

StudioCentral™ Library Overview

Document Number 007-3830-002

CONTRIBUTORS

Written by Ardene Kannelly and Susan Patick

Illustrated by Dany Galgani

Production by Amy Swenson

Edited by Cindy Kleinfeld

Engineering and other contributions by Tim Campos, George Eckel, Jeff Hollar,
J.K. Muthukumarasamy, Paul Stevens

St. Peter's Basilica image courtesy of ENEL SpA and InfoByte SpA. Disk Thrower
image courtesy of Xavier Berenguer, Animatica.

© 1999, Silicon Graphics, Inc.— All Rights Reserved

The contents of this document may not be copied or duplicated in any form, in whole
or in part, without the prior written permission of Silicon Graphics, Inc.

LIMITED AND RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set
forth in the Rights in Data clause at FARS 52.227-14 and/or in similar or successor
clauses in the FAR, or in the DOD, DOE or NASA FAR Supplements. Unpublished
rights reserved under the Copyright Laws of the United States.

Contractor/manufacture is SGI, 1600 Amphitheatre Pkwy., Mountain View, CA
94043-1351.

Silicon Graphics and IRIX are registered trademarks and SGI, the Silicon Graphics
logo, O2, Origin, StudioCentral, and XFS are trademarks of Silicon Graphics, Inc.
AMASS and EMASS are registered trademarks of ADIC. Adobe Photoshop is a
registered trademark of Adobe Systems, Inc. Apple and Macintosh are registered
trademarks of Apple Computer, Inc. Avalon Archive Manager is a trademark of
Avalon Consulting Group. INFORMIX is a registered trademark and INFORMIX
Online Dynamic Server is a trademark of Informix Corporation. Microsoft,
Windows, and Windows NT are registered trademarks of Microsoft Corporation.
Oracle is a trademark of Oracle Corporation.

Contents

	List of Figures	vii
	About This Guide	ix
	Audience for This Guide	ix
	Related Reading	ix
	Typographical Conventions Used in This Guide	x
	Important Terms in This Book	xi
	StudioCentral Terminology	xi
	StudioCentral Library User Terminology	xii
1.	Functional Overview	1
2.	StudioCentral Applications	7
	StudioCentral Clients	7
	StudioCentral Windows NT Client Scenario	8
	StudioCentral Macintosh Client Scenario	14
	Admin Tool	19

- 3. **Software Components** 23
 - Software Components 23
 - Core Software 25
 - Metadata Services 25
 - Content Services 26
 - Database Keys 28
 - Content Server Keys 28
 - Asset Typing 28
 - Datamodel Manager 29
 - Backup and Restore Services 29
 - Application Programming Interface 29
 - Asset Store API 29
 - Datamodel Manager API 30
 - Access Control 30
 - Asset Transfer Service 30
 - Repository Plug-ins 30
 - Policy Modules 31
 - Asset Services 31
 - Migration Manager Service 31
 - Consistency Checking Services 31
 - Developing New StudioCentral Components 32
 - Client Applications 32
 - ATS 33
 - C++ 33
 - Command-Line Utilities 33
 - SC/Perl 33
 - Content Servers 33
- 4. **Configurations** 35
 - Configuration Types 35
 - Single-Server Configurations 35
 - Multiple-Server Configurations 36

StudioCentral Installation and Administration Tasks	37
StudioCentral Administrator	37
Database Administrator	38
Glossary	39
Index	45

List of Figures

Figure 1-1	Functional Overview	4
Figure 2-1	Windows NT Client: StudioCentral Explorer Window	9
Figure 2-2	Windows NT Client: Catalog Window	10
Figure 2-3	Windows NT Client: Find Window	11
Figure 2-4	Windows NT Client: Find Results Display	11
Figure 2-5	Windows NT Client: Information Window	12
Figure 2-6	Windows Client: Second Version of Asset in Information Window	13
Figure 2-7	Macintosh Client: Asset Browser	14
Figure 2-8	Macintosh Client: Check In Window	15
Figure 2-9	Macintosh Client: Find Window	15
Figure 2-10	Macintosh Client: Find Results Window	16
Figure 2-11	Macintosh Client: Information Window	17
Figure 2-12	Macintosh Client: Second Version of Asset in Information Window	18
Figure 2-13	Admin Tool: Adding a User to a Group	19
Figure 2-14	Admin Tool: Setting Permissions for an Asset	20
Figure 2-15	Admin Tool: Searching for Assets	21
Figure 2-16	Admin Tool: Determining Status of Services	22
Figure 3-1	StudioCentral Software Components	24
Figure 4-1	Single-Server Configuration	36

About This Guide

This document contains an overview of the StudioCentral Library digital asset management system:

- Chapter 1, “Functional Overview”
- Chapter 2, “StudioCentral Applications”
- Chapter 3, “Software Components”
- Chapter 4, “Configurations”
- “Glossary”

Audience for This Guide

This guide is for StudioCentral Library administrators, developers, and others who want an overview of StudioCentral Library and how it is used. This guide is a prerequisite to understanding the documents that describe how to install, configure, and administer StudioCentral Library, and how to develop StudioCentral Library applications.

Related Reading

The following are related documents:

- *StudioCentral Library Release Notes* contains a summary of this release and special conditions and requirements that may apply to your use of StudioCentral Library.
- *StudioCentral Library Installation Instructions (007-3934-*nnn*)* is the booklet that accompanies the StudioCentral Library CD and provides information for performing a basic installation.
- *StudioCentral Library Installation and Configuration Guide (007-3319-*nnn*)* describes how to install and configure a basic, advanced, or custom StudioCentral Library system.

- *StudioCentral Library User's Guide for Windows NT and Windows 95 Clients* (007-3621-*nnn*) contains information on how to use the StudioCentral Library Windows NT® and Windows® 95 clients to create and manage digital assets.
- *StudioCentral Library User's Guide for Macintosh Clients* (007-3989-*nnn*) contains information on how to use the StudioCentral Library Apple® Macintosh® clients to create and manage digital assets.
- *StudioCentral Library Concepts* (007-3974-*nnn*) describes the basic StudioCentral Library concepts, such as how assets are stored and how the Migration Manager transfers assets.
- *StudioCentral Library Administrator's Guide* (007-3623-*nnn*) contains information on tuning the database; operating the server; troubleshooting problems; administering multimedia assets, users and groups, and Datamodels; and using the Web-based Admin Tool and command-line utilities.
- *StudioCentral Developer's Guide* (007-3246-*nnn*) is for software engineers who develop end-user applications, services, and plug-ins that extend the system.

Typographical Conventions Used in This Guide

The following type and symbol conventions are used in this guide:

<i>Italics</i>	Used for filenames, glossary entries (online, these show up as underlined), document titles, new terms, onscreen button names, tools, utilities
Bold	Used for Datamodel and attribute names
Ellipsis (...)	Indicates that the preceding is repeated
<i>nnn</i>	Placeholder for the version number in the part number: 007-1234- <i>nnn</i>
Right angle brackets (>)	Indicate a path through menus to a menu option. For example, "File > Open" means "Under the File menu, choose the Open option."
StudioCentral	Refers to the StudioCentral Library product.
Windows	Refers to a personal computer that runs the Windows NT or Windows 95 operating system. Where there is a distinction, the pertinent product name is used.

Following are the differences between the online and print versions of this document:

- Cross-reference links appear as quoted strings on paper and as colored text when viewed onscreen.
- At the beginning of each chapter is a list of topics discussed in that chapter. If you are viewing this book online, you can jump to the discussion of any of the listed topics by clicking an item in the list.
- Glossary entries appear in italics in print and as underlined links in the online document. If you view the document online, you can display the glossary definition by clicking the underlined link.

Important Terms in This Book

You will find a glossary in the back of this book. To save you from having to look up many words as you first start reading, this section contains a partial glossary of some of the more important terms.

StudioCentral Terminology

The following terms refer to StudioCentral features and functionality:

Asset	A representation of data stored in StudioCentral. An asset consists of two parts: <i>metadata</i> (descriptive information) and <i>content</i> (actual digital data). Assets can contain references to other assets, which allows relationships between assets to be stored.
Content	The actual multimedia data: digital images, sound files, and so on. Content is stored separately from metadata.
Datamodel	StudioCentral feature that is used for structuring metadata, which is an asset's descriptive information. Datamodels organize and define an asset's metadata and its relationship to other assets.
Metadata	Descriptive information about an asset, such as the compression format of a video asset. Each piece of information is called an <i>attribute</i> . Metadata is stored separately from content.

Repository Storehouse for digital assets. StudioCentral uses a database to store the metadata and at least one type of storage for the content. The combination of the database and the content storage is considered the repository.

StudioCentral Library User Terminology

The following terms refer to persons who use or administer StudioCentral Library:

Administrator Individual who supports end users by performing tasks such as defining Datamodels, starting and stopping servers, giving users access to assets, and so on.

Developer Software engineer who uses one of the StudioCentral Library application programming interfaces to write end-user application programs, services, and plug-ins, which extend the StudioCentral Library system.

End user User of StudioCentral Library application programs, such as Macintosh or Windows NT clients

Functional Overview

This chapter contains a functional overview of StudioCentral Library (called StudioCentral here), which is a digital asset management system that is used to store, manage, retrieve, preview, and distribute multimedia data. This data includes video clips, movie frames, sound files, animation cells, presentation storyboards, engineering designs, documentation, or any type of multimedia data that can be stored in a digital format.

