UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NINTENDO CO., LTD., and
NINTENDO OF AMERICA INC.
Petitioners

v.

POLARIS POWERLED TECHNOLOGIES, LLC
Patent Owner

Case IPR 2023-00778
U.S. Patent No. 8,223,117 B2
Issue Date: July 27, 2012

Title: METHOD AND APPARATUS TO CONTROL DISPLAY BRIGHTNESS WITH AMBIENT LIGHT CORRECTION

DECLARATION OF DR. INGRID HSIEH-YEE

TABLE OF CONTENTS

I. Introduction....3

A. Qualifications and Professional Experience....4

B. MARC Standards and Cataloging Practices....5

C. Scope of This Declaration....12

D. Person of Ordinary Skill in the Art ("POSA")....14

E. Evidence Considered in Forming My Opinions....15

II. Authenticity and Public Availability of *Kerman* (Ex. 1014)....18

A. Authentication....18

B. Amazon Copy of *Kerman*....20

C. Library of Congress Records....21

D. Copyright Record....28

E. Loyola University Chicago Library Records....29

F. Indiana University of Pennsylvania Library Records....31

G. Summary of My Opinion on *Kerman*....33

III. Authenticity and Public Availability of *Rosenzweig* (Ex. 1015)....35

A. Authentication....35

B. Amazon Copy of *Rosenzweig*....36

C. Library of Congress Records....37

D. Mississippi State University Library Records....42

E. Copyright Record 45

F. Usage Record 46

G. Summary of My Opinion on Rosenzweig.....47

IV. Conclusion 48

I. Introduction

I, Ingrid Hsieh-Yee, Ph.D., do hereby declare as follows:

1. I have been retained as an expert by Nintendo Co., Ltd. and Nintendo of America Inc. (collectively, "Petitioners" or "Nintendo"), who I am informed are the Petitioners in this IPR proceeding.

2. I have prepared this declaration at the request of the Petitioners to provide my expert opinion regarding the public availability of specific books, identified below.

Exhibit	Description
1014	Excerpts from Philip Kerman, *ActionScripting in FLASH* (2001) ("*Kerman*")
1015	Excerpts from Gary Rosenzweig, *Using Macromedia Director 8* (2000) ("*Rosenzweig*")

3. My report sets forth my opinions in detail and provides the basis for my opinions regarding the public availability of each book. The materials that I considered and relied upon are discussed explicitly in this declaration. I reserve the right to supplement or amend my opinions, and bases for them, in response to any additional evidence, testimony, discovery, argument, and/or other additional information that may be provided to me after the date of this declaration.

4. I am being compensated for my work in this matter at my customary hourly rate. I am also being reimbursed for any reasonable expenses associated with my work and testimony in this investigation. My compensation is not contingent on the results of my study, the substance of my opinions, or the outcome of this matter.

A. Qualifications and Professional Experience

5. My complete qualifications and professional experience are described in my academic curriculum vitae **(Appendix A**). The following is a brief summary of my relevant qualifications and professional experience.

6. In 2022 I was conferred the title of Professor Emerita after 32 years as a professor in the Department of the Library and Information Science of the Catholic University of America. I have experience working in an academic library, a medical library, and a legislative library. I hold a Ph.D. in Library and Information Studies from the University of Wisconsin-Madison and a Masters in Library and Information Studies from the University of Wisconsin-Madison.

7. I am an expert on library cataloging and classification and have published two editions on this subject, *Organizing Audiovisual and Electronic Resources for Access*: *A Cataloging Guide* (2000 and 2006). I taught a variety of courses, including Cataloging and Classification, Advanced Cataloging and Classification, Organization of Internet Resources, Organization of Information,

Digital Content Creation and Management, Metadata, Internet Searches and Web Design, Information Literacy Instruction, Advanced Information Retrieval and Analysis Strategies, and The Information Professions in Society. I am familiar with metadata schema design and implementation. In my teaching I covered the design and implementation of metadata in libraries, databases, search engines, digital repositories, digital libraries, and digital archives. I also covered how the ways we organize information resources affect the discovery and access to digital resources on the Internet. My research interests cover cataloging and classification, information organization, metadata, information retrieval, information architecture, digital collections, scholarly communication, social media, user interaction with information systems, and others.

