions as shown in Figure 1–4 and described as follows:

1. Client Tier The video client sends necessary configuration information (and, in a VOD situation, the request for video).

The client device can be:

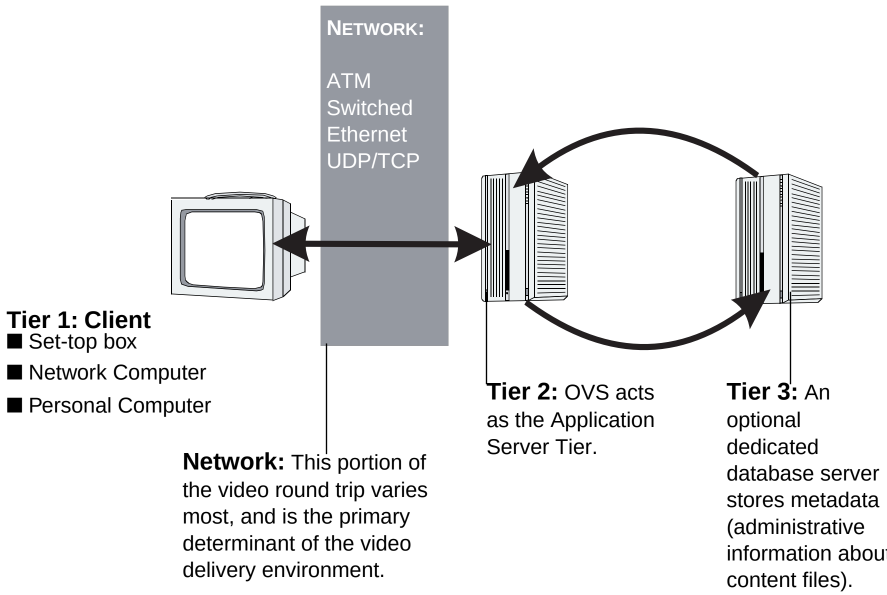
- A set-top box connected to a television set or monitor and to a public or a private network. This is a typical client device for use in a broadband delivery environment.
- An NC (Network Computer) connected to a network. Through the network, the Network Computer accesses an NC server and the OVS.
- A PC (Personal Computer) connected to a network.

2. Network The network receives, processes, and forwards information between the client device and OVS. This portion of the video round trip is the part that primarily defines the environment as broadband, enterprise, or Internet.

3. Application Server Tier Oracle Video Server receives video requests, processes them, and (if necessary) communicates with the video-database server to obtain digital-video titles and files.

4. Data Server Tier (Optional) The video database maintains metadata concerning the actual digital video files, accesses and delivers video-file information to OVS.

Figure 1–4 A basic video round trip

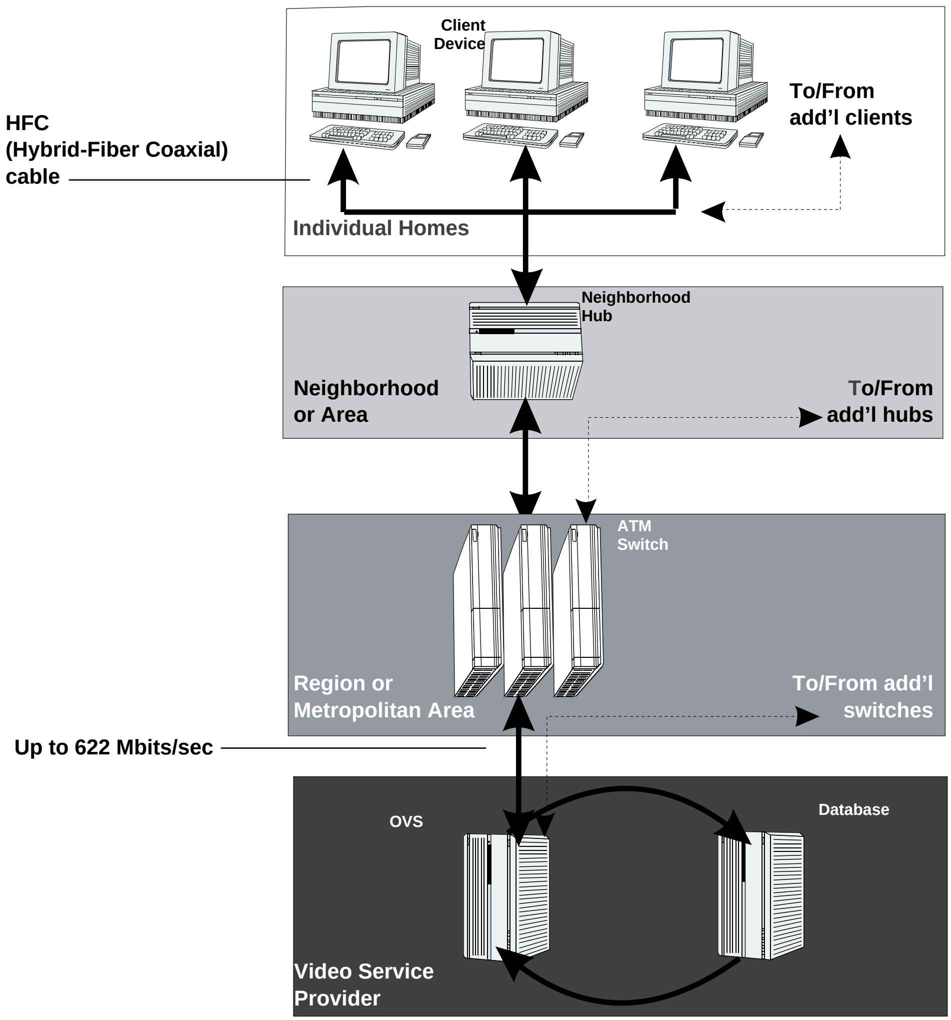


Scenario 1: Broadband Environment

The broadband environment is characterized by delivery of large amounts of digital video data at very high rates to dedicated set-top devices that receive and decode the digital video. Video is typically shown full-screen and full-motion.

Figure 1–5 shows an example of a network configuration using Asynchronous Transfer Mode (ATM) in a broadband delivery environment. The network itself has been somewhat simplified, to show a typical configuration; actual broadband deployment might involve many more variables in terms of particular hardware used, the number of neighborhood hubs, and the type and number of ATM multiplexers.

Figure 1–5 Typical ATM network for the broadband delivery environment



1. In an ATM environment, the neighborhood hub receives messages from, and delivers video to, the client device. The hub has the IP address of the client, to ensure proper delivery.

2. The neighborhood interfaces with an ATM switch serving an area or metropolitan region (several hubs).

3. The switch forwards client messages to OVS (which communicates with the database server), and delivers streamed video coming from the server.

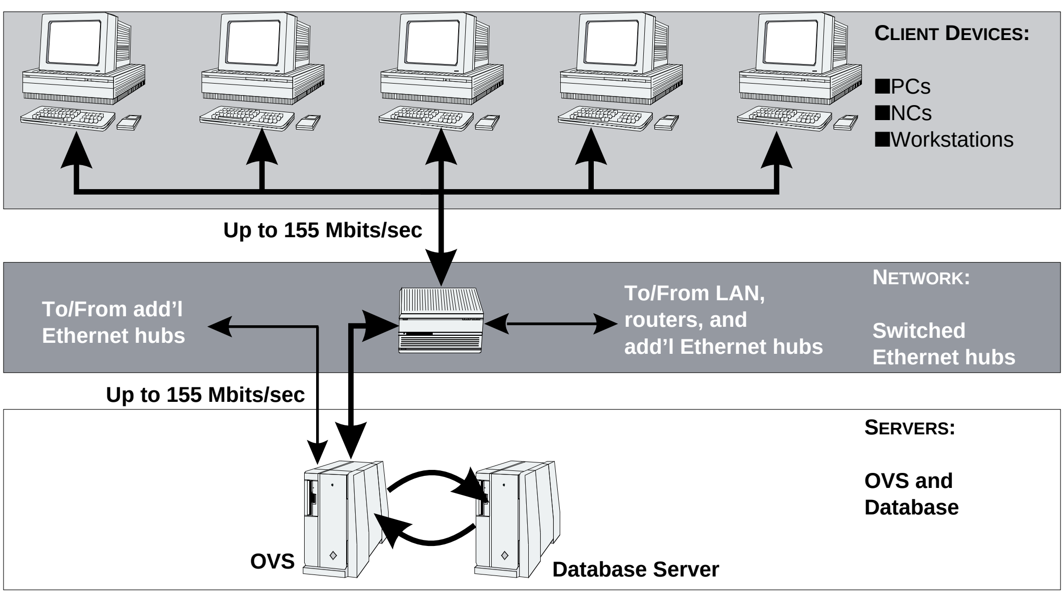
Scenario 2: Enterprise Environment

In the enterprise environment, the processes of requesting titles and delivering digital video are much the same as in the broadband environment (for details, see *Scenario 1: Broadband Environment* on page 1–16), except that:

- An enterprise environment does not typically schedule video broadcast. Instead, it will usually make digital videos available as requested from client machines.
- The network communication layer, rather than consisting of a dedicated high-speed broadband network, will instead consist of a corporate-wide **LAN** (Local Area Network) or **WAN** (Wide Area Network). This layer might use such communication protocols as Switched Ethernet, CDDI, or FDDI (Copper or Fiber Data Distributed Interface).