StudioCentral is a complete solution that provides end users with desktop tools for managing their digital assets. The tools can be used with computers running Windows NT or Windows 95, Macintosh computers, SGI workstations, and Web browsers, in installations that range from small workgroups to enterprise-wide installations.

Digital assets stored in StudioCentral consist of:

- *Content*, which is the actual multimedia data, such as the graphic, digital video, or digital audio. Content can be stored in any format desired, such as JPEG or MPEG.
- *Metadata*, which is descriptive information about the multimedia data. The metadata consists of attributes that describe the asset (for example, its name and format) and its relationship to other assets.

Note: Administrators and application developers can customize the metadata attributes stored with an asset by defining StudioCentral Datamodels.

By separating the descriptive information from the content, each can be stored in a way that optimizes its access. For example, the metadata is stored in industry-standard databases, letting users search the descriptive information for assets that match certain criteria without having to read the content. And, the content can be stored in a variety of devices, including archive storage such as tape, enabling efficient use of the more expensive online storage.

Metadata and content are stored in a StudioCentral Asset Manager Server (AMS). Within the same AMS, multiple versions of the same asset can be stored, letting users track revisions that are made to the asset.

All assets in the Asset Manager Server have metadata associated with them, but not all assets have content. For example, an asset that is used to group assets that contain different digital formats of the same video content, may not have content itself. However, the assets that it groups, or points to, does have content.

The content for an asset is brought into the StudioCentral server by copying it from external files or transferring it from another StudioCentral server. When the content is checked into (that is, stored in) StudioCentral, it is cataloged. During this cataloging, or *asset typing*, process, if the content file's digital format is one of the types recognized by StudioCentral, the following steps are performed:

- The content file's header is read.
- Information about the asset is extracted from the content file and used to populate the asset's metadata. The attributes that are used are based on the type of the asset. For example, when a video asset is cataloged, the duration, digital format, bit rate, and image size are stored in metadata attributes.
- If appropriate, a thumbnail, which is a pictorial representation of the asset, is generated. For example, a thumbnail is generated for movie assets but not for audio assets.
- The metadata and content are stored in StudioCentral.

When assets are checked in, the end user or StudioCentral administrator can use graphical tools or command-line utilities to enter additional metadata attribute values, in addition to the ones that are automatically generated during the asset typing process. For example, the user may want to enter notes for the editor who will be reviewing an asset when it is completed, or a description of a new version of an existing asset.

Once assets are stored in a StudioCentral Asset Manager Server, they can be:

- *Queried*, where users search for assets. StudioCentral provides the Asset Query Language (AQL) that lets users search for assets based on criteria that they specify. For example, a user can search for all assets checked in by Joe Smith since July 9.
- *Browsed*, where users scan lists of assets on their desktops. For example, a user can perform a query that returns several assets, and then browse the returned assets before deciding which content to view.
- *Viewed*, where the asset's content is viewed on the desktop. For example, an asset containing a movie can be viewed by having SGI MediaBase stream the content to the desktop.
- *Checked out* (that is, reserved) or *retrieved* (that is, not reserved) by another user. This enables sharing of assets by members of a work group.
- *Sent to a video server for playout*. For example, assets can be transferred to a video server, such as one created with SGI's Video Server Toolkit, for real-time, frame-accurate playback of broadcast-quality digital media data.
- *Transferred to another StudioCentral system*. For example, a central Asset Manager Server (AMS) may store all assets managed by a news department. Individual assets can then be transferred from that central AMS to one in a regional office.
- *Moved to an archive system*, in either near-line or offline storage, which frees the online storage for the content of frequently used assets.

This functionality is shown in Figure 1-1.

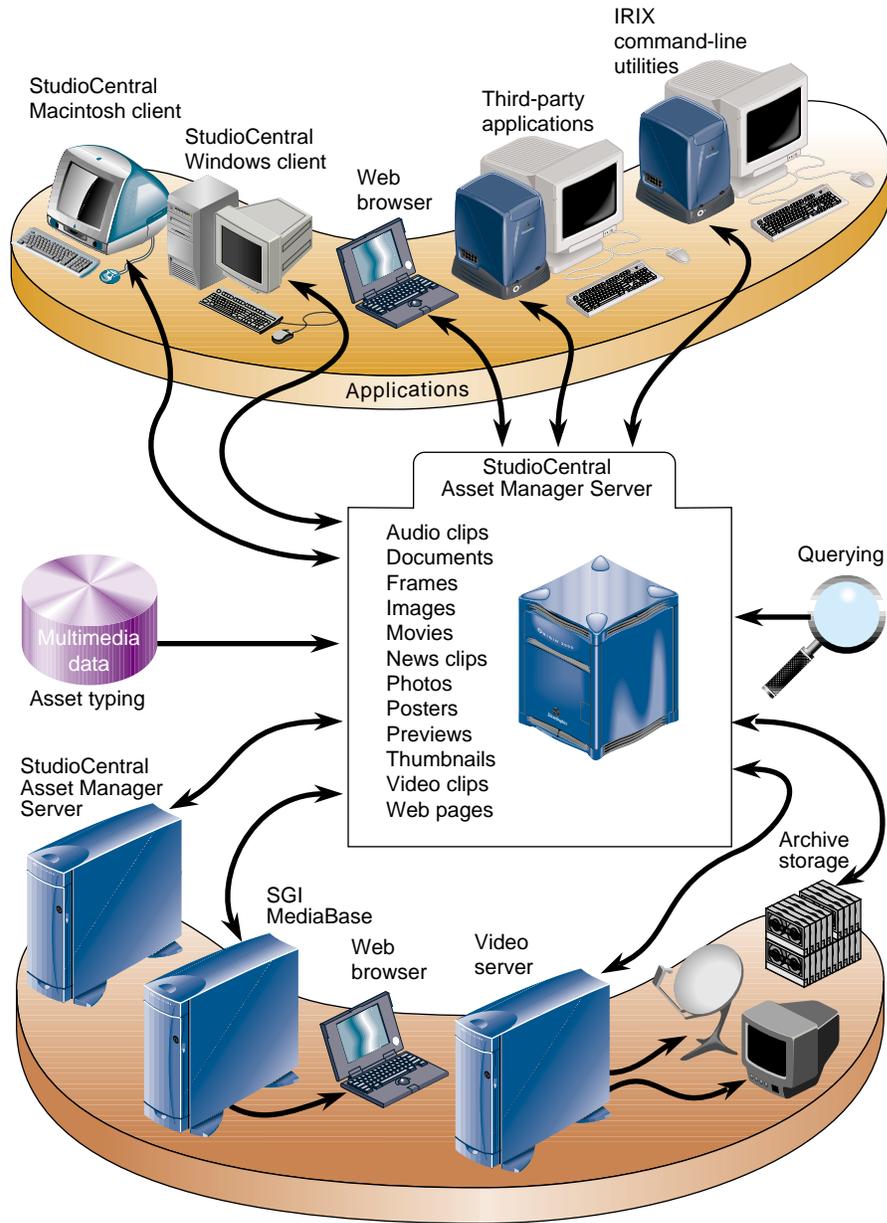


Figure 1-1 Functional Overview

The following applications are available with StudioCentral:

- *StudioCentral Windows NT and Windows 95 Client* and *StudioCentral Macintosh Client*, which run on Microsoft Windows and Apple Macintosh computers, respectively. These clients store, query, view, and manage digital assets. Chapter 2, “StudioCentral Applications,” has an example of each of these applications.
- *Admin Tool*, which runs in a Web browser, performs administration tasks such as transferring assets, registering Datamodels, and setting up and administering security. Chapter 2, “StudioCentral Applications,” has examples of this tool.
- *Third-party applications*, developed by SGI partners, perform tasks such as cataloging, archiving, and browsing assets.

Note: StudioCentral also has command-line utilities, which are documented in the *StudioCentral Library Administrator’s Guide*, that can be used to perform most of the tasks available through the Windows and Macintosh clients and the Admin Tool.

In addition to these applications, software engineers can use StudioCentral services or the StudioCentral application programming interface to develop applications that are tailored to specific needs. (See “Developing New StudioCentral Components” on page 32 for more information.)

StudioCentral Applications

This chapter describes the StudioCentral client applications, which store, query, view, and manage assets, and the Admin Tool, which is a Web-based application to administer StudioCentral repositories, users, Datamodels, and assets:

- “StudioCentral Clients” on page 7
- “Admin Tool” on page 19

StudioCentral Clients

The StudioCentral clients are available for Windows NT, Windows 95, and Macintosh platforms. These clients allow end users to store, retrieve, query, and manage StudioCentral assets using their native desktop environments. The clients can be used in a distributed environment in which assets are shared by workgroup members.

The StudioCentral clients are integrated with other native desktop applications, allowing assets to be dragged and dropped between StudioCentral and other applications such as Adobe Photoshop. Digital assets can also be stored in StudioCentral by having it monitor a specific directory so that new or modified files in that directory are automatically brought into StudioCentral. For example, a user working in Photoshop can save files into the directory that the StudioCentral Checkin Service monitors. Each time the user stores a new image in the Photoshop directory, the image is automatically stored in StudioCentral, enabling the user to check in assets without leaving the Photoshop application.

Regardless of how assets are brought in, the StudioCentral asset typing facility can catalog them automatically, storing metadata attributes that are appropriate for each asset type.