B. MARC Standards and Cataloging Practices

8. I am very familiar with a library cataloging encoding standard known as the "Machine-Readable Cataloging" standard, also known as "MARC," which became the national standard for sharing bibliographic data in the United States by 1971 and the international standard by 1973. MARC is the primary communications protocol for the transfer and storage of bibliographic metadata in libraries. Experts in my field reasonably rely upon MARC records when forming their opinions.

9. A MARC record consists of several fields, each of which contains specific data about the work. Each field is identified by a standardized, unique, three-digit code corresponding to the type of data that follows. **Appendix B** is a true and correct copy of Parts VII to X of "Understanding MARC Bibliographic: Machine-Readable Cataloging" (http://www.loc.gov/marc/umb/um07to10.html) from the Library of Congress. *Understanding MARC Bibliographic* was originally published by the Follett Software Co. and copyrighted in 1988. Editions in 1989, 1990, 1994, 1998, 2000, 2003 and 2009 include changes. The latest edition was copyrighted in 2009. This document explains commonly-used MARC fields. For example, the personal author of the work is recorded in Field 100, the title is recorded in Field 245, publisher information is recorded in Field 260, the physical volume and characteristics of a publication are recorded in Field 300, and topical subjects are recorded in Field 650.

10. The Online Computer Library Center (OCLC) is the largest bibliographic network of the world, with more than 554 million records and thousands of member institutions (many of which are libraries of some type) in more than 100 countries. OCLC was founded in 1967 to promote and support library cooperation. According to the "Third Article, Amended Articles of Incorporation of OCLC Online Computer Library Center, Inc.," OCLC was created "to establish,

maintain and operate a computerized library network and to promote the evolution of library use, of libraries themselves, and of librarianship, and to provide processes and products for the benefit of library users and libraries, including such objectives as increasing availability of library resources to individual library patrons and reducing the rate of rise of library per-unit costs, all for the fundamental public purpose of furthering ease of access to and use of the ever-expanding body of worldwide scientific, literary and educational knowledge and information" (source: https://www.oclc.org/content/dam/oclc/membership/articles-of-incorporation.pdf).

11. OCLC members can contribute original cataloging records in MARC to the OCLC system or derive cataloging records from existing records, an activity referred to as "copy cataloging." When an OCLC participating institution acquires a work, it can create an original MARC record for the work in OCLC's Connexion system (a system for catalogers to create and share MARC records), and the system will automatically generate a code for the date of record creation in the yymmdd format, and the creating library's OCLC symbol is recorded in subfield "a" of the 040 field. Once the MARC record is in Connexion, it becomes available to other OCLC members for adoption to their local online catalogs (*i.e.*, copy cataloging).

12. After a MARC record is created in Connexion, it also becomes searchable and viewable on WorldCat, which is a free web portal for users to explore

the collections of more than 10,000 libraries worldwide. The record in WorldCat, however, is not presented in MARC fields. Instead, the data elements are labeled to help users interpret the record. Thus, the information stored in MARC records in Connexion is available to the interested public through the free and user-friendly WorldCat web portal.

13. WorldCat (http://www.worldcat.org) is "the world's largest network of library content and services" and its features are summarized in "about WorldCat" (http://www.worldcat.org/about). WorldCat began operation in January 1998 as a free web portal to more than 10,000 library collections worldwide. Through WorldCat, users can search for information resources in their local libraries and libraries around the world. WorldCat allows users to search for books, CDs, videos, and many new types of digital content, such as audiobooks, in many languages. Users can also retrieve research materials and article citations with links to their full text. After an item is retrieved, WorldCat helps users identify a library nearby that holds the item or all the libraries that hold the item. WorldCat is an efficient way to explore library collections on the Internet. WorldCat is publicly accessible, and there is no fee to perform searches on WorldCat.