  A Switched Ethernet topology provides optimal throughput and reliability in this environment. Figure 1–6 shows a typical configuration for the Switched Ethernet network used for video delivery in an enterprise environment. See *Enterprise* on page 1–13 for more information on using Switched Ethernet in an OVS system.

Figure 1–6 An enterprise (Switched Ethernet) OVS system



- No "membership subscription" is required to the corporate network. The client device is a computer which is also used for other purposes, rather than a set-top box. The delivery network is also not solely dedicated to video.
- The client application used for video display might be deployed as part of another application such as an e-mail or groupware package. For example, you might choose to deploy a video client as an ActiveX component in a Windows-based application you are developing, to provide "one-on-one" video training to the application's new users. Or, you might use the Web plug-in client to enable viewers to browse corporate intranet pages and view the videos those pages reference.
- Depending on the bandwidth available, video may be shown full-screen or in a window on the PC, and at various frame rates.
- Logical content is available (assuming a database is used to store such content), and the nature of it is tailored to corporate needs. For example, a digital video introducing a new product might begin with a clip of the company CEO giving a short speech. For a description of the capabilities of logical content, see *Clips and Logical Content* on page 1–9.

Scenario 3: Internet Environment

In the Internet environment, the process of delivering digital video is much the same as in the broadband environment (for details, see *Scenario 1: Broadband Environment* on page 1–16), except that:

- The network communication layer, rather than consisting of a dedicated high-speed broadband network, will instead consist of **PSTN** (Public Switched Telephone Network) over the publicly accessible Internet, or another protocol if the Internet is accessed from within a corporate-specific intranet. Over the Internet, this layer uses such communication protocols as **TCP** (Transmission Control Protocol) or **UDP** (User Datagram Protocol).
- The client device is a PC, NC, or set-top box running an HTML browser or applications written in the Java programming language.
- The client application used for video display might be deployed as a **plug-in** (an extension) on the associated **HTML** (Hypertext Markup Language, the basic language of Internet pages) page, or as a Java application or applet.
- Video is generally viewed in a small window with quality reflecting the bandwidth of the connection.

You can learn more about designing video-delivery systems for the Internet environment in the *Oracle Video Server's Administrator's Guide and Command Reference*.

Beyond the Basics

Maintaining an OVS System

The OVS provides powerful and flexible tools for maintaining the video-delivery system. These tools include:

- Oracle Video Server Manager (VSM). VSM is a Java-based management tool that provides a graphical means of viewing and managing your OVS system. It enables you to perform a variety of system-maintenance and content-management tasks, including:
  - Starting and stopping the OVS
  - Adding, removing, and updating content files within your digital-video storage
  - Checking and editing the status of clips making up your various logical content titles
  - Registering new or existing content files, so that OVS recognizes them and allocates necessary resources for storing and delivering them
  - Defragmenting MDS volumes to minimize unusable empty disk space

  To learn more about VSM, begin with *Getting Started with Oracle Video Server Manager*, then continue with the online *Oracle Video Server Manager Quick Tour* and Help system.
- Command-line utilities that enhance and extend VSM capabilities. Some tasks require extensive personalization or customization, and OVS includes a library of command-line utilities to perform tasks that lie outside the usual domain of VSM.

Creating Applications

One of the greatest strengths of OVS is its extensibility; you can create custom-built versions of both the server and the client portions of this digital-video system.

The Oracle Video Client (OVC) is designed with customization in mind. While this portion of the OVS system serves basic digital video display needs out of the box, it also offers the ability to customize and create a wide variety of tailored client applications. The video client is available, and provides for creation of custom applications, in an assortment of platforms, including:

- Platform-native versions for Microsoft Windows 95 and Microsoft Windows NT
- Java- and JavaScript-based versions for deployment on a variety of operating systems
- a plug-in for use in Web pages

You can learn more about the OVC and its extensibility in the *Oracle Video Client Developer's Guide*.

The server is designed to provide an out-of-the-box solution to your delivery needs for digital video. However, you can tailor it to exactly suit your needs, by using the rich application programming interfaces (APIs) included with the OVS software. For example, you could build a client application for your set-top box or a real-time encoder. To learn more about using the OVS APIs, see the *Oracle Video Server Developer's Guide*.

2

System Architecture

This chapter is an overview of the architecture used in the Oracle Video Server (OVS) system. The information in this chapter is intended to expand on the basic digital-video overview provided in Chapter 1, "Concepts". If you are not familiar with basic concepts of streamed digital video, read Chapter 1 to learn about digital video, video servers, and the OVS system, before reading this chapter to gain in-depth understanding of the system architecture.

The chapter describes these Oracle components that are used in the OVS system:

- Oracle Video Server (OVS)
- Oracle Video Server Manager (VSM)
- Oracle Media Net
- Oracle Video Client (OVC)

It also describes:

- OVS System Hardware and Network Architecture
- How the OVS System Communicates
- Components of Oracle Video Server and Oracle Media Net
- Networking in the OVS System

Oracle Video Server (OVS)

The Oracle Video Server (OVS) is a scalable video software engine that stores video on a video server computer and can deliver the video to multiple concurrent clients in real time. This chapter uses the term *video* to refer to any data streamed in real time, including digitized video and/or audio.

This section describes these OVS features and benefits:

- Content
- Storage
- Delivery
- Network

Content

The video that the OVS stores and delivers to clients is called **content**. The OVS stores content in these forms:

- Physical Content — the actual physical files containing multimedia data
- Logical Content — logical representations of the physical data that you can define to customize the multimedia data your clients can request

Physical Content

Physical video content files are created from actual video through a process called **encoding** performed by a third-party encoding software or hardware called a codec that is not part of the OVS. To create content, the encoder uses these formats:

- The way in which an encoder compresses video frames so they use less space is called a **compression format** or a **codec**.
- The way in which an encoder multiplexes, or puts together, video and audio is called a **container** or **mux format**.

Compression Formats The OVS can stream many compression formats including:

- MPEG (Motion Pictures Experts Group)
- Iterated Systems ClearVideo
- Radius CinePak
- Intel Indeo
- Motion JPEG (Joint Photographic Experts Group)

MPEG (Motion Pictures Experts Group) The OVS supports these compression formats defined by MPEG:

- MPEG-1 video
- MPEG-2 video
- MPEG audio

MPEG formats provide full-motion (up to 60 frames per second), full-screen video with high fidelity / stereo audio playback.

By taking advantage of redundancies among frames in a video sequence, MPEG formats provide a storage compression ratio of 100 to 1 over storing each frame individually, reducing the amount of disk space required to store video.

Container (Mux) Formats The OVS supports these container, or multipexed (mux), formats:

- MPEG-1 System
- MPEG-2 Transport
- RKF (Raw Key Frame) Container Formats such as:
  - AVI
  - WAV

MPEG-1 System and MPEG-2 Transport Table 2–1 shows combinations of MPEG codecs and container formats in which the OVS can deliver video.

Table 2–1 MPEG Codecs and container formats

		Container Formats	
		MPEG-1 System	**MPEG-2 Transport**
Codecs	**MPEG-1**	✓	✓
	MPEG-2		✓

RKF (Raw Key Frame) Container Formats The OVS can also deliver video files in any container format meeting the **raw keyframe** criteria:

- **stateless** — all the data for displaying the picture in a video frame is contained entirely in that frame, rather than in any previous frames
- **contiguous** — all the data for a frame is stored together in the video file and no other data mixed with it

OVS supports delivering files in the RKF formats AVI (Audio Visual Interleave) and WAV (Wavefore audio-only) to Oracle Video Client by converting them to OSF (Oracle Streaming Format). OSF is an RKF format designed by Oracle Corporation for efficient delivery of content that has not been optimized for streaming.

Tag Files The OVS creates and stores a tag file associated with each content file. Tag files contain **metadata**, information the OVS needs to perform rate control sucha as pausing and blind seeking and (on supported clients) visual fast forward and rewind. If a client application requests a rate control operation on a content file, the OVS reads the associated tag file to determine which parts of the content file to deliver to the client.