Once assets are stored in StudioCentral, the client can be used to search for assets that match specified criteria and then to view the assets on the desktop. For example, the user might want to search for and view all Photoshop assets created by a specific user on a given date.

The following sections present a scenario to illustrate how end users would store and retrieve assets using the StudioCentral clients. The same scenario is used for each client:

- “StudioCentral Windows NT Client Scenario” on page 8
- “StudioCentral Macintosh Client Scenario” on page 14

Note: The Windows NT client is used for one of the scenarios, but the Windows 95 client could be used in exactly the same way.

StudioCentral Windows NT Client Scenario

The following scenario illustrates how the StudioCentral Windows NT client helps an artist manage the creation of a new set design for a television show. The artist needs to create an image and have an editor review it. The artist wants to be able to save several versions of the image and wants to guarantee the work against loss:

- The artist obtains his specifications and logs in to his Windows NT desktop.
- He uses Windows Explorer to create a new directory for his local files. He then uses Adobe Photoshop to create his first image, storing it in the new directory.
- After creating his Photoshop file, the artist is ready to check it in to StudioCentral and have his editor review it. He starts StudioCentral by double-clicking the StudioCentral icon on his desktop, and then he logs in with his StudioCentral username and password. After he is logged in, the StudioCentral Explorer window is displayed, as shown in Figure 2-1.

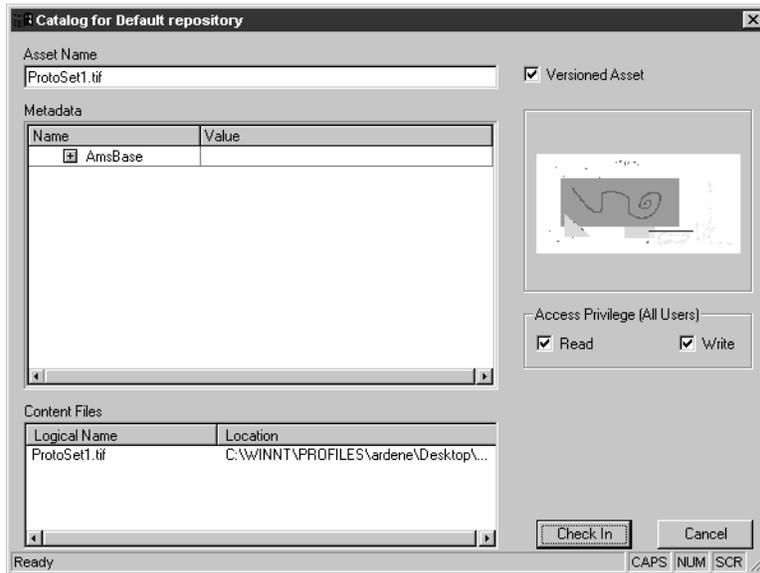


Figure 2-2 Windows NT Client: Catalog Window

- The Catalog window lets the artist enter information for the asset. He clicks the AmsBase Datamodel to display the Keywords field, and enters the keyword for the project, Acme, in the text field. Then, he clicks the *Check In* button.

Note: In addition to the information the artist enters during checkin, StudioCentral reads the file and generates metadata attributes that are appropriate for the type of content it contains.

- After the asset is checked in, the artist tells his editor that it is in StudioCentral. The editor can now review his work.

The editor uses her Windows NT workstation to find and review the artist’s work:

- The editor uses the Find window to look for the asset. Figure 2-3 shows the Find window.

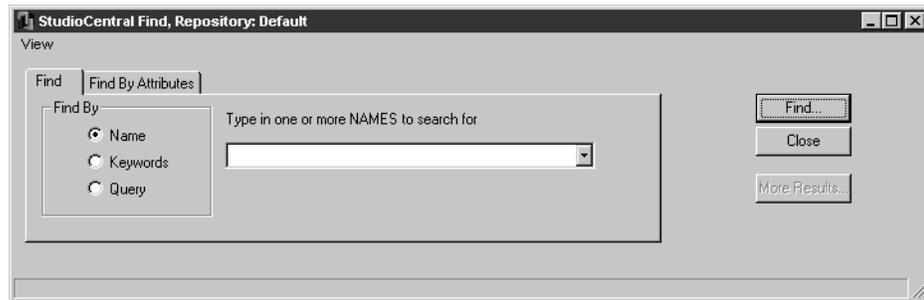


Figure 2-3 Windows NT Client: Find Window

- The editor clicks the Keyword radio button and enters the keyword Acme in the text box. After the editor clicks the *Find* button, the results of the Find are appended to the Find window, as shown in Figure 2-4.

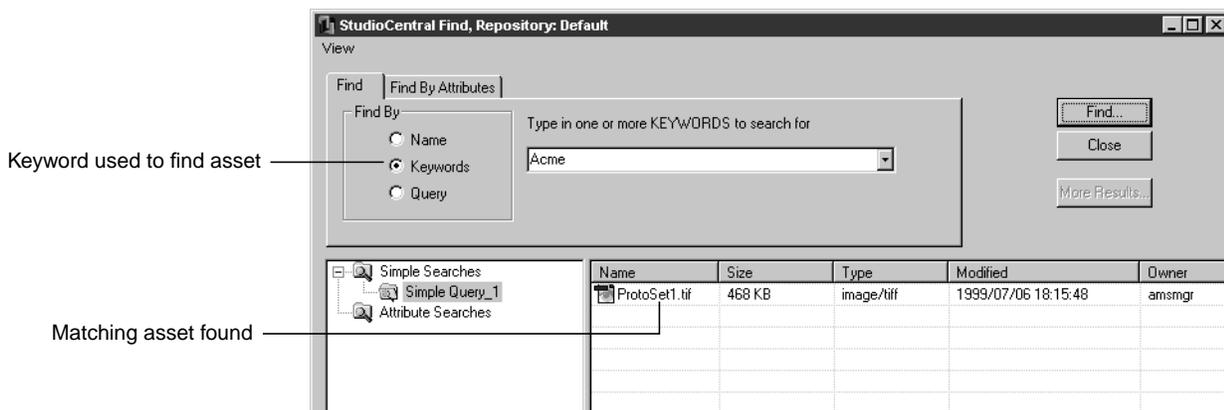


Figure 2-4 Windows NT Client: Find Results Display

- The editor selects the asset and right clicks to invoke the popup menu for asset operations. She selects the Show Info menu item to display information for the asset. Figure 2-5 shows the Information window.

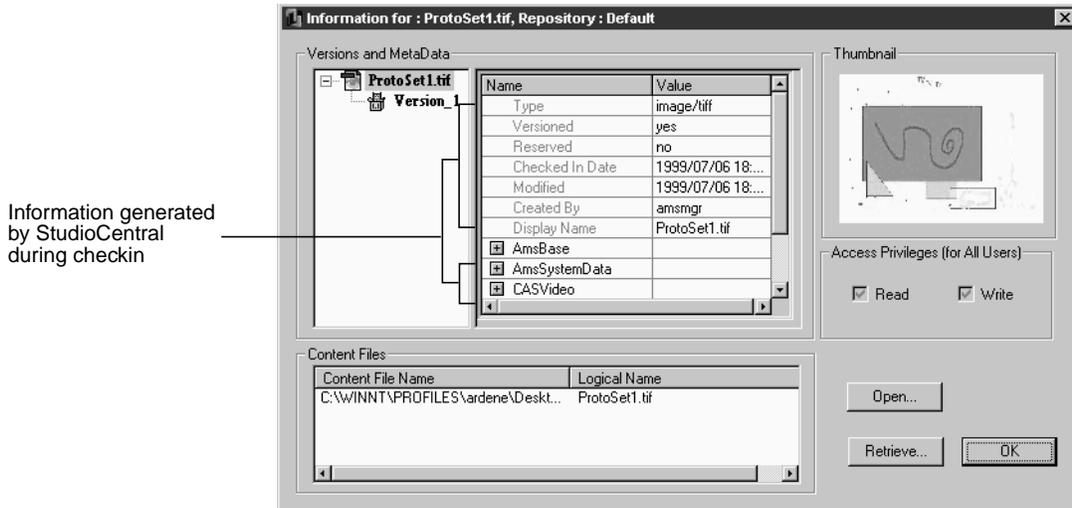


Figure 2-5 Windows NT Client: Information Window

- The editor wishes to make a change to the asset, so she retrieves it by clicking the *Retrieve* button in the Information window and saving the file on her Windows NT desktop.
- The editor uses her Photoshop application to adjust the colors. After making her changes, she checks the asset back in to StudioCentral, adding comments in the Notes field of the AmsBase Datamodel about the changes she made and how she wants the artist to proceed. Checking in the asset again creates a new version of the asset. The asset now has two versions as shown in Figure 2-6

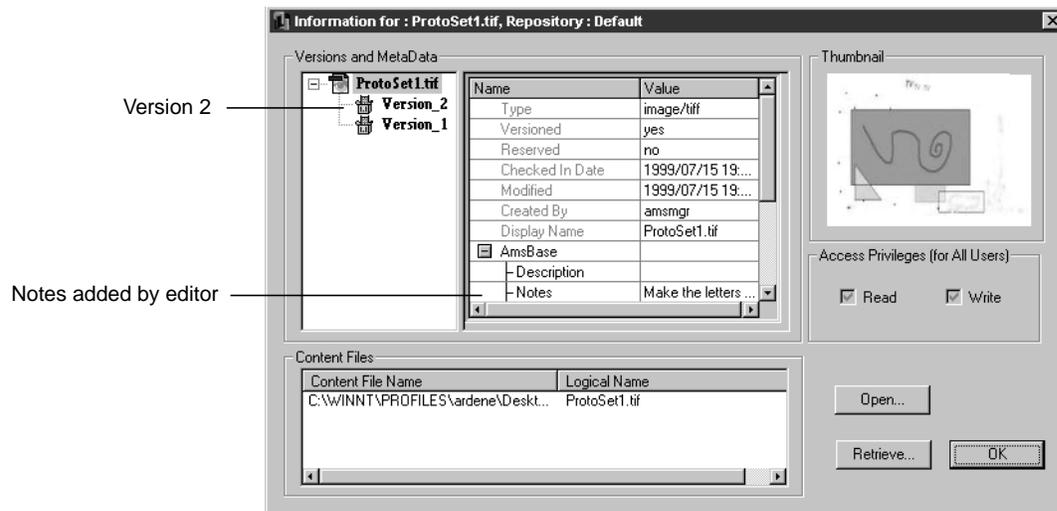


Figure 2-6 Windows Client: Second Version of Asset in Information Window

- Now the artist can read the editor's notes, retrieve the new version by clicking the Retrieve button, and make his changes in Photoshop.