14. Library online catalogs gained acceptance in the early 1980s and many libraries migrated their systems to the World Wide Web in the mid-1990s. Library

online catalogs are based on MARC records that represent their collections in order to help the public understand what materials are publicly accessible in those libraries. Most libraries with online catalogs have made their catalogs freely available on the Web. These online catalogs offer user-friendly search interfaces. Strong user interest in keyword searches and the popularity of Google have led to the "googlization" of library search systems. As a result, many library catalogs now provide a single search box for users to conduct keyword searches, with additional support for searches by author, title, subject terms, and other data elements such as ISBN (International Standard Book Number). Library catalogs these days also offer features for users to narrow their search results by language, year, format, and other elements. Many libraries display MARC records on their online catalogs with labels for the data elements to help the public interpret MARC records. Many libraries also offer the option to display MARC records in MARC fields.

15. Libraries create MARC records for works they acquire to facilitate the discovery and access to works in their collections, which may include books, serials, motion pictures, and publications in other formats. Monograph cataloging is fairly common in libraries, and most libraries make a newly cataloged monograph available to the public soon after the cataloging work is completed, usually within a week. Libraries can create original cataloging records or use an existing record in

OCLC to create a copy cataloging record. As soon as the cataloging record is completed, it is added to the library's online catalog for users. If the record is an original record, it is also entered into the OCLC system. If it is a copy cataloging record, the library's holding symbol is attached to the existing original record in the OCLC system to facilitate searching and interlibrary loan.

16. A common workflow for making a newly acquired book publicly accessible in libraries is as follows. After a new book is received, the cataloging work begins. National libraries, such as the Library of Congress and the British Library, usually date stamp newly received materials to record the date of receipt, while other types of libraries do not date stamp all newly received materials, except for new issues of journals and other types of materials that are of special local interest.

17. After a new book is cataloged, its MARC record is added to the library's online catalog to make the record available for searching. This step enables users to discover the newly cataloged book. Meanwhile, the library processes the newly cataloged book quickly to make it publicly available, usually within a week after the MARC record is created, though some libraries with more processing volumes (like the Library of Congress) can take one to three months. If a user

discovers a new book and wishes to access it but the book is still in processing, the library will arrange to notify the user when the work is ready for user access.

18. A POSA would be able to discover materials in libraries in a number of ways. For example, A POSA could walk into a library and ask the librarian for materials in a certain field, by a certain author, or with a certain title. A POSA could also conduct searches himself or herself, using keywords, author, title, or subject headings, because those elements are usually included in MARC records. A POSA could explore library collections by using computers at libraries to search the online catalogs or searching library online catalogs on the Internet.

19. I am personally familiar with many online catalogs, databases, and search engines. In preparing for this declaration, I used the following authoritative information systems to search for records:

- Google Scholar (https://scholar.google.com)
- Online catalog of Indiana University of Pennsylvania (https://www.iup.edu/library/)
- Online catalog of Loyola University Chicago Libraries (http://libraries.luc.edu/)
- Online catalog of the Library of Congress (https://catalog.loc.gov)

- Online catalog of the library of Mississippi State University (https://mlp.ent.sirsi.net/client/en_US/msstate/?)
- Public catalog of the US Copyright Office (https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?DB=local&PAGE=First)
- WorldCat (free for public access on the Web) (https://www.worldcat.org)

C. Scope of This Declaration

20. I have been asked to offer an opinion on the authenticity and public availability date of the following documents:

- Excerpt of Kerman, P., (2001, April), *ActionScripting in FLASH*, SAMS, **Ex. 1014** ("*Kerman*");
- Portions of Rosenzweig, G., (2000, January), *Using Macromedia Director 8, Special Edition* (also known as *Special Edition Using Macromedia Director 8*), Indianapolis, Ind.: Que, **Ex. 1015** ("*Rosenzweig*").

21. I am rendering my expert opinion on the authenticity of the documents referenced herein and on whether they were published. I will provide my opinion on when and how these documents were publicly disseminated or otherwise made publicly available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could have located and retrieved these documents.