When you use Oracle Video Server Manager (VSM) to load a content file into the OVS, VSM automatically creates a tag file for the content file. The OVS also provides utilities to:

- create tag files for MPEG content files
- convert AVI and WAV files to OSF and create tag files for them
- create empty tag files for playing video without rate control in formats other than MPEG, OSF, and full raw keyframe formats

The OVS also provides a utility to register existing tag files.

To support rate control for tag files in another compression format that meets the full raw keyframe criteria, you must provide a tagging utility for the format.

VSM and the provided utilities also register the tag files in the table of contents of the OVS and in an Oracle database in your OVS system (if you are using one).

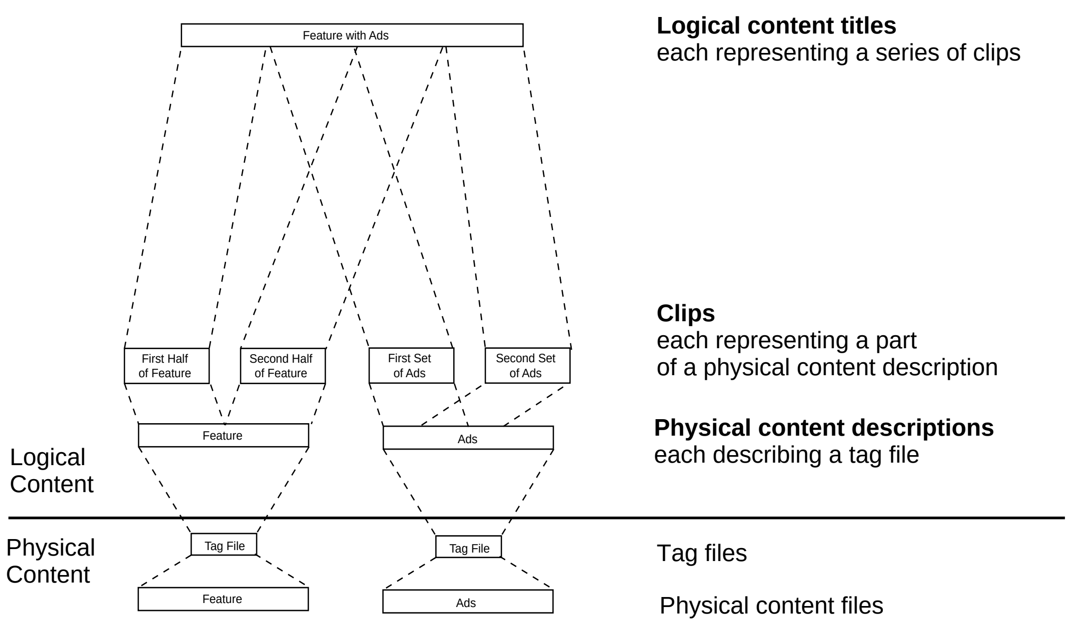
One-Step Encoding Making content available on the OVS typically involves obtaining a complete content file from an encoder, loading the file from tape, and creating an associated tag file with a tagging utility. Some encoders can encode a video content file and generate a tag file in real time (while the source video is playing) through a process called **one-step encoding**. The OVS can accept and store content from encoders in one step making new content available in real time.

Continuous Real-Time Feeds The OVS can store the most recent portion of a **continuous real-time feed**, or an endless stream of one-step encoded video. A continuous real-time feed enables a client to search the most recent portion of ongoing video, such as the last 12 hours of a 24-hour-a-day news broadcast.

Logical Content

Logical content is a collection of titles, each representing a series of parts of physical content files. As an administrator, you can define logical content to customize the multimedia data your clients can request. Figure 2–1 shows how you can define a logical content title representing a television feature containing advertisements based on individual physical content files containing the feature and the advertisement.

Figure 2–1 Logical content and physical content



Whenever you create a tag file for a physical content file, the tagging utility automatically stores this information in the database:

- a physical content description of the tag file
- a clip representing the entire physical content description
- a logical content title representing the clip

If you as the administrator do not define additional logical content titles and clips to customize the content your clients can request as in Figure 2–1, clients can only request the logical content titles for individual physical content files.

Since logical content definitions are stored in a database, defining logical content does not create new physical content files or modify existing ones.

Storage

The OVS stores physical content in the Oracle MDS (Media Data Store), a real-time file system for storing and delivering uninterrupted video in real time. This section describes the MDS.

The OVS optionally stores logical content in an Oracle database. For information on how Oracle databases store data, see the *Oracle Concepts* manual.

Real-Time Access

The MDS can tolerate disk errors and variance in disk latency and still provide real-time access to content. The MDS can also deliver video in real time while you are administering the MDS.

Files

The MDS can store these types of files:

- video files
- audio files
- BLOB (binary large object) files
- tag files

Write Consistency The MDS enforces write consistency on all MDS files:

- A file can be written by only one client at a time.
- A file cannot be renamed, removed, truncated, or locked into read-only mode while it is being written.
- If a process fails while writing to a file, the MDS makes the file available to be written by other processes after a few minutes.

Volumes

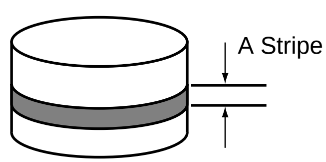
MDS files are stored in MDS **volumes** — named collections of disks. Each MDS volume stores files in a flat namespace. The MDS provides quick access to a file given a unique identifier, such as a filename, but is not meant to be a general purpose file system with nested hierarchical subdirectories to search and browse. Logical content provides that level of interaction.

Each volume has a **table of contents** that lists the files in the volume and their locations on disk.

Striping **Striping** means dividing a file into pieces and storing each piece on a different disk. Striping a file distributes access to the file across many disks, rather than concentrating it on one, reducing requests for each disk and improving performance when the file is accessed by many concurrent clients.

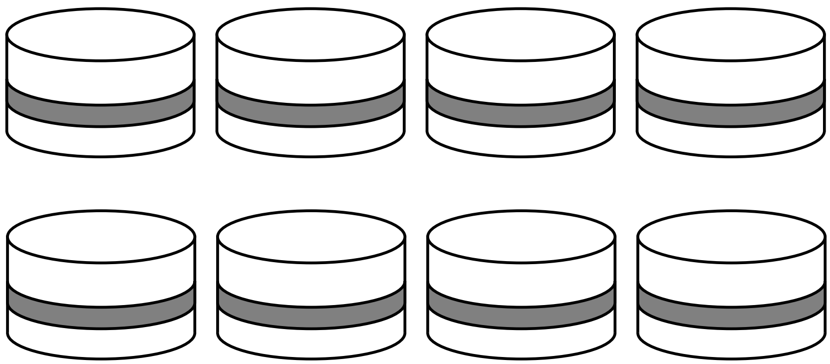
A **stripe** is a portion of a disk reserved for a single piece of a striped file. The size of a stripe is called a **stripe size**. The stripe size is the same for every disk in the volume. Figure 2–2 identifies a single stripe on a disk.

Figure 2–2 A stripe



The MDS supports striping on a volume-by-volume basis. In a striped volume each file is striped across all the volume's disks. To write a file to a striped volume, the MDS writes a stripe on each of the volume's disks beginning with the first. If the file requires more space than a single stripe on each disk in the volume, the MDS writes another stripe to each disk beginning with the first. The MDS continues to write stripes across all the volume's disks until the entire file has been written. The shaded area in Figure 2–3 shows a single stripe on each disk in a volume.

Figure 2–3 A striped volume



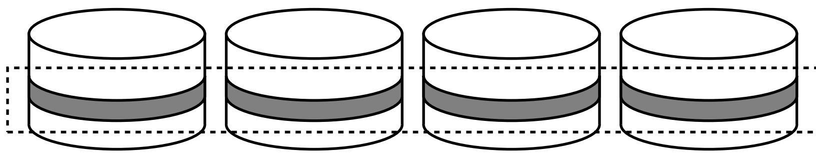
Striping is transparent to the user and a striped file appears logically as one contiguous sequence of bytes.

RAID (Redundant Arrays of Inexpensive Disks) Protection **RAID protection** means storing data redundantly so it remains accessible in the event of a disk failure. The RAID protection implemented for the MDS is similar to hardware-based RAID protection but also guarantees on-time delivery even if a disk or controller fails. RAID is similar to disk mirroring, but uses less space for redundant data. Together, striping and RAID offer tremendous performance benefits with a high degree of reliability.

The MDS supports RAID protection on a volume-by-volume basis. If RAID protection is enabled for a volume, the disks in the volume are divided into RAID sets. The number of disks in each RAID set is called the **RAID size**. The RAID size must be consistent across all RAID sets in a volume, so the number of disks in a volume must be a multiple of the RAID size.