The *StudioCentral Library User's Guide for Windows NT and Windows 95 Clients* describes the StudioCentral Windows client in detail.

StudioCentral Macintosh Client Scenario

The following scenario illustrates how the StudioCentral Macintosh client helps an artist manage the creation of a new set design for a television show. The artist needs to create an image and have an editor review it. The artist wants to have several versions of the image and wants to guarantee the work against loss:

- The artist obtains his specifications and creates a new folder on his Macintosh for his local files. He then uses Adobe Photoshop to create his first image, storing it in the new folder.
- After creating his Photoshop file, the artist is ready to check it in to StudioCentral and have his editor review it. He starts StudioCentral by double-clicking the StudioCentral icon on his desktop, and then he logs in with his StudioCentral username and password. After he is logged in, the StudioCentral Asset Browser window is displayed, as shown in Figure 2-7.

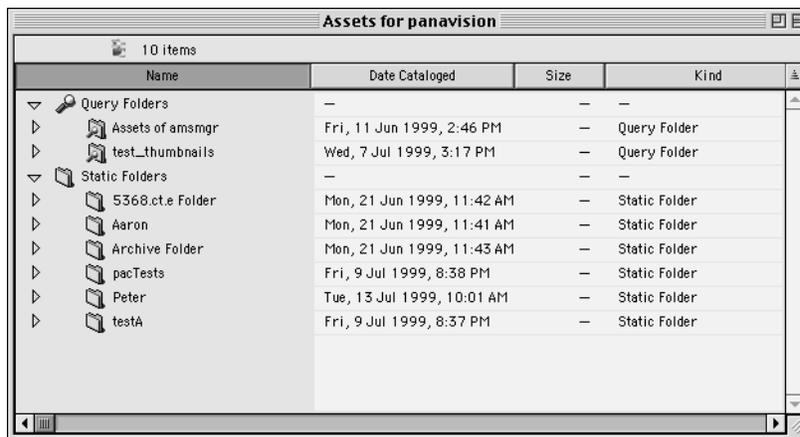


Figure 2-7 Macintosh Client: Asset Browser

- From his Macintosh folder, the artist drags the file he created and drops it into the StudioCentral Asset Browser window. After he drops the file, StudioCentral displays the Check In dialog, which is shown in Figure 2-8.

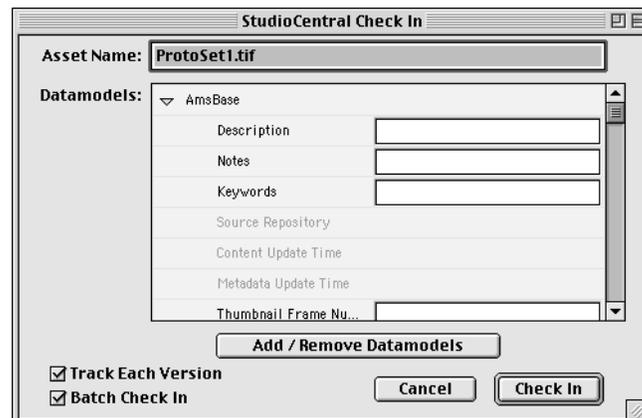


Figure 2-8 Macintosh Client: Check In Window

- The Check In window lets the artist enter information for the asset. He adds the keyword for the project, which is Acme, and then he clicks the *Check In* button.

Note: In addition to the information the artist enters during checkin, StudioCentral reads the file and generates metadata attributes that are appropriate for the type of content in the asset.

- After the file is checked in, the artist tells his editor that the asset is in StudioCentral. The editor can now review his work.

The editor uses her Macintosh to look for and review the artist's work.

- The editor uses the Find window to look for the asset. Figure 2-9 shows the Find window.

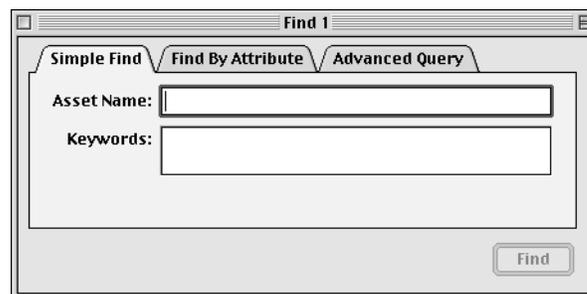


Figure 2-9 Macintosh Client: Find Window

- The editor enters the keyword for the project in the Keywords text box and clicks the *Find* button. The results of the search are displayed in a results window, as shown in Figure 2-10.

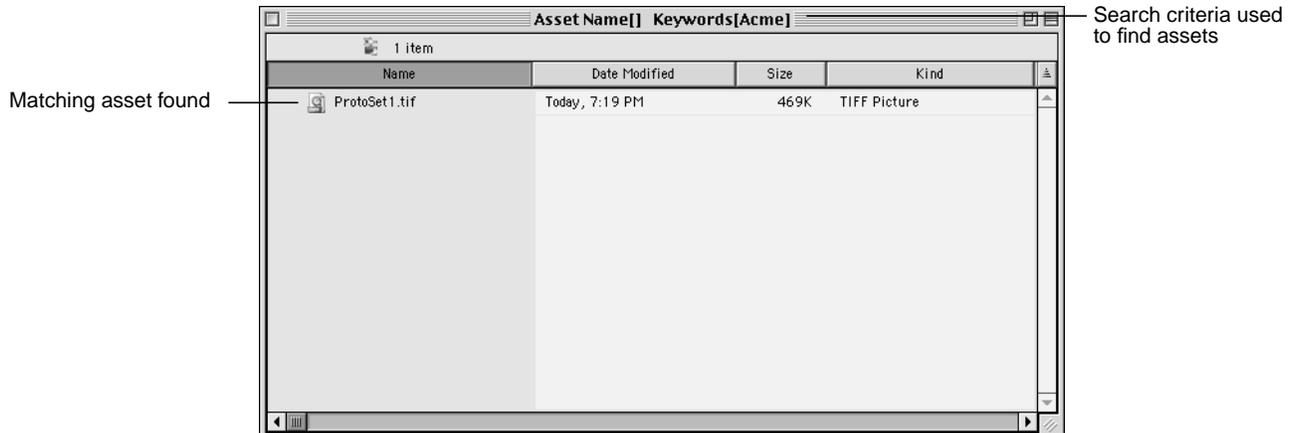


Figure 2-10 Macintosh Client: Find Results Window

- The editor selects the asset and chooses File > Get Info to look at the information for the asset. Figure 2-11 shows the Information window, which includes the information that was entered manually and that was generated by StudioCentral during the checkin process.

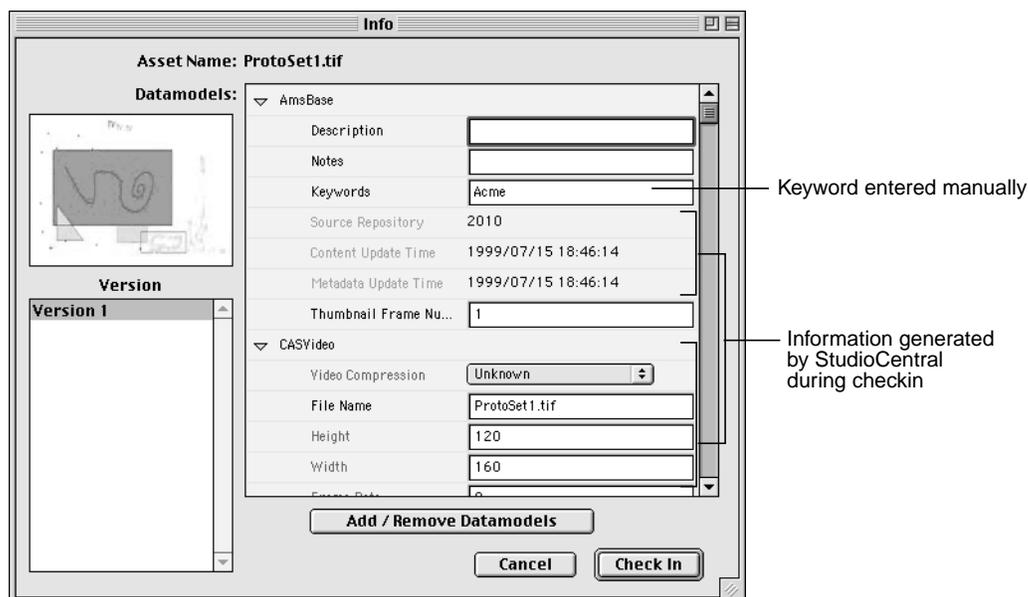


Figure 2-11 Macintosh Client: Information Window

- The editor wishes to make a change to the asset, so she retrieves it by dragging Version 1 from the Version area of the Info window to her Macintosh desktop. Then she clicks the *Cancel* button to close the Info window.
- The editor uses her Photoshop application to adjust the colors. After making her changes, she checks the asset back in to StudioCentral, adding comments in the notes field about the changes she made and how she wants the artist to proceed. Checking in the asset again creates a new version of the asset. The asset now has two versions as shown in Figure 2-12.