22. I am informed by counsel that an item is considered authentic if there is sufficient evidence to support a finding that the item is what it is claimed to be. I am also informed that authenticity can be established based on the contents of the documents themselves, such as the appearance, content, substance, internal patterns, protocols followed or other distinctive characteristics of the item.

23. I am informed by counsel that a given reference is "published" if it is publicly accessible upon a satisfactory showing that such a document has been disseminated or otherwise made publicly available to the extent that persons interested and ordinarily skilled in the subject matter or art could locate it exercising reasonable diligence and obtain the document. I have also been informed by counsel that materials available in a library constitute "printed publications" if they are cataloged and indexed according to general library practices and protocols that make the references available and accessible to members of the interested public.

24. I am told by counsel that the subject matter of this proceeding generally relates to controlling display brightness. I am told by counsel to assume for purposes of this declaration only, that the alleged priority date is March 28, 2003, which is the priority date Patent Owner (Polaris) is asserting in the district court litigation with Petitioners.

D. Person of Ordinary Skill in the Art ("POSA")

25. I am told by counsel that a "person of ordinary skill in the art at the time of the inventions" is a hypothetical person who is presumed to be familiar with the relevant field and its literature at the time of the alleged invention. This hypothetical person is also a person of ordinary creativity, capable of understanding the scientific principles applicable to the pertinent field.

26. I am told by counsel that a person of ordinary skill ("POSA") in this field would typically be someone who possessed at least:

> A bachelor's degree in electrical engineering, physics, optics or a related field, and approximately three years of experience in the fields of visual displays, circuit design, and related technologies.

27. I am told by counsel that various other proposed definitions of a POSITA have been proffered also:

> District Court Litigation With Petitioners: *Patent Owner*: A bachelor's degree in electrical engineering, physics, or computer science, and 2-4 years of experience in the field of visual displays and related technologies. *Petitioners*: A Bachelor of Science degree in electrical engineering, physics, or a similar field; and three years of industry experience designing analog electronics; or equivalent background and experience. Alternatively, a POSITA would have been someone with a Master of Science or Master of Engineering degree in electrical engineering, physics, or a similar field; and one year of industry experience designing analog electronics; or equivalent background and experience.
>
> District Court Litigation Between Patent Owner and Samsung: *Samsung*: A person of ordinary skill in the art at the time of filing

> (March 2003) would have at least a bachelor's degree in electrical engineering, physics, or optics, and at least three (3) years of experience with analog circuit design and optical sensors. *Patent Owner*: A person of ordinary skill in the art at the time of the invention of the '117 patent would have at least a bachelor's degree in electrical engineering, physics, or computer science, or 2-4 years of experience in the field of visual displays and related technologies.

28. Regardless of the specific definition applied, I understand that a person of ordinary skill in the art would have been familiar with and able to understand the information known in the art relating to these fields, including the books discussed in this declaration.

E. Evidence Considered in Forming My Opinions

29. In the preparation of this declaration, I have reviewed the documents referenced below and any other documents I reference herein, and each of these is a type of material that experts in my field would reasonably rely upon when forming their opinions:

(1) The document referenced above in Section I.B;

(2) Excerpt of *Kerman* from a used copy obtained from Amazon, **Appendix 1014A**;

(3) Bibliographic and MARC records for *Kerman*, available at https://lccn.loc.gov/00108995 from the online catalog of the Library of Congress, accessed and obtained on March 1, 2023, **Appendix 1014B**;

(4) Entry of Library of Congress available at https://librarytechnology.org/library/2263 from libraries.org, an online directory of libraries throughout the world, and an archived copy of the online catalog of the Library of Congress, available at https://web.archive.org/web/19991012141847/http://catalog.loc.gov/ from the Internet Archive, accessed and obtained on March 24, 2023, **Appendix 1014C**;