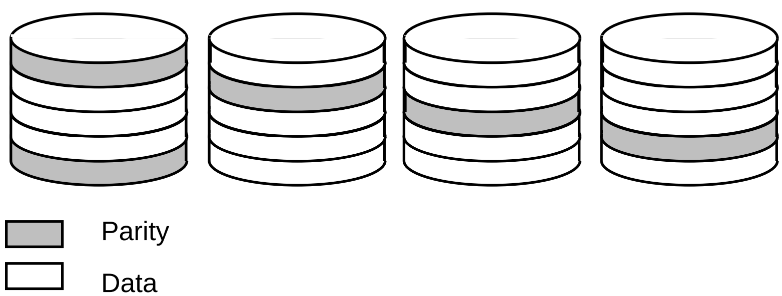
In a striped volume, the set of stripes that have the same logical location on the disks of a RAID set is called a **RAID stripe**. Figure 2–4 shows a RAID stripe within the dotted rectangle.

Figure 2–4 A RAID stripe



The MDS designates one stripe in each RAID stripe to store parity information, technically the "bitwise exclusive OR" of the other stripes. The disk containing the parity stripe rotates from one RAID stripe to the next as shown in Figure 2–5.

Figure 2–5 A RAID set with parity stripes



The amount of data contained in a RAID stripe is:

stripesize x RAIDsize

Since one of the stripes in the RAID stripe stores parity information, the amount of non-redundant data stored in a RAID stripe is:

stripesize x (RAIDsize–1)

The RAID stripe is the smallest unit of allocation in the MDS.

If a disk fails, the MDS can dynamically reconstruct its data from the other disks in the RAID set, using the parity information. The reconstruction occurs as the data is requested and is transparent to the user. You can also rebuild data from a failed disk onto a new disk with the rebuild utility.

The MDS can tolerate one failed disk in each RAID set and continue to deliver video without glitches, or disturbances in the video. If there are multiple concurrent failures in a RAID set, the missing data cannot be reconstructed and the volume can no longer play video.

Space Allocation When writing a file to a volume, the MDS writes each stripe sequentially across disks in each RAID set and writes each RAID stripe sequentially across RAID sets in the volume.

A volume’s table of contents is stored on the first RAID stripes of the first few RAID sets, depending on how the volume is defined. The MDS begins storing files in an empty volume with the first RAID stripe on the RAID set after the table of contents. The MDS writes sequentially across disks in the RAID set, writing one stripe on each disk. After filling a RAID stripe, the MDS writes sequentially across the first RAID stripe in the next RAID set, and so on. When the MDS reaches the last disk in the last RAID set, it returns to the first RAID set and proceeds to write to the second RAID stripe. The last RAID stripe allocated for the file may not be completely filled by the file. This space remains empty. To write another file to the volume, the MDS begins with the first disk in the **next** RAID set.

The number of stripes taken up by a file depends on the length of the file as illustrated in Figure 2–6. Each RAID stripe is labeled by the file containing it (A, B, or C) and the number of the RAID stripe within the file. Note that the figure does not distinguish data from parity information. All RAID stripes, including the one containing directory structure information, include an associated parity stripe.

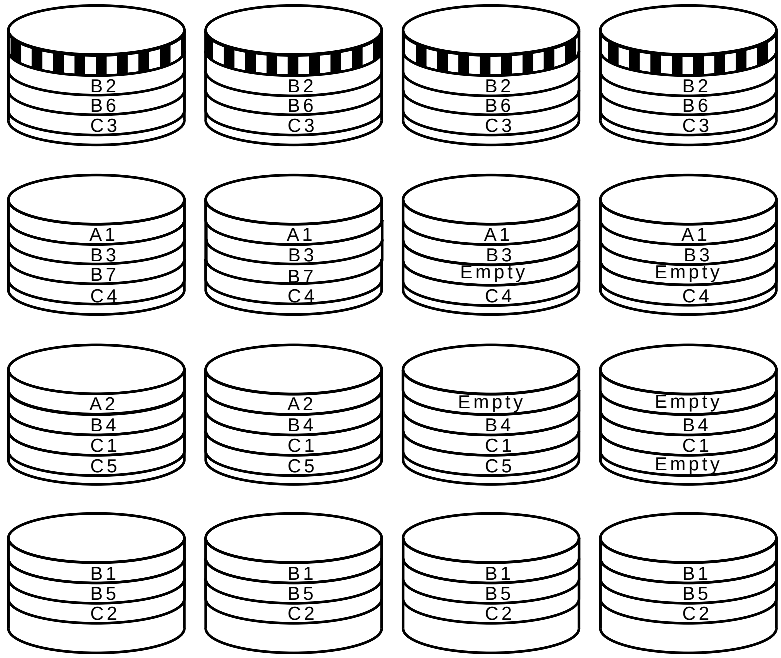
The space allocated for a file may exceed the actual file size by:

(stripesize x (RAIDsize–1))-1

for a 1 byte file.

If a file is deleted from a volume, the MDS can reclaim that file’s space by storing one or more new files that fit there. The MDS cannot distribute a file over noncontiguous RAID stripes. The defragmenter utility can make the noncontiguous empty RAID stripes contiguous.

Figure 2–6 MDS Volume Showing Space Allocations



= table of contents
A1 = first RAID stripe of file A
A2 = second RAID stripe of file A
B1 = first RAID stripe of file B
Empty = unused portion of allocated RAID stripe

Spare Disks A **spare disk** is a designated empty disk in a volume onto which data from a failed disk can be rebuilt. A spare disk is not part of a RAID set. On a video server computer that does not support hot-swapping disks (replacing disks while the computer is online), a volume must contain a spare disk to support rebuilding data while delivering video. Without a spare disk the OVS cannot deliver video while the failed disk is replaced and rebuilt. Spare disks are unnecessary on video server computers that support hot-swapping disks.

MDS Utilities

The OVS provides a group of general operating system command line utilities that allow you to perform standard file operations on files and volumes in the MDS. More specialized MDS utilities include:

- tape archival utility—enables you to archive, backup, and retrieve a group of files to and from a single archive file in tape archive (TAR) format or a sequence of tapes in an Exabyte stacker
- defragmenter utility—allows you to defragment volumes to reduce wasted space without taking the OVS offline
- rebuild utility—enables you to:
    - replace a failed disk and rebuild data onto the replacement while the MDS delivers video, if your operating system supports hot-swapping disks, or to
    - rebuild data from a failed disk onto a spare disk from redundantly stored data while the MDS plays video, or to
    - take the OVS offline, replace a failed disk, and rebuild data onto the replacement

MDS utilities accept wildcard specifications as general operating system command-line utilities do.

HSM (Hierarchical Storage Management)

The MDS is part of the HSM (hierarchical storage management) supported by the OVS that stores data in:

- memory (primary storage)
- MDS volumes on disk (secondary storage)
- a tape archive (tertiary storage)

You can store large numbers of files in a tertiary tape archive.

HSM Utilities The OVS provides HSM utilities that enable you to:

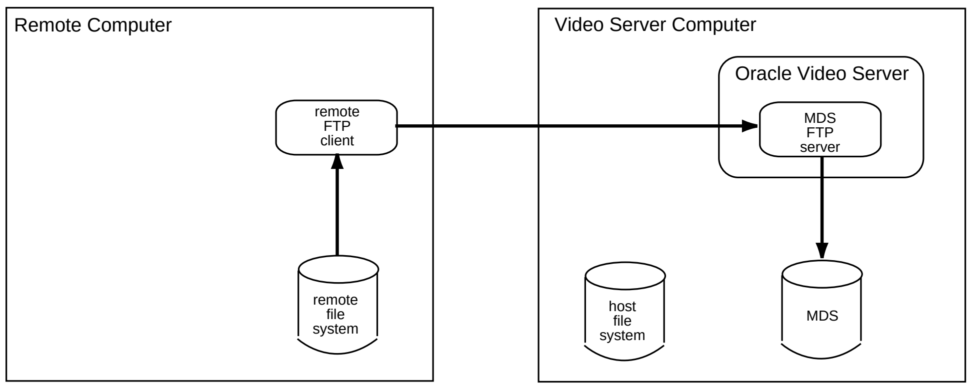
- load files from tertiary storage to the MDS before clients need to access them
- archive files from the MDS to tertiary storage if clients will not need them for a period of time, to make room in the MDS for other files

Remote Access

The MDS is accessible by processes on remote computers as well as on the video server computer.

The OVS also supports binary file transfer between FTP clients on remote computers and the MDS. Figure 2–7 shows a direct FTP file transfer from a remote computer to the MDS.

Figure 2–7 Remote file transfer to MDS through MDS FTP server



The alternative, an FTP transfer from the remote computer to the video server computer host file system and then a transfer to the MDS with an MDS utility, would require an additional copy operation.

Delivery

The OVS can deliver a video stream to a client in real time on demand. A *stream* of video is played by the client as it is delivered by the OVS, rather than completely downloaded by the OVS and then played by the client.