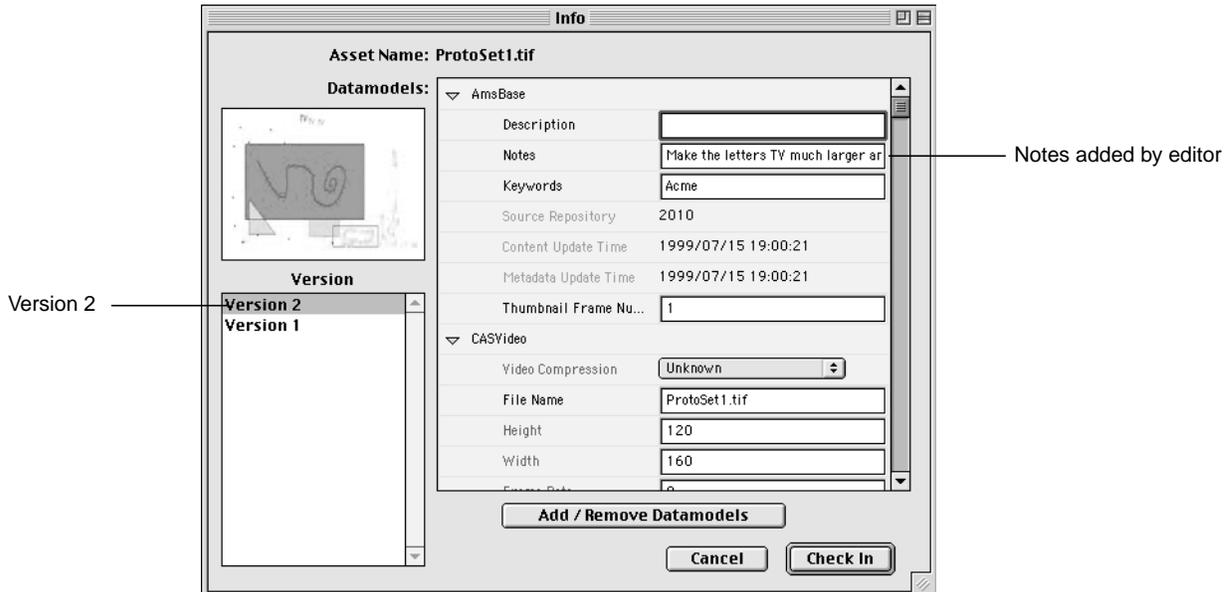


Figure 2-12 Macintosh Client: Second Version of Asset in Information Window

- Now the artist can read the editor's notes, retrieve the new version by dragging and dropping Version 2 from the Info window to his computer, and make his changes in Photoshop.

The *StudioCentral Library User's Guide for Macintosh Clients* describes the StudioCentral Macintosh client in detail.

Admin Tool

The Admin Tool, which runs in a Web browser, performs repository and asset administration tasks. For example, the tool is used by administrators to give users access to StudioCentral assets, transfer assets, and check the repository for consistency.

Figure 2-13 shows how the tool is used to add a user to a group. (A “group” is useful when several users have the same privileges in StudioCentral because it lets the administrator assign the privileges to the group instead of to each user individually.) In this case, the user “hsmith” is being added to the “production” group.



The screenshot shows a web form titled "StudioCentral Add User to Group". Below the title is a horizontal line and the instruction "Please Fill In / Verify Required Parameters". There are three input fields: "User Name:" with the value "hsmith", "Group Name:" with the value "production", and a "Submit" button. The text "Add User to Group:" is positioned to the left of the Submit button.

Figure 2-13 Admin Tool: Adding a User to a Group

The tool can also be used to define new users and groups, give users and groups permission to access certain assets, and remove users and groups. For example, Figure 2-14 shows how the user “hsmith” receives permission to read the second version of an asset whose name is “brandNew2.” (“@@2” indicates the version number.) In this case, because the permission is given only to the individual user and not to the group, other users in the group cannot read the asset unless permission to do so is given to those users individually or to the group as a whole.



Figure 2-14 Admin Tool: Setting Permissions for an Asset

The Admin Tool can search StudioCentral for assets that match criteria that are either predefined or entered by the user. For example, Figure 2-15 shows how to search for assets whose content is at least 10 gigabytes (that is, 10,000,000,000 bytes). If this search is performed frequently, the user can easily add it to the Admin Tool as a predefined search. This lets the user perform the search again, without having to enter the criteria.

When the search in Figure 2-15 is performed, a table that identifies the assets that satisfy the search criteria is displayed. The table displays the name of each asset as a hyperlink, and the size of each asset's content (content_length). Clicking the hyperlink displays the metadata for the asset.

S t u d i o C e n t r a l
Find Assets
(User:amsmgr, Repository:Default)

Attribute Name		Attribute Value
content_length	=	1000000000
and or		
	=	

Number of Report Attributes: 3

Attribute Names

version_name

content_length

Find All Versions

Figure 2-15 Admin Tool: Searching for Assets

The Admin Tool can also start, stop, and determine the status of services that run on the Asset Manager Server. For example, Figure 2-16 shows the page that displays the status of services. For each service, a colored bullet to the left of the service name indicates its status. The color of the bullet indicates whether the associated service is up and running, running but problems have been detected, or not running.

Status	Service	Messages
OCS Services		
	Three Tier Metadata Server	Ping succeeded
	Three Tier Metadata Broker	Ping succeeded
	StudioCentral Migration Manager	StudioCentral/AssetServices: not found
	Content Server: DefaultContentStore	Ping succeeded
	Content Server: flotsom_mbase	Ping succeeded
Individually-Checked Services		
	Name Server	Service is running.
	Informix Database	Service is not running.
	Oracle Database	Service is not running.
	StudioCentral Purge Daemon	Service is not running.

Key:



Service is running
No problems detected



Service is running
Problems detected



Service not running



Figure 2-16 Admin Tool: Determining Status of Services

The Admin Tool is based on industry-standard practice for Web browsers and Web servers, and is easily modified and customized. It is described in detail in the *StudioCentral Library Administrator's Guide*.

Software Components

This chapter describes the StudioCentral Library software components and the ways in which they can be used to develop new applications:

- “Software Components” on page 23
 - “Core Software” on page 25
 - “Application Programming Interface” on page 29
 - “Asset Services” on page 31
- “Developing New StudioCentral Components” on page 32
 - “Client Applications” on page 32
 - “Content Servers” on page 33

Software Components

The focal point of the StudioCentral software is the Asset Manager Server, which contains the following software components:

- *Core software*, which provides the basic functionality that enables assets to be stored in, and retrieved from, StudioCentral
- *Application programming interface (API)*, which provides the interfaces used by software engineers to develop end-user applications, services, and plug-ins written in the C, C++, PERL, SC/PERL, Java, and ATS programming languages
- *Asset Services*, which provide a service to either applications or other StudioCentral services

Figure 3-1 shows the primary software components that make up the StudioCentral software.