(5) Copyright registration record for *Kerman*, available at https://cocatalog.loc.gov/cgi-bin/Pwebrecon.cgi?v1=2&ti=1,2&Search%5FArg=Kerman%2C%20Phillip&Search%5FCode=NALL&CNT=25&PID=N_qzBGv0qsv0BspSBppPINN5XAsf&SEQ=20230302111318&SID=1 from the public catalog of the US Copyright Office, accessed and obtained on March 1, 2023, **Appendix 1014D**;

(6) Bibliographic and MARC records for *Kerman*, available at https://luc.primo.exlibrisgroup.com/discovery/fulldisplay?docid=alma9910099403402506&context=L&vid=01LUC_INST:01LUC&lang=en, from the online catalog of Loyola University of Chicago, accessed and obtained on March 2, 2023, **Appendix 1014E**;

(7) Bibliographic and MARC records for *Kerman*, available at https://sshelco-primo.hosted.exlibrisgroup.com/permalink/f/1g49mer/01SSHELCO_IUP_ALMA21151820400003564 from the online catalog of Indiana University of Pennsylvania, accessed and obtained on March 2, 2023, **Appendix 1014F**;

(8) Portions of *Rosenzweig* from a used copy obtained from Amazon, **Appendix 1015A**;

(9) Bibliographic and MARC records for *Rosenzweig*, available at https://lccn.loc.gov/00100575 from the online catalog of the Library of Congress, accessed and obtained on March 7, 2023, **Appendix 1015B**;

(10) Bibliographic and MARC records for *Rosenzweig*, available at https://mlp.ent.sirsi.net/client/en_US/msstate/search/detailnonmodal/ent:$002f$002fSD_ILS$002f0$002fSD_ILS:760147/ada?rt=CKEY|||CKEY|||false from the online catalog of the library of Mississippi State University, accessed and obtained on March 19, 2023, **Appendix 1015C**;

(11) Entry of Mississippi State University Library available at https://librarytechnology.org/library/1151 from libraries.org, an online directory of libraries throughout the world and an archived copy of the library website, available at https://web.archive.org/web/20010405184544/http://library.msstate.edu:80/collections/galaxy.asp from the Internet Archive; accessed and obtained on March 24, 2023, **Appendix 1015D**;

(12) Copyright registration record for *Rosenzweig*, available at %20Director&Search%5FCode=TALL&CNT=25&PID=-z6hgBGJsINmQ9csfpuIiYa6RUP_&SEQ=20230307211612&SID=1 from the public catalog of the US Copyright Office, accessed and obtained on March 7, 2023, **Appendix 1015E**;

(13) Early citations to *Rosenzweig*, obtained from Google Scholar, **Appendix 1015F**.

II. Authenticity and Public Availability of *Kerman* (Ex. 1014)

A. Authentication

29. **Ex. 1014** is a true and correct copy of an excerpt of *ActionScripting in FLASH*, by Phillip Kerman, published in 2001 by SAMS ("*Kerman*"), that I obtained from counsel. When I was asked to prepare this declaration, I searched WorldCat by the title of this book for records, and found a record at https://worldcat.org/title/47038202 that informed me the Library of Congress is one of the holding libraries. I then verified the holdings information in the online catalog of the Library of Congress. After that, counsel obtained a copy of an excerpt from the Library of Congress for me. This copy is presented as **Ex. 1014** in this declaration.

30. Page 1 of **Ex. 1014** shows the covers and spine of a bound volume identified as "KERMAN, ACTIONSCRIPTING IN FLASH, FT MEADE GenColl, TR 897.7 .K468 2001 Copy 2" and the back cover shows a Library of Congress barcode for this work. Page 2 is the book cover that shows Phillip Kerman is the author, "ActionScripting in FLASH" is the title, and SAMS is the publisher. Page 3 is the title page that shows the same author, title, and publisher information as the

book cover. It also shows a date stamp of "Library of Congress SEP 04 2001 Copyright Office," and a barcode with a label of "TX5-377-798" (also as TX0005377798) and indicates SAMS is located in Indianapolis, Indiana. Page 4 is the copyright page that shows a "2001" copyright date for this work, and Phillip Kerman as the copyright holder. This page also shows the book's International Standard Book Number is "0-672-32078-9" and its Library of Congress Catalog Card number is "00-108995." In addition, the copyright page shows "First Printing: April 2001."