Requesting Video

A client can request the OVS play video by requesting it. In a request, a client identifies video with an **asset cookie**, which stands for either a logical content title or a tag file. The client can request a series of asset cookies so that the OVS plays series of logical content titles or tag files without interruption.

Rate Control

Rate control enables a client to pause, resume, and reposition a video stream. The OVS supports these rate control functions:

- Pause—stop the video and maintain its current position
- Resume—continue to play video from the point at which it was paused
- Seek forward or backward—jump directly to a later or earlier point in the video
- Scan forward or backward—visual fast forward or rewind for selected formats and clients

Real-Time Feeds

The OVS can deliver a **real-time feed**, or a stream of one-step encoded video as it is encoded and stored. A client viewing a real-time feed can scan and seek backward to access any of the encoded video. For example, a client viewing a live sports event through a real-time feed can replay key points in the event as the event continues to be encoded and stored.

Scheduled Video

The OVS enables you to schedule a logical content title to begin playing on a specific network channel at a specific time. Clients can then “tune in” to see the title they want with no signals required from the client to the OVS. Scheduling enables you to implement:

- regularly-scheduled television broadcasting
- pay-per-view
- NVOD (near video-on-demand)

NVOD (**near video-on-demand**) enables a client to view a film from the beginning at regular intervals (for example, every 15 minutes). By scheduling the same title on different channels at staggered times, you can enable clients to see the film from the beginning by tuning to a channel when the film is beginning there. Also, a client can effectively pause and seek forward and backward in 15-minute increments by tuning among different channels that began the film at different times.

Looping Content

The OVS can *loop*, or continuously repeat playback of a logical content title. Looping content can be useful for displaying a moving logo or for filling the time between the end of a scheduled film and the beginning of the next with advertisements or public service announcements.

Network

The OVS can deliver data at different bit rates over a wide variety of networks and protocols. Supported protocols include:

- UDP (User Datagram Protocol)
- TCP (Transmission Control Protocol)
- ATM (Asynchronous Transfer Mode)
- AAL5 (ATM Adaptation Layer 5)
- DVB (Digital Video Broadcast)

Oracle Video Server Manager (VSM)

Oracle Video Server Manager (VSM) is an application with a graphical user interface that monitors and manages the OVS and its clients. VSM enables you as the administrator to perform tasks including:

- starting and stopping instances of the OVS
- viewing the status of critical OVS services
- create and manage logical content titles and clips
- monitoring:
  - MDS disks and files
  - HSM tertiary storage tapes
  - real-time feeds
  - clients of the OVS
- defragment MDS volumes
- loading and registering physical content files
- scheduling and playing videos

For more information on VSM, see *Getting Started with Oracle Video Server Manager.*

Graphical User Interface

The VSM application provides a Java-based user interface. It is designed for ease of use so you as the administrator can effectively manage the OVS system quickly with minimal training.

Oracle Media Net

Oracle Media Net is a networking infrastructure that enables the OVS and its clients to communicate in a distributed computing environment. Using heterogeneous network protocols, Oracle Media Net enables connectionless communication among the various OVS system components running on different platforms. Oracle Media Net is Oracle's implementation of CORBA (Common Object Request Broker Architecture), so heterogeneous services programmed in different languages and distributed over different computers in a network can communicate without concern for each other's location or the details in transporting or converting data among them. CORBA is an open standard for distributed network objects defined by the OMG (Object Management Group) that provides the mechanisms by which applications transparently make requests and receive responses.

Oracle Media Net abstracts the network for the application developer and the network protocols underlying Oracle Media Net are transparent to the components using it. You can create an application once and deploy it on many different platforms without changing its Oracle Media Net calls. Oracle Media Net also automatically bridges connections across network types, making multiple protocols in the same network transparent and presents a reliable message service, even if the underlying network protocols use unreliable messaging.

For complete conceptual information on Oracle Media Net, see the *Oracle Media Net Developer's Guide*.

Clients and Servers

When an application in an Oracle Media Net network needs to perform an operation that requires resources or functionality not available locally, it requests that a remote application perform the operation. The requesting application is called the *client* and the remote application performing the operation is called the *server.*

Objects, Interfaces, and Servers

This section briefly defines terms used in a CORBA environment:

- An *object* is a means of encapsulating functionality and providing a simple and consistent interface that defines how clients can use that functionality.
- An *interface* defines a named set of related operations. The description of an object is a set of operations defined in an interface.
- A *server* is a single computer process that implements one or more interfaces.

You can define the interfaces for your own objects by writing an IDL (Interface Definition Language) file. Oracle Media Net provides an IDL compiler for creating C programming language files from your IDL files.

Oracle Media Net also provides an IFR (interface repository) that you can load with object interface definitions. You can then write applications that use DII (dynamic invocation interface) to query the IFR at run time and construct requests for objects defined there.

Scalability and Load Balancing

If demand for a server (such as an OVS service) increases, you can start additional instances of the server to better handle it. Oracle Media Net can distribute requests for a server across available instances to balance the load among them. If one instance fails, Oracle Media Net can redistribute new requests across the remaining instances.

Logging Error and Warning Messages

Oracle Media Net logs error and warning messages for the OVS system. For example, if the OVS rejects a client connection request, messages in the log explain why.

Logging is configurable:

- You can configure a common event logging daemon to write messages for all components of the OVS system to a common logfile.
- You can run multiple event logging daemons. For example, you can configure multiple daemons in a hierarchy so that several daemons each log messages for a group of components and forward only the severe messages to a single daemon. This daemon then writes to a logfile or console and alerts an application service of only the most severe messages. You can then program the application service to notify the administrator.

Oracle Video Client (OVC)

Oracle Video Client (OVC) software is useful for developing multimedia applications that receive and display video and for accessing and receiving video from the OVS at run time.

Application Development

OVC software enables you as the application developer to develop interactive, video-based multimedia applications. OVC software includes these tools to help you build applications:

- **Oracle Video Java Library**
  The Oracle Video Java Library enables you as the application developer to embed video in a Java application that runs with the appletviewer or from the command line.
- **Oracle Video ActiveX Control**
  The Oracle Video ActiveX Control is an ActiveX control that enables you to embed video in compliant 32-bit multimedia applications, such as Microsoft Visual Basic, Developer 2000 / Oracle Forms, and Oracle Power Objects, that start, stop, and seek locations within video streams from the OVS.
- **Oracle Video Web Plug-in**
  The Oracle Video Web Plug-in enables World Wide Web pages to embed video streams from the OVS. The Oracle Video Web Plug-in is compatible with Internet browsers that support Netscape plug-ins, such as Netscape Navigator and Microsoft Internet Explorer. You can also use Java and JavaScript and with Netscape's LiveConnect interface to customize the control of the Oracle Video Web Plug-in, for example, to add clickable buttons or icons to play, pause, and seek video and pop up lists of available content.

Run-Time Capability

At run time, OVC software runs on the client device and:

- connects to the OVS
- accepts input from the client application
- requests video and audio from the OVS
- receives video and audio in MPEG or OSF format and displays it