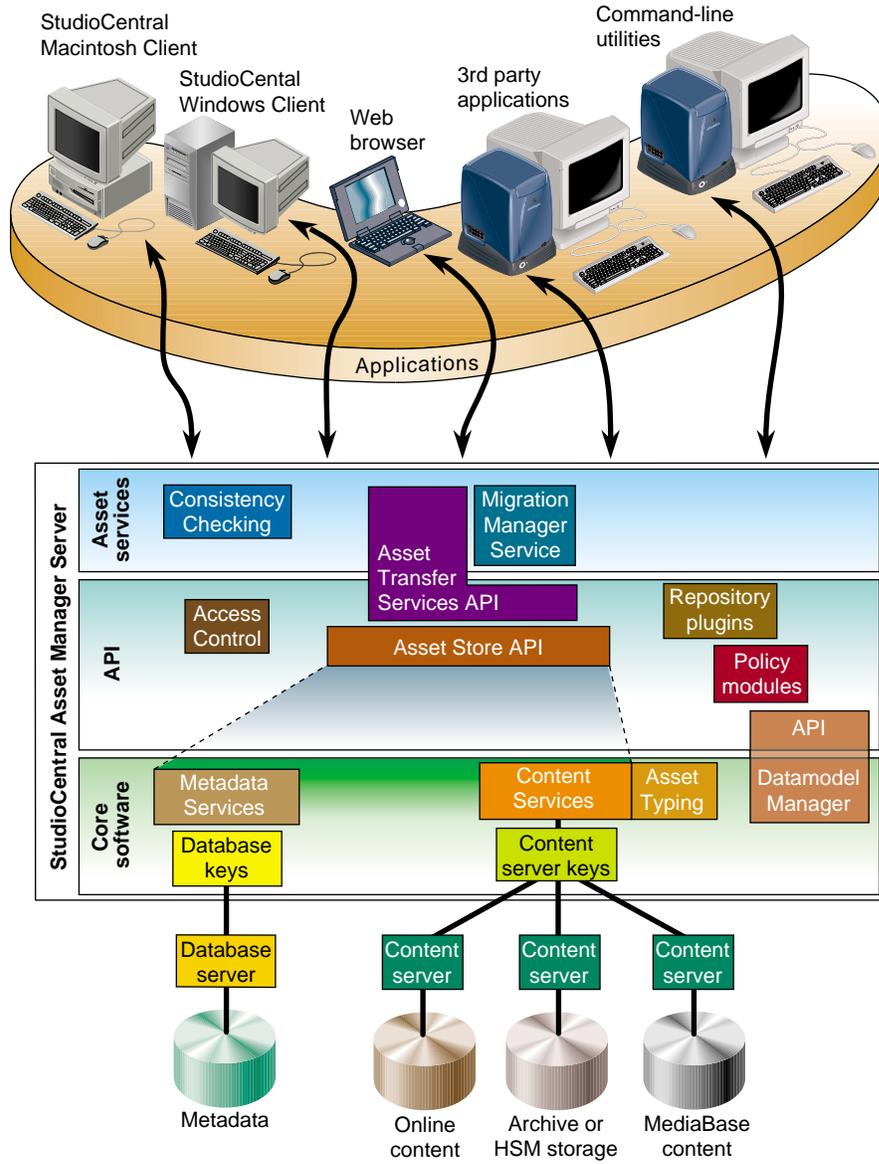


Figure 3-1 StudioCentral Software Components

The remainder of this section describes the components shown in Figure 3-1:

- “Core Software” on page 25
- “Application Programming Interface” on page 29
- “Asset Services” on page 31

Core Software

This subsection describes the core software, which provides the basic functionality for the StudioCentral system:

- “Metadata Services” on page 25
- “Content Services” on page 26
- “Database Keys” on page 28
- “Content Server Keys” on page 28
- “Asset Typing” on page 28
- “Datamodel Manager” on page 29
- “Backup and Restore Services” on page 29

Note: The combination of Metadata Services and Content Services provides the primary functionality of the StudioCentral software. Together they manage the StudioCentral repository, which consists of the metadata and content for all assets stored in the Asset Manager Server.

Metadata Services

Metadata Services supports the storing, retrieving, and managing of metadata. Among other functions, Metadata Services provides the mappings that enable StudioCentral metadata to be stored in, and retrieved from, industry-standard database management systems.

Metadata Services is independent of the specific database used. The database-dependent code exists in a software component called a *database key*, which is an interface between Metadata Services and the actual database management system. These database keys enable each application to take advantage of features specific to a given database.

StudioCentral currently has database keys for the following:

- Informix Online Dynamic Server (ODS)
- Informix ODS with Universal Data Option
- Oracle Server

Content Services

Content Services, which enables asset content to be typed, stored, retrieved, managed, and archived supports the storage of content in the following:

- *Online storage*, which provides the fastest access to asset content and is normally used for currently active assets. Online storage is the most expensive type of content storage and is used primarily for frequently accessed content.

Online storage is provided by the XFS filesystem.

- *Hierarchical storage*, which enables assets to be moved between online, near-line, or offline (archive) storage, either automatically based on use or manually at the user's request. Content stored in near-line storage can be retrieved much faster than content stored in offline storage. In fact, offline storage may be at a location that is physically removed from the StudioCentral repository.

Hierarchical storage is provided by the Data Migration Facility (DMF) system developed by SGI. It provides access to media content by quickly bringing offline content onto a disk cache for fast access.

- *Archive storage*, which enables online storage to be freed for just the content that is currently active or most likely to be accessed.

For archive storage, StudioCentral supports the Archive Management Storage System (AMASS) technology from EMASS and the Avalon Archive Manager.

- *SGI MediaBase*, which allows assets stored in StudioCentral to be streamed to end-user desktops.

Note: System integrators and application developers can write content servers for other storage devices or storage management systems. See the *StudioCentral Developer's Guide* for information.

As with Metadata Services, Content Services is independent of the specific content storage that is used. The content storage-dependent code exists in software components called *content server keys*, which are interfaces between Content Services and the content servers that manage the actual content stores. These content server keys, which are analogous to the Metadata Services database keys, enable all StudioCentral applications to be independent of the storage used. They also let each application take advantage of features specific to a given storage device.

For example, there is a key for a Filesystem Content Server, which manages content stored in online filesystems, and a key for a DMF Content Server, which manages content stored in SGI Data Migration Facility hierarchical storage management systems. An Asset Manager Server usually has keys for a Filesystem Content Server for online storage, and zero or more other content servers, depending on application needs. For example, a broadcast news department might have:

- an Informix database for storing metadata
- a Filesystem Content Server for managing online content
- an AMASS Content Server for managing archive storage
- an SGI MediaBase Content Server for streaming videos to Web browsers

Metadata that describes the archived footage can be kept online in the StudioCentral Informix database, and the low-resolution versions of the assets that have a high likelihood of being reused can be kept online in the XFS filesystem. (The low-resolution versions take significantly less storage than the high-resolution versions, making it feasible to keep them online.) This allows users to query the metadata and view low-resolution versions of the footage, before deciding whether to retrieve the high-resolution footage from archive storage.

The high-resolution content of all the footage that has passed through the facility can be stored on tape in the archive storage, making it an invaluable resource for the newsroom in the creation of new stories. Its use is illustrated by the following:

- A user checks an asset into StudioCentral. The Informix database server stores the metadata in the database and the Filesystem Content Server stores the content in the online filesystem it manages. This content is a high-resolution format that is suitable for playback by a broadcast system.
- The user checks in a low-resolution format of the same content, which is suitable for playback in a Web browser. The user specifies that the low-resolution content be stored by the MediaBase Content Server so that it can be streamed through the SGI MediaBase application into Web browsers.

- To free online storage, the user archives the high-resolution content, which takes significantly more storage than the low-resolution content. This is done by the AMASS Content Server. The low-resolution content is kept in the online storage and the asset's metadata is kept in the metadata database. This lets the user query the asset and view the low-resolution content in a Web browser.
- The user performs a query that returns the asset.
- The user views the low-resolution version of the asset in a Web browser, which causes the content to be streamed by SGI MediaBase.
- The user decides to use this asset in a broadcast program but wants to edit it first. Because the user expects to access the high-resolution content frequently during the editing phase, he requests that the content be transferred from archive storage to online storage. This transfer is done by the content servers for the archive storage and the online (filesystem) storage.

Note: The user can access the high-resolution content directly from the archive storage. However, this is a much slower operation and is recommended only for infrequently accessed content.

Database Keys

Database keys allow StudioCentral clients to access metadata that is stored in different kinds of databases.

Content Server Keys

Content server keys allow StudioCentral clients to access digital data that is stored in different kinds of devices and storage management systems.

Asset Typing

Asset Typing automatically generates metadata attributes and, if appropriate, a *thumbnail* when an asset with content is checked into StudioCentral. It works with Content Services when an asset is checked into StudioCentral.

Datamodel Manager

The Datamodel Manager is used to define and manage Datamodels, which enable metadata attributes to be added to assets and complex asset structures to be defined. For example, the Datamodel Manager lets you create an asset that contains separate tracks of an audio asset, where each track is, itself, a StudioCentral asset.

Backup and Restore Services

The Backup and Restore Services provide command-line utilities for taking and restoring a snapshot of the metadata and content in an Asset Manager Server. The Restore Service retains the database schema and restores the content files for all assets. The Backup and Restore Services can be used for content that is stored in a Filesystem Content Server only.

Application Programming Interface

This section discusses the StudioCentral application programming interface (API), which provides the interfaces that are used to develop applications, services, and plug-ins:

- “Asset Store API” on page 29
- “Datamodel Manager API” on page 30
- “Access Control” on page 30
- “Asset Transfer Service” on page 30
- “Repository Plug-ins” on page 30
- “Policy Modules” on page 31

Asset Store API

The Asset Store API makes available to the application developer the features that are provided by the Metadata and Content Services, which are part of the core software components. For example, Asset Store has C++ functions that are used by application developers to store assets in the Asset Manager Server and update the metadata for existing assets.

Datamodel Manager API

The Datamodel Manager API makes available to the application programmer the features provided by the Datamodel Manager.

Access Control

Access Control manages access to StudioCentral assets. For example, the command-line utilities that give users permissions to access assets, use the Access Control functions of the API.

Asset Transfer Service

The Asset Transfer Service (ATS) provides a TCP-based interface for storing and retrieving the metadata and content of assets stored in StudioCentral repositories. ATS provides a lightweight, simple protocol that can be used by client applications. For example, the StudioCentral Macintosh Client uses the Asset Transfer Service to store, retrieve, and manage assets stored in an Asset Manager Server. Also, ATS can be used with SGI's Video Server Toolkit (VST), which is used to develop applications for real-time, frame-accurate playback of broadcast-quality digital media data.

The ATS interface is very similar in architecture and implementation to the industry standard File Transfer Protocol (FTP). ATS, however, works on assets, which consist of metadata and content, while FTP works only on file content.