31. Pages 5 and 6 present an overview of the parts and chapters of this book. Pages 7 to 13 are the table of contents that shows Part I Foundation includes 14 chapters, Part II Workshop includes beginning, intermediate, and advanced workshops and has a total of 21 chapters, and Part III Appendices includes two appendices and an index that begins on page 589.

32. Page 14 of **Ex. 1014** is the first page of "Part I Foundation" that lists 14 chapters included in this part, and pages 16 to 18 (internal pages [3] to 5) are the introduction that describes the 14 topics covered in Part I. Pages 20-23 (internal pages [7] to 10) are the beginning pages of Chapter 1 Flash Basics and page 24 (internal page 46) is the last page of Chapter 1. Pages 25 to 38 (internal pages [47] to 60) are the pages of Chapter 4 Basic Programming in Flash. Pages 39 to 51

(internal pages [389] to 401) are the pages of Chapter 4 Workshop on "Creating a Horizontal Slider." Page 53 is the first page of Chapter 5 Workshop on "Building a Slide Show" and page 54 is the back cover that shows the ISBN of this book is "0-672-32078-9."

B. Amazon Copy of *Kerman*

33. To authenticate **Ex. 1014**, I obtained a used copy of *Kerman* from Amazon and scanned the pages corresponding to those included **Ex. 1014** from my Amazon copy. The pages from the Amazon copy are presented as **Appendix 1014A** in this declaration. I have closely compared **Ex. 1014** with **Appendix 1014A** and concluded that the two copies have the same content. The only differences are that (1) **Ex. 1014** includes the covers of the bound volume, and (2) **Appendix 1014A** includes page 618, which is the last numbered page of *Kerman*, so that I could verify the pagination of this book in the Library of Congress records.

34. **Ex. 1014** is in a condition that creates no suspicion about its authenticity. Specifically, the portions included in **Ex. 1014** are not missing any intermediate pages (within their sequences), the text on each page appears to flow seamlessly from one page to the next, and there are no visible alterations to the document. Furthermore, **Ex. 1014** has the same content as the Amazon copy I scanned from a used copy (**Appendix 1014A)**, confirming that the pages in **Ex. 1014**

are from a published and publicly accessible work. I therefore see no reason to question the authenticity of **Ex. 1014**.

C. Library of Congress Records

35. As evidence of the public accessibility of *Kerman* in libraries, I obtained records for this book from the Library of Congress online catalog. I searched the Library of Congress because a WorldCat record I located at https://worldcat.org/title/47038202 informed me that the Library of Congress was a holding library. **Appendix 1014B** contains a true and correct copy of the bibliographic and MARC records for *Kerman*. I personally identified, located, and obtained these records from the online catalog of the Library of Congress. They are the type of records experts in my field would reasonably rely upon when forming their opinion.

36. Pages 1 and 2 of **Appendix 1014B** are the bibliographic record that shows the author, title, publisher, and physical description of *Kerman*. It also shows the ISBN of this book is "0672320789" and the LCCN (Library of Congress control number) for this book is "00108995." The "Item Available" area shows the book's call number is "TR897.7 .K468 2001 FT MEADE Copy 2" and it can be requested at the "Jefferson or Adams Building Reading Rooms." The book is "STORED OFFSITE" and the status of copy 2 is "Charged – Due – (Internal Loan)" (meaning

it was being used at the library at the moment). Interested users can consult reference librarians to find out if copy 2 has been returned to the stacks.