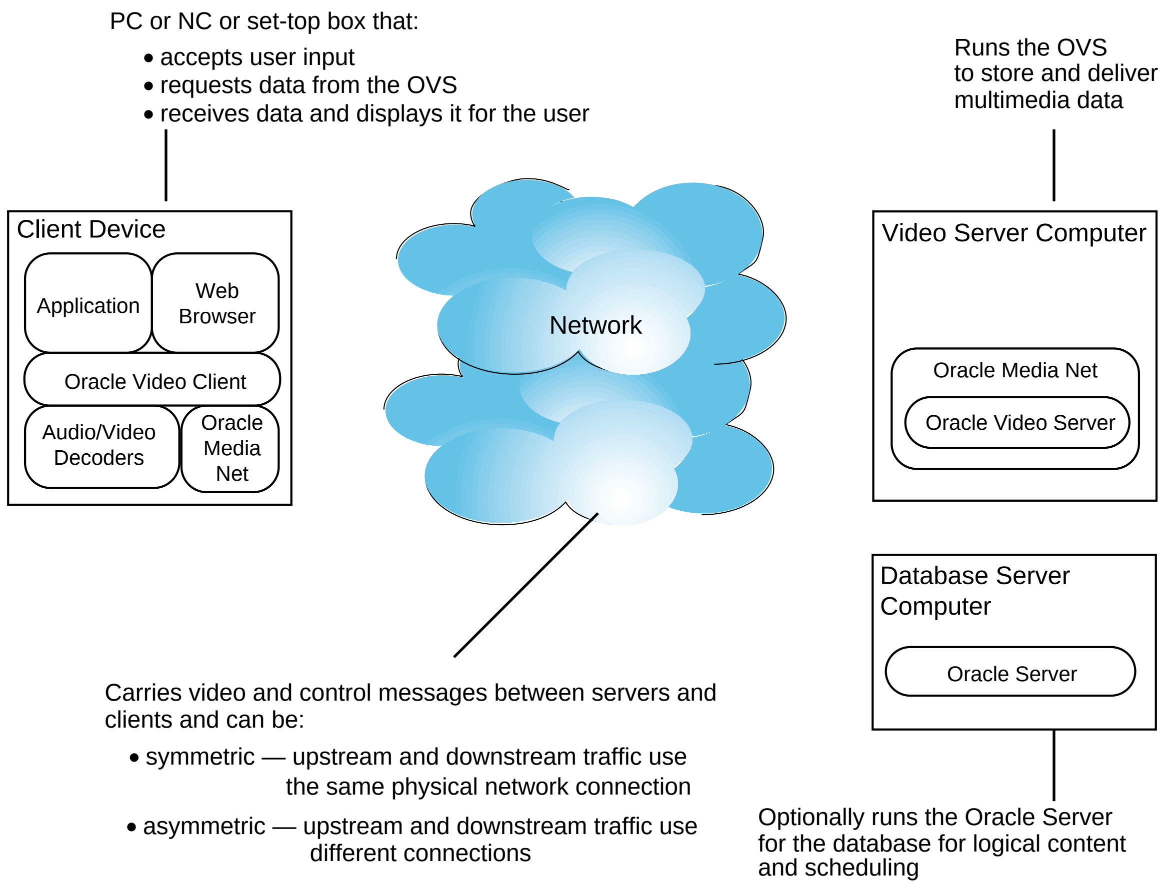
Specialized third-party software or hardware decoders decompress video data. OVC software applications can receive and display video content using Microsoft's ActiveMovie framework and ActiveMovie-compliant decoders. The decoder requires smooth, reliable video delivery and the network may sometimes deliver packets late or out of order. OVC software ensures proper decoder operation by re-ordering packets properly and buffering a small amount of data so the decoder is consistently provided with ordered video data.

OVC software can dynamically connect to and disconnect from different instances of the OVS without stopping and restarting.

OVS System Hardware and Network Architecture

This section presents the OVS system hardware. The OVS system uses the hardware components shown in Figure 2–8.

Figure 2–8 OVS system hardware



In a production environment, to ensure real-time delivery with no glitches, the video server computer must be dedicated to running the OVS:

- If you use logical content or scheduling, the Oracle Server for the database must run on a separate database server computer.
- If you develop server-side applications to run with your OVS, such as an application responsible for authenticating and billing clients, they must run on separate application server computers.

How the OVS System Communicates

This section presents these concepts related to communication in the OVS:

- Circuits
- Sessions

and discusses these processes:

- How Client Devices Connect to the OVS
- How the OVS Delivers Video

Circuits

A **circuit** is a path of communication between the client and the OVS. A circuit can be one of these types:

- **upstream**—carrying control messages (such as requests for video) from the client to the OVS
- **downstream**—carrying control messages (such as acknowledgments of requests) and video from the OVS to the client
- **bidirectional**—carrying control messages in both directions between the client and the OVS

Sessions

A client connecting to the OVS establishes a session. A *session* consists of:

- an identifier for the client
- one or more circuits through which control messages and video are transported between the client and the OVS
- **resources**, or state information for the client and its connection to the OVS

The types of circuits within a session depend on the symmetry of the network the OVS system uses. This section discusses:

- Sessions in Symmetric Networks
- Sessions in Asymmetric Networks

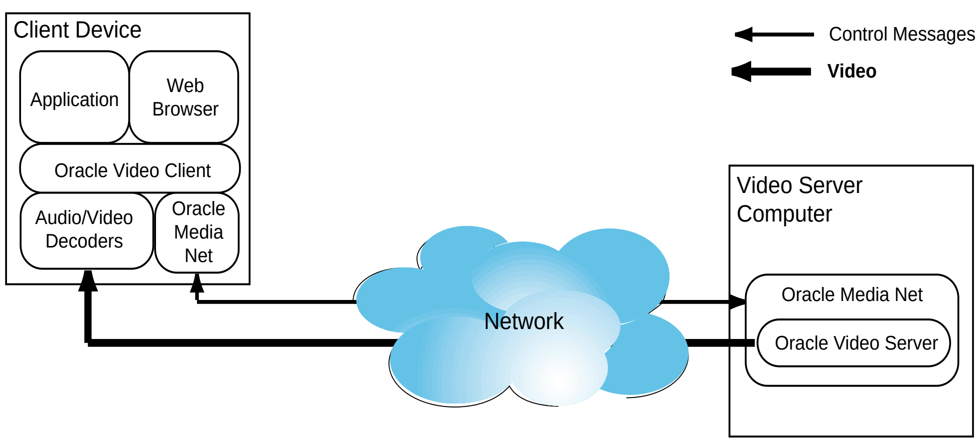
Sessions in Symmetric Networks

In symmetric networks each client session contains two circuits:

- a bidirectional circuit that carries control messages between the client and the OVS
- a downstream circuit that delivers video from the OVS to the client

Figure 2–9 shows a client and its session with the OVS.

Figure 2–9 Communication in a symmetric network



All IP (Internet Protocol) networks are symmetric by definition.

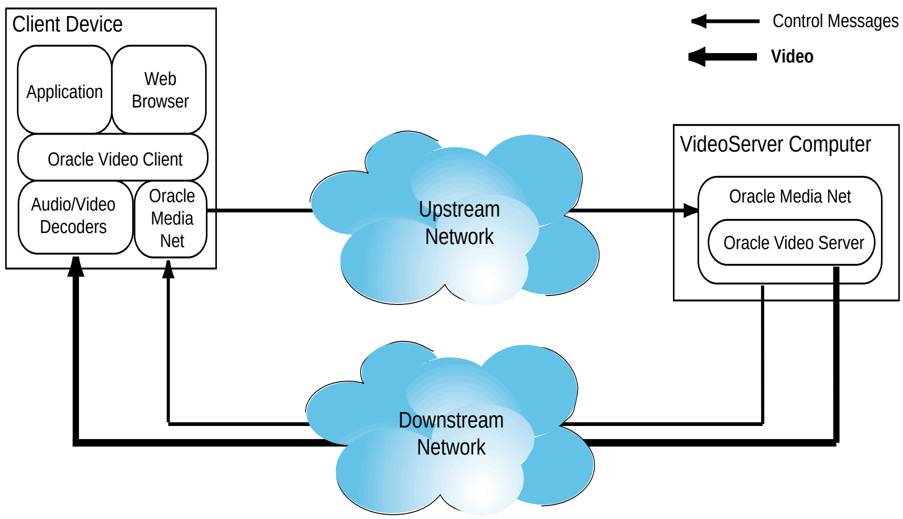
Sessions in Asymmetric Networks

In asymmetric networks each client session contains:

- an upstream circuit carrying control messages from the client to the OVS, and
- a downstream circuit carrying control messages and delivers video from the OVS to the client

Figure 2–10 shows a client and its session with the OVS.

Figure 2–10 Communication in an asymmetric network



How Client Devices Connect to the OVS

A client connects to the OVS by following these steps:

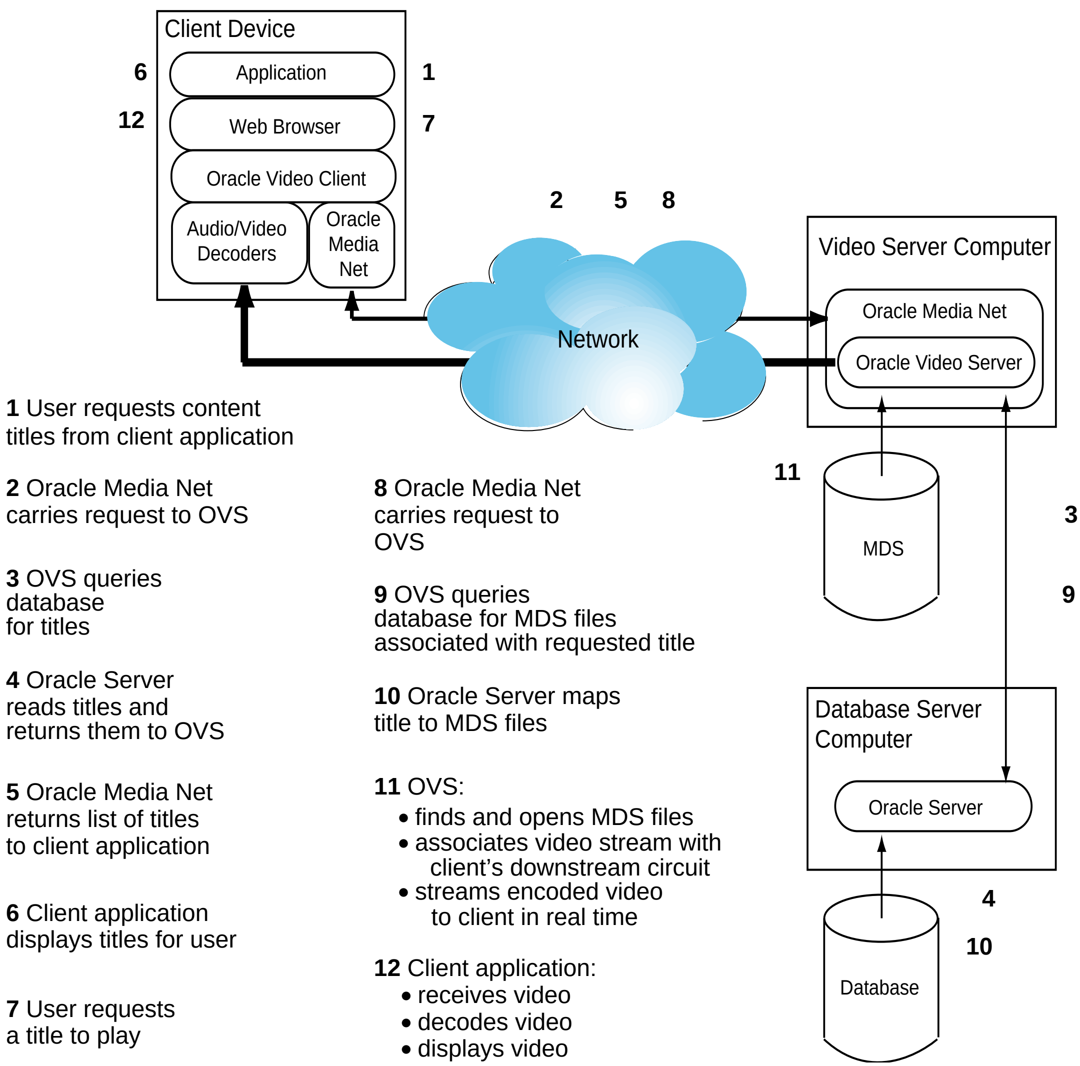
1. OVC obtains a Media Net address from the OVS. This address allows the client to communicate with components of the OVS through Oracle Media Net.
2. OVC establishes a session with the OVS that includes one or more circuits so that the OVS can send and receive control messages to and from the client and deliver video to the client.

Some broadband networks contain a service provider called a **level 1 gateway** (L1GW) that acts as a proxy for the client by contacting the OVS and establishing a session and circuits for the client.

How the OVS Delivers Video

Figure 2–11 illustrates steps the OVS system follows to deliver video. This example uses a symmetric network. Steps 7 through 12 make up a video **round trip**, or a request from an application met by video delivery from the OVS.

Figure 2–11 An OVS system round trip



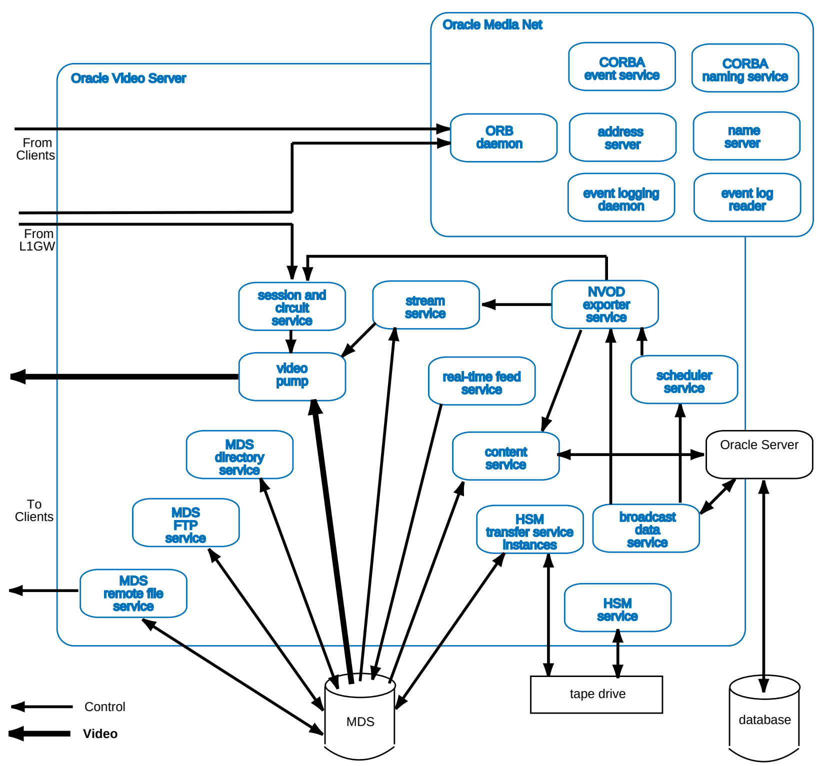
Components of Oracle Video Server and Oracle Media Net

The individual components of Oracle Video Server and Oracle Media Net work closely with each other and with client applications. This section describes Oracle Video Server and Oracle Media Net components shown in Figure 2–12.

Figure 2–12 also shows typical paths of communication for connecting to Oracle Video Server and delivering video. Many messages that do not appear in the figure pass between Oracle Video Server components and client devices via the ORB daemon.

Each Oracle Video Server and Oracle Media Net process can accommodate multiple clients, depending on the type of video server computer. For large numbers of clients, the Oracle Video Server can run multiple instances of most processes. Figure 2–12 shows only one instance of each process.

Figure 2–12 Components of Oracle Video Server and Oracle Media Net



Oracle Media Net ORB Daemon

The Oracle Media Net ORB (object request broker) daemon receives requests for services from clients and routes them to the appropriate servers. For servers with multiple instances, the ORB daemon balances the load of requests across the instances and redistributes requests if an instance fails. These clients and servers can be located on either the same computer or on different computers on a network. The ORB daemon manages the following details and keeps them transparent from clients and servers:

- where client and server applications are located
- how to transport and convert data between different types of computers

When a service is started, it registers its name and Media Net address with the ORB daemon.

Oracle Media Net Name Server

The Oracle Media Net name server maps the names of ORB daemons to their Media Net addresses. When an ORB daemon is started, it registers its name and Media Net address with the name server.

An OVS system can run only one name server.

Do not confuse the name server with the Oracle Media Net CORBA naming service discussed later in this section.

Oracle Media Net Address Server

Each component has an Media Net address that uniquely identifies it in the Oracle Media Net network. The Oracle Media Net address server maps Media Net addresses to corresponding physical addresses:

- When a client device connects to the OVS, it provides its physical address and is assigned a Media Net address by the address server.
- When a server process is started, it registers its physical address with the address server and is assigned a Media Net address.
- Before a client can send a message to the server, it contacts the address server to find the server’s physical address.

An OVS system can run only one address server.

The address server is responsible for invalidating addresses of failed processes. When a process on the OVS system is idle, it periodically sends a **heartbeat**, a message announcing its existence, to the address server. If the address server does not detect a message from a process for a period of minutes, the address server assumes a failure and invalidates the process' address.

Oracle Media Net Event Logging Daemon

The Oracle Media Net event logging daemon writes error and warning messages for the OVS system components to a logfile. The event logging daemon is configurable:

- You can configure the event logging daemon to log either all messages or only those above a severity threshold.
- You can use message logging filters to configure the event logging daemon to log specific information about components.
- You can configure the maximum size of a log file. After filling a log file, the event logging daemon automatically begins writing to a new one.

Oracle Media Net Event Log Reader

The Oracle Media Net event log reader interprets and displays information written to the logfile by the Oracle Media Net event logging daemon to make this information easier to interpret.

Oracle Media Net CORBA Event Service

An *event* is data that indicates a change in an object. The Oracle Media Net CORBA event service provides for the notification of clients when an event occurs. When something of note occurs in an object, an event may be generated to notify clients of the occurrence so each client can react. For example, when a client device requests a video content file, the stream service prepares to play the file and notifies the CORBA event service. A billing application may note this event to bill the client application for playing the video.

An OVS system can run only one CORBA event service.

Oracle Media Net CORBA Naming Service

The Oracle Media Net CORBA naming service enables clients to locate and obtain object references from name strings. The CORBA naming service associates objects with names and organizes these object names hierarchically in contexts similar to the way a file system organizes file names in directories. For example, the Oracle Media Net CORBA event service uses the CORBA naming service to name event channels.

An OVS system can run only one CORBA naming service.

Do not confuse the CORBA naming service with the Oracle Media Net name server discussed earlier in this section.

Session and Circuit Service

The session and circuit service allocates sessions and circuits for clients connecting to the OVS. An OVS can run only one session and circuit service.