ATS can be used to develop applications and services that do not require the full functionality provided by the StudioCentral C++ API. For example, the Migration Manager Service uses a remote ATS to perform transfers between repositories.

Repository Plug-ins

Repository plug-ins invoke operations within the Asset Manager Server when certain events occur. For example, a plug-in can notify a user when a particular asset is checked in.

Repository plug-ins are written by system integrators and application developers to extend the StudioCentral functionality.

Policy Modules

Policy modules define the rules for determining in which content server(s) an asset's content should be stored. Policy modules can serve a variety of purposes, including load balancing, high availability, and administration of associated assets.

Policy modules are written by system integrators and application developers to customize StudioCentral.

Asset Services

Asset services, which provide a service to either applications or other services, consist of the following:

- "Migration Manager Service" on page 31
- "Consistency Checking Services" on page 31

Migration Manager Service

The Migration Manager Service performs background tasks that transfer assets between Asset Manager Servers, move content between content servers in the same Asset Manager Server, and delete assets. For example, the Migration Manager Service can transfer assets from a centralized Asset Manager Server in a broadcast network to regional stations or affiliates for local storage and playback.

Requests to the Migration Manager are issued using either command-line utilities or applications that invoke these commands. For example, the Admin Tool invokes the Migration Manager *transferAsset* command-line utility to transfer assets.

Consistency Checking Services

Consistency Checking searches for and fixes inconsistencies in the way assets are stored in the repository. For example, Consistency Checking can identify content files that are not pointed to by assets.

Developing New StudioCentral Components

StudioCentral provides interfaces for building applications and services that use StudioCentral, or for integrating existing applications into StudioCentral. Detailed information about how to use these interfaces is provided in the *StudioCentral Library Developer's Guide*. This section provides an overview of how to use the interfaces to develop the following components:

- “Client Applications” on page 32
- “Content Servers” on page 33

Client Applications

The tasks required to perform asset management in any client application are generally the same:

- Create and store asset contents and metadata.
- Retrieve asset content and metadata.
- Modify information describing the asset content.
- Delete asset content and its metadata.

Each of the application programming interfaces (APIs) described in this section enable you to create client applications that accomplish these tasks. Detailed information about how to use these interfaces to develop applications is provided in the *StudioCentral Library Developer's Guide*.

The interfaces for developing client applications are:

- “ATS” on page 33
- “C++” on page 33
- “Command-Line Utilities” on page 33
- “SC/Perl” on page 33

ATS

The Asset Transfer Service (ATS), similar in nature to the File Transfer Protocol (FTP), gives developers a direct means of managing and querying assets. ATS can be used at the command line, or incorporated into Perl scripts, java, C++, or applications in most any other computer language to create a client application. ATS is the API of choice for the future.

C++

The C++ API enables C++ programmers to create a client application that has low-level control of asset management features.

Command-Line Utilities

The StudioCentral command-line utilities are executable, ready-to-use programs that perform a variety of operations on assets. The utilities are based on and provide a higher level of functionality for the C++ API. You can use these utilities for immediate action and feedback, or for the creation of a client application.

SC/Perl

The SC/Perl Module allows Perl programmers to directly access the StudioCentral API functions and objects using Perl language constructs. SC/Perl is essentially composed of *sc.pm* (the Perl Module), and *sc.so* (a shared object/library).

The SC/Perl Module covers an important subset of the overall SC API. This includes asset creation, checkin and checkout, asset queries, getting and setting attribute values, reading and writing content files, and basic access control (ACL) support.

Content Servers

StudioCentral provides C++ APIs for writing services that store and retrieve digital data files. After using the StudioCentral interfaces to implement a content server, the content server can be used to store and retrieve the content for assets that are checked in to a StudioCentral Asset Manager Server. Depending on the implementation, the content server can run on the same machine as the StudioCentral Asset Manager Server or on a remote machine that is accessible through the network.

Configurations

This chapter describes the StudioCentral Library configurations and the tasks required to install, configure, and support them:

- “Configuration Types” on page 35
- “StudioCentral Installation and Administration Tasks” on page 37

Configuration Types

StudioCentral can be used in a distributed configuration, where many users store assets in centralized repositories. This lets multiple users work on assets under the version control and security of StudioCentral.

This section describes the following StudioCentral configurations:

- “Single-Server Configurations” on page 35
- “Multiple-Server Configurations” on page 36

Single-Server Configurations

In the single-server configuration, which is shown in Figure 4-1, an Asset Manager Server (AMS) running the IRIX operating system serves as a centralized repository for a network that includes Windows and Macintosh computers, SGI workstations, and Web browsers. In this configuration, assets that are stored in the AMS can be accessed by users on the network. The content, archive storage, and database are all stored on the AMS, along with the StudioCentral software.

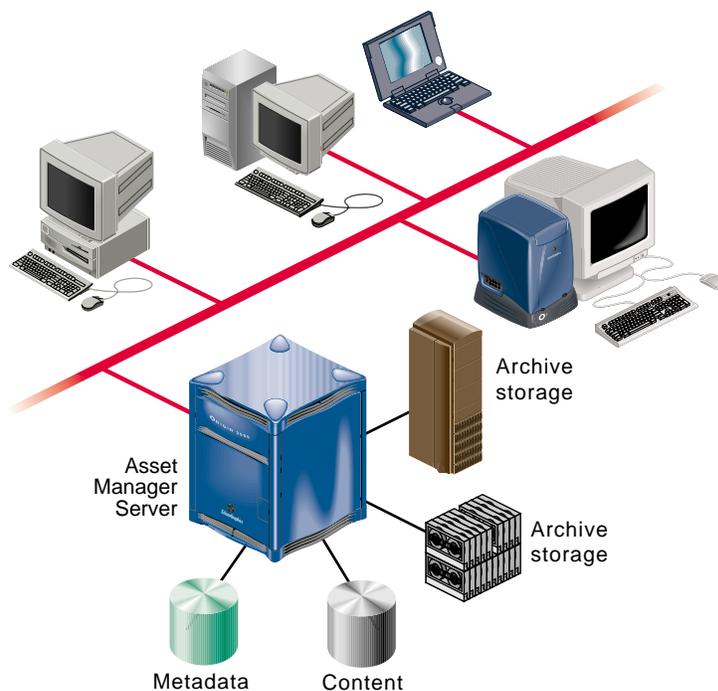


Figure 4-1 Single-Server Configuration

The StudioCentral Configuration Tool, which is part of the StudioCentral Admin Tool, is used to install and configure the Asset Manager Server configuration shown in Figure 4-1. It provides an easy-to-use interface that runs in a Web browser. (Using the tool to install and configuring a basic, single-server configuration is described in the *StudioCentral Library Installation and Configuration Guide*.)

Multiple-Server Configurations

StudioCentral has the ability to run in a fully distributed environment on multiple server systems. For example, the Asset Manager Server, metadata database, and content servers can be on different servers.

Installation and configuration of a multiple-server configuration requires a knowledgeable system administrator or integrator, who may need to manually install

and configure the StudioCentral services. (The *StudioCentral Library Installation and Configuration Guide* describes how to do this.)

StudioCentral Installation and Administration Tasks

This section lists the procedures required to install, configure, and administer StudioCentral. The section is organized by administrative function. In some environments, one individual may be responsible for all functions; in others, more than one individual may be responsible. When more than one individual is involved, it is important to coordinate the administrative tasks for the successful installation and operation of StudioCentral.

StudioCentral Administrator

The StudioCentral administrator is responsible for the overall installation and operation of the StudioCentral servers and clients. This individual performs the following tasks:

- installs, configures, and operates the StudioCentral servers (with help from the database administrator)
- installs and configures the StudioCentral clients (with help from the database administrator)
- creates IRIX user IDs
- updates and maintains client configuration files
- updates and maintains content server configuration files
- performs queries and data management on assets
- scans log files to troubleshoot problems
- starts and stops the StudioCentral servers
- adds and manages StudioCentral users and their security privileges

Database Administrator

The database administrator is responsible for the database management system that is used with StudioCentral. This individual is also responsible for helping the system administrator during the installation and configuration of StudioCentral. The following tasks are performed by the database administrator:

- helps install, configure, and maintain the Informix or Oracle database management system used by StudioCentral
- defines the initial database configuration
- uses a StudioCentral script to create the StudioCentral schema file
- manages database user IDs for the database(s) used in StudioCentral
- modifies and tunes the database for performance and reliability
- establishes and performs database backup and maintenance procedures for failure recovery of data

Glossary

Access Control

The API function that manages access to StudioCentral assets. For example, Access Control gives end users permission to access assets created by an administrator.

asset

Unit of storage in a *digital asset management system*. Each asset consists of descriptive information such as the title and duration; digital media data, if the asset has content; and an index, if required by the content format. For example, each movie, commercial, trailer, thumbnail, and so on, stored in an archive system is an asset. Assets may consist of only descriptive information if those assets are used to group other assets.

Asset Query Language (AQL)

Provides a way to search for assets using metadata attributes. The syntax for an AQL query resembles that of the database structured query language (SQL) select statement, except that AQL queries specify Datamodels and attributes instead of database tables and columns. See also *attribute* and *Datamodel*.

Asset Transfer Service (ATS)

Provides a TCP-based interface for storing and retrieving the metadata and content of assets stored in StudioCentral repositories. This interface provides a lightweight, simple protocol that can be used by client applications and video servers. ATS can be used to develop applications and services that do not require the full functionality provided by other components of the StudioCentral API.

asset typing

The process by which an asset's type is determined, its content file is read, and the following are automatically generated: metadata attributes based on the asset type and, if appropriate, a *thumbnail*. For example, for a video asset, the duration, bit rate, and size of its image are extracted from the content file and automatically stored in the asset's metadata when it is checked into StudioCentral. Asset typing is part of cataloging.

attribute

A single piece of an asset's metadata information. Metadata consists of system attributes, which are defined by StudioCentral and exist in all assets, and Datamodel attributes, which extend the attributes associated with an asset. For example, there is a system attribute that contains an asset's name. See also *Datamodel*.

cataloging

The process by which the information for an asset is determined and stored with the asset as metadata attributes. Cataloging occurs when an asset is first checked in or an existing asset is updated. Asset typing is part of cataloging at initial checkin of an asset.

check in

To add a new asset to StudioCentral, you must check it in. At check-in time, StudioCentral automatically assigns the asset a unique identifier (*OID*) in order to locate it in the repository, and secures it in the database. Afterwards, each time you check out or retrieve the asset (to change it) and check it back in (to secure your changes), StudioCentral creates another, most recent, version of the asset. See also *check out*, *repository*, *zeroth version*.

check out

To modify an asset, you can check it out of or retrieve it from the StudioCentral repository. When you check out an asset, StudioCentral gives you a "working copy" of the asset, and locks the "real" file to prohibit changes by other users until you check in your modifications. See also *retrieve*.

composite asset

An asset that points to, or contains, other assets, some of which may also be composite assets.

content

The data portion of an asset (for example, the image, scene, sound sequence, or song). StudioCentral stores an asset's content separately from its metadata. See also *asset*, *metadata*.

content server key

The software component that enables a content store to be used with StudioCentral. For example, the key for the Filesystem Content Server enables content files to be stored in XFS filesystems.

database key

The software component that enables a database management system to be used with StudioCentral. For example, StudioCentral has a database key for the Informix Online Dynamic Server.

Datamodel

A data structure that defines metadata attributes; it is used for modeling data. An asset can have more than one Datamodel associated with it. If you want to extend the metadata attributes of an asset, you can create a new Datamodel for it.

Datamodel template

StudioCentral's "stylesheet" for a Datamodel. It contains the definitions of all attributes within that Datamodel.

developer

Software engineer who writes applications using the StudioCentral application programming interface (API). See also *end user*.

digital asset management system

A system for storing, managing, and distributing digital multimedia data.

end user

End user of StudioCentral application programs.

metadata

A series of attributes that describe an asset. Metadata searches for, or queries, assets that meet specified criteria. For example, if you have an asset that contains one scene in a movie and its metadata specifies that the scene uses special sound effects, you find this asset when you query the database for all movie scenes that use special sound effects. To speed queries, StudioCentral stores metadata separately from content. See also *asset*, *content*, *query*.

multiformat

A type of asset that groups other assets, each of which has the same digital content but is encoded in a different digital format.

non-versioned

When you first check in an asset, you can check it in as a non-versioned, or unversioned, asset rather than a *versioned* asset. StudioCentral does not track subsequent revisions (*versions*) of a non-versioned asset. Therefore, even after revising it several times, there is only one version of it in StudioCentral. See also *version*, *zeroth version*.

OID

The unique identifier that StudioCentral assigns to each version of an asset when you check it in. OIDs comprise three numbers: the repository ID, a number that uniquely identifies each version of an asset, and a number that uniquely identifies each version tree. OIDs define asset structures in which one asset points to, or contains, another asset. See also *check in*, *version*, *version tree*.

persistent

Protected from removal from the repository.

pinning assets

Preventing assets from being deleted.

preview

A smaller version of an asset's content. For example, the preview may be a short segment or a lower-resolution version of the asset.

query

The process whereby users search a StudioCentral Asset Manager Server for assets that satisfy given criteria. For example, a user might search for all video assets that contain footage of tornadoes. Users perform queries by specifying values for metadata attributes.

queryable

A property of an *attribute* that lets it be used as a condition when searching for, or querying, assets. For example, if *owner_name* is a queryable attribute, you can search for assets that have an *owner_name* attribute whose value is "Smith." See also *query*.

repository

A storage area for digital assets. Each repository consists of one database, for storing metadata, and one or more content servers, for storing the actual digital data. See also *asset*.

retrieve

To modify an asset, you can retrieve it from or check it out of the StudioCentral repository. When you retrieve an asset, StudioCentral gives you a “working copy” of the asset. Other users are locked out from retrieving the asset only during the time you are cataloging or checking in the asset. See also *check out*.

root

The first version of a versioned asset.

SGI MediaBase

Media serving software from SGI for delivering high-quality MPEG-1, MPEG-2, and H.263 video and audio streams to Web clients using IP or native ATM networks.

thumbnail

A pictorial representation of the asset. For example, the thumbnail for a movie asset might be a still from the actual movie.

trunk

All the versions that make up a versioned asset.

version

A subsequent version of an asset. When you check in an asset, StudioCentral creates another version of the asset. See also *check in*, *check out*, *non-versioned*, *versioned*.

version tree

A schema for identifying all the versions of a particular versioned asset. As users continue to modify an asset (checking it in and out), StudioCentral “grows” its asset tree by creating a new version that incorporates each new change. Each version tree consists of a root and a trunk. See also *root*, *trunk*.

versioned

When you first check in an asset, you can check it in as a versioned asset so that StudioCentral will track subsequent revisions (*versions*) of the asset.

Video Server Toolkit (VST)

A toolkit for creating digital data playback and record engines for SGI Origin servers and O2 workstations. VST provides real-time, frame-accurate recording and playback of broadcast-quality digital media data.

zeroth version

The latest version of an asset. See also *version*, *version tree*, *trunk*.

Index

A

Access Control, 30
Admin Tool, 5, 19-22, 31, 36
AMASS Archive Management Storage System, 26
API. *See* application programming interface
application programming interface, 23, 29-31, 32
applications, 5, 7-22
 developing, 5, 29, 30, 32
AQL. *See* Asset Query Language
Asset Manager Server, 2, 27, 31, 35
 status of services, 22
Asset Query Language, 3
assets, xi, 1-5, 7
 access to, 19
 administering, 19
 archiving, 3, 28
 browsing, 3
 checking in, 3, 7, 8, 10, 12, 14, 15, 17, 27, 28
 checking out, 3
 deleting, 31
 moving content, 31
 retrieving, 3, 12, 17
 searching for, 3, 8, 10, 15, 20, 27
 storing. *See* assets
 checking in, 7
 transferring, 3, 19, 31
 viewing, 3, 27
Asset Store, 29
Asset Transfer Service, 30, 33
asset typing, 2, 7, 28

ATS. *See* Asset Transfer Service
attributes, xi, 10, 15, 29
Avalon Archive Manager, 26
Avalon Content Server, 27

B

Backup and Restore Services, 29

C

C++ API, 33
cataloging. *See* asset typing
Catalog window, 9
Check In window, 14
client applications, 7-18
 developing, 32
command-line utilities, 5, 31
 developing applications, 33
configurations, 35-38
 multiple-server, 36
 single-server, 35
Consistency Checking Services, 31
content, xi, 1-2, 20, 26, 27, 30
 moving, 31
content server keys, 27, 28
content servers, 27, 31, 33
Content Services, 25, 26

core software, 23, 25-29

D

database keys, 25, 28

databases, 2, 25, 38

Informix, 26, 27

Oracle, 26

Data Migration Facility, 26

Datamodel Manager, 29, 30

Datamodels, xi, 29, 30

DMF Content Server, 27

DMF. *See* Data Migration Facility

F

Filesystem Content Server, 27

G

groups

defining, 19

removing, 19

K

keywords, 10, 15

M

Macintosh Asset Browser, 14

Macintosh client, 5, 14-18, 30

Macintosh computers, 1, 5, 7

metadata, xi, 1-2, 7, 10, 15, 25, 29, 30

Metadata Services, 25

Migration Manager, 30, 31

multimedia data, 1

P

policy modules, 31

R

repositories, xii, 19, 25, 30

checking for consistency, 19, 31

plug-ins, 30

S

SC/Perl Module, 33

services, 23, ??-29, 31-??

SGI MediaBase, 26, 27, 28

SGI workstations, 1

software, 23-33

application programming interface, 23, 29-31

core, 23, 25-29

services, ??-29, 31-??

status, 22

storage

archive, 26

hierarchical, 26

online, 26

StudioCentral

applications, 7-22

configurations, 35-38

functional overview, 1-5

installing and configuring, 37

software, 23-33

T

thumbnails, 2, 28

U

users

adding to groups, 19

defining, 19

removing, 19

V

versions, 2, 8, 14

Video Server Toolkit, 3, 30

W

Web browsers, 1, 5, 19

Windows client, 5, 8-13

Windows computers, 1, 5, 7

Windows Explorer, 8

X

XFS filesystems, 26

Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-3830-002.

Thank you!

Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
 - On the Internet: techpubs@sgi.com
 - For UUCP mail (through any backbone site): *[your_site]!sgi!techpubs*
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 650-932-0801
- To send your comments by **traditional mail**, use this address:

Technical Publications
Silicon Graphics, Inc.
1600 Amphitheatre Parkway, M/S 535
Mountain View, California 94043-1351