The session and circuit service may consult a third-party component called a channel manager when allocating a circuit for a client. The channel manager has specific network implementation information that is not available to the session and circuit service. After allocating a circuit, the session and circuit service sends it to the channel manager for approval. The channel manager can approve the circuit or suggest another one.

When a client ends its session, the session and circuit service deallocates all the session’s circuits and resources.

Oracle MDS Directory Service

The MDS (Media Data Store) directory server controls access to MDS files and manages their layout on disk.

Processes that read or write MDS files must first gain access through the MDS directory server. For example, when a video pump is instructed to play a video file, it opens the file by sending a message to the MDS directory server that returns a small structure describing the layout of the file on disk. With this structure, the video pump accesses the file’s contents directly. This direct access prevents the MDS directory server from becoming an I/O bottleneck.

Oracle MDS Remote File Service

The MDS remote file server:

- provides access to MDS files for remote processes such as MDS utilities and applications on remote computers
- delivers BLOBs to clients on request

Oracle MDS FTP Service

The MDS FTP (file transfer protocol) service performs binary file transfers between the MDS and FTP clients on remote computers. A single MDS FTP service runs on the video server computer. The MDS FTP service listens for requests from remote clients on a dedicated port.

The MDS FTP service only accesses files in the MDS, not on the video server computer host file system. Host files can be accessed through the video server computer host FTP service which listens for requests on a different dedicated port.

HSM Service

The HSM (hierarchical storage management) server controls access to files in tertiary storage and manages their layout in a way similar to how the MDS directory server manages files in the MDS.

HSM Transfer Service

The HSM transfer service copies files between tertiary storage and MDS volumes on disk when requested by an HSM utility.

Stream Service

The stream service handles requests from clients for video and audio content. When the stream service receives a request, it obtains necessary information to meet the request:

- The stream service contacts the content service to resolve a request for a logical content title to one or more tag files.
- The stream service reads the tag file(s) associated with the request. Each tag file describes which portion of a requested file must be delivered to meet the request.

The stream service tells the video pump which parts of the physical content files to play.

Video Pump

The video pump reads video files from the MDS and delivers them to the network in real time. When a client requests video, the video pump receives a message from the stream service, reads the appropriate portion of the file from the MDS, and sends video data over the network through the appropriate downstream channel to the client.

Real-Time Feed Service

The real-time feed service accepts this input from a one-step encoder:

- encoded content
- associated tag information

To support one-step encoding, the real-time feed service writes the encoded content to one or more content files and the tag information to an associated tag file. The stream service and video pump can then play the content.

To support a continuous real-time feed, the real-time feed service maintains a pool of physical content files and writes the encoded content to one file after another. The real-time feed service deletes the oldest files as they become out of date and writes new ones. The real-time feed service also maintains a single tag file for all physical content files in the pool.

Content Service

The content service maintains information that maps logical content to physical content in the database.

When creating a tag file for a physical content file, the tagging utility calls the content service to generate logical content based on the content file. As the administrator, you can also call the content service to generate logical content titles each representing a series of parts of physical content files.

A client application can query the content available in the OVS system through the content service. On request of the client application, the content service reads from the database and returns a list of logical content to the client.

The content service also receives requests from the stream service to resolve logical content to physical tag files.

Broadcast Data Service

The broadcast data service manages broadcast scheduling information in the database. The broadcast data service reads broadcast scheduling information, such as schedules, from the database and writes it to memory where it can be read by the scheduler service and by exporter services, such as the NVOD exporter service. The broadcast data service also provides interfaces enabling you to schedule broadcast events.

Scheduler Service

The scheduler service keeps track of the current time and a list of scheduled events and notifies the appropriate exporter service, such as the NVOD exporter service, when a broadcast event is scheduled to occur.

NVOD Exporter Service

The NVOD exporter service plays a scheduled logical content title on a scheduled channel when notified by the schedule service. You can use this service to handle any scheduled video, including NVOD, pay-per-view, or regular TV broadcasting.

Networking in the OVS System

The OVS system can be deployed on many different network types. Depending on the environment, these networks can be either:

- a PSTN (public switched telephone network) using IP (Internet Protocol) and TCP (Transmission Control Protocol) or UDP (User Datagram Protocol) network used in Internet environments
- a LAN (local-area network), typically used in intranet environments
- Broadband Network, used in the broadband environment

PSTN

In Internet environments, clients communicate with both an ISP (Internet Service Provider) and the OVS. These communications use separate network connections:

- Clients communicate with an ISP through a PSTN using IP. A PSTN may consist of:
  - POTS (plain old telephone service)
  - ISDN (Integrated Services Digital Network)
  - ADSL (Asymmetric Digital Subscriber Loop)
- Clients communicate with the OVS through a TCP or UDP network connection.

LAN

In intranet environments, the OVS system uses switched Ethernet on a LAN to ensure consistent error-free video delivery and to reduce network congestion. Unlike shared Ethernet in which every computer reads every data packet, switched Ethernet uses a hardware hub that provides a dedicated link to each client. The dedicated link allows the hub to physically map each client IP address to the communication channel between the hub and the client. The switched Ethernet hub examines the IP address of each packet as it moves through the hub and routes it directly to its destination segment ensuring that data in the communication channel between the hub and a client is destined for that client. Each client views only its data and does not spend time examining packets addressed to others on the same communication channel.

The switched Ethernet hub also has high capacity input allowing for the multiplexing of video streams from the server to many clients from a single line. A high capacity interface, such as FDDI (Fiber Distributed Data Interface) or CDDI (Copper Distributed Data Interface), handles many video streams from the video server computer at once, with a dedicated segment for each client, ensuring optimum performance for both the dedicated segments and the entire network.

The rest of this section explains why an OVS system using a LAN requires switched Ethernet and discusses these topics:

- Ethernet
- Demands of Video on LANs

Ethernet

This section discusses these Ethernet terms:

- Packets
- Bus/Broadcast Topology
- Best-Effort Delivery

Packets A **packet** is the basic unit of information that travels between computers on a LAN. All data transmitted over a LAN, whether a small message or a large video file, is divided into packets. Each packet contains a header and data. The header contains:

- information about data in the packet
- sequencing information
- the address of the receiving computer
- a checksum of the packet data

For Oracle Media Net packets, the receiving computer gathers the packets it receives and reassembles them based on the packet sequence in the header. The receiver can identify missing packets based on information in the headers of the received packets and corrupted packets based on the checksum information in their headers. If a packet is missing or has been corrupted, the receiver requests that the sending computer resend the packet and waits for its return before reassembling the data for the user.

Bus/Broadcast Topology The most common type of Ethernet topology is a **bus topology**, in which all computers on the network are connected to the same communication channel. When one computer wants to send data over the network to another, it checks the data channel for data packet traffic and broadcasts the first packet when the channel is clear. If the data takes up multiple packets, the sender waits again until the channel is clear to transmit the next packet. This continues until all the packets have been sent. As a packet is broadcast on the channel, it is seen by all computers on the network. The packet is only useful to the computer to which it is addressed, denoted by the address in the packet header. This type of network is called **shared Ethernet** because all computers on the network share the same data channel.

Best-Effort Delivery Ethernet uses **best-effort delivery**, which means that when data is sent over the network, every effort is made to deliver the data. However, if the receiving computer is turned off or if there is a network problem, the data is lost. Ethernet does not notify the sending computer when a message is received or dropped.

Demands of Video on LANs

A shared Ethernet network is sufficient for non-real-time applications such as databases or word processors, in which delays of a fraction of a second in receiving data are not critical and are often not noticed by users.

However, applications receiving video are time-sensitive and depend on isochronous, or time-based, delivery. Data packets must arrive on time to be useful, or the video suffers a brief interruption and the user notices a glitch in the video. Glitch-free video is not easy to deliver on a shared Ethernet network because:

- All data traffic from one computer to another is read by all computers on the network.
- Video cannot be delivered as a continuous string of packets, but as multiple individual packets that compete equally with other applications for network resources.
- The large number of packets needed to deliver video consumes network resources. For example, 20 video streams at an encoding rate of 1.5 Mbps (Megabits per second) require a total network rate of 30 Mbps, which is much greater than standard 10baseT Ethernet network rates.
- A shared Ethernet network card can drop packets due to transmission timeouts in heavy network traffic.

Once a video stream starts, it must continue and finish uninterrupted by other data transfers or by glitches. To ensure uninterrupted video:

- Video cannot be interrupted by other network traffic such as large file transfers.
- The network must be capable of sustained data transfer rates that support the maximum video encoded rate multiplied by the number of concurrent video streams.

Switched Ethernet meets both of these requirements.

Broadband Network

In a broadband environment, one broadband network delivers video from the OVS to clients and another carries messages in both directions between clients and the OVS.

The network may use various protocols, such as ATM (Asynchronous Transfer Mode) or X.25 (for control messages only), depending on the network protocol.

Index

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W