37. Page 3 is the MARC record that shows the date of record creation and revision. The first six digits of Field 008 show the MARC record was created on "000915" (*i.e.*, September 15, 2000), and the "s2001" code following this creation date indicates this publication has a single publication date of 2001. Field 955 is a field for recording local cataloging activities, and it shows the MARC record was created on "09-15-00," a copy of the book was sent to ASCD (Arts and Sciences Cataloging Division) on "2001-08-09" (i.e., August 9, 2001), two copies were sent to the catalogers to assign the Dewey Decimal Classification number on "2002-04-11" (i.e., April 11, 2004), and the cataloging work was completed on "2002-05-03" (i.e., May 3, 2002). Subfields "a" and "c" of Field 040 shows that "DLC" created the original record. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "DLC" is the OCLC library symbol for the Library of Congress. Fields 008, 040 and 955 inform my opinion that the original MARC record was a Cataloging-in-Publication (CIP) record based on the information supplied by the publisher. After the CIP record was completed, the Library of Congress received the published work on August 9, 2001, and their catalogers completed the cataloging work on May 3, 2002. A customary practice of

libraries is to process a newly cataloged monograph quickly to make it publicly accessible, usually within a week after the cataloging work is completed. Considering the large amount of works processed by the Library of Congress and my personal experience with their collections, it is my opinion that *Kerman* would have been publicly accessible one to three months after the cataloging work was completed in May 2002. This means that *Kerman* would have been publicly accessible at the Library of Congress as early as June 3, 2002, and no later than August 3, 2002.

38. The Cataloging in Publication Program (CIP) of the Library of Congress is responsible for cataloging books in advance of publication to alert the library community to forthcoming publications and facilitate acquisition. The initial CIP record is created based on information submitted by the publisher. The Library of Congress distributes CIP records in MARC format to large libraries, bibliographic services, and book vendors worldwide.

39. CIP records are also entered into OCLC Connexion system, the cataloging system containing more than 517 million MARC records, for OCLC member libraries to use it for copy cataloging and acquisition purposes. After the book is published, the publisher is obligated to send a copy to the CIP Program, and Library of Congress catalogers use the newly received publication to verify and

update data in the original CIP record.

40. Field 010 of the MARC record (**Appendix 1014B**) shows the Library of Congress control number assigned to *Kerman* and Field 020 shows the book's ISBN. Field 100 shows the author and Field 245 shows the title and author statement. Field 260 shows SAMS of Indianapolis, IN published this work in 2001. Field 300 shows the front matter of *Kerman* ends on page xi, the last numbered page is page 618, and the book includes illustrations. Field 500 shows this book includes an index. The LCCN, ISBN, author, title, publisher, publication place, publication date, pagination, and the note about the index match the information presented in **Ex. 1014** and **Appendix 1014A** and confirm this MARC record represents *Kerman*.

41. Subjects of this book are represented in Field 050 by a Library of Congress Classification (LCC) number of "TR897.7" that represents general works on "Computer animation" (source: Library of Congress Classification – Class T, https://www.loc.gov/aba/publications/FreeLCC/LCC_T2019TEXT.pdf) and in Field 082 by a Dewey Decimal Classification (DDC) number of "006.6/96" that represents "Digital video" (source: WebDewey at https://dewey.org/webdewey/login/login.html (subscription required for access)). Subjects are also represented in Field 650 by three Library of Congress topical subject headings, "Computer animation," "ActionScript (Computer program

language)," and "Web sites |x Design" ("|x" indicates a topical subdivision). In addition, Field 630 shows a subject heading "Flash (Computer file)" for the multimedia software platform to help users discover this book. These subject representations enable users, such as a POSITA, to conduct subject searches to find this book by *Kerman*.

42. Library MARC records are the foundation of online catalogs. These records provide essential information about materials in a collection to help users discover what they need. The records also include data elements, such as author, title, variant title, series title, table of contents, subject headings, subject keywords, classification numbers, and related organizations, to help users discover works of interest. Because of its simplicity, keyword searching is highly popular among users, and many online catalogs support keyword searching by checking user keywords in data fields that are rich with subject information, such as fields for title, table of contents, and subject headings. In addition, library catalogs offer many ways for searchers to make the search results manageable. For example, if an initial keyword search retrieves many items, users can use language, date, type of material (book, article, thesis, etc.), format (print or electronic), and access method (onsite or online) to make the retrieved set smaller. Users can also add more keywords to refine a retrieved set. Interested users, including a person of ordinary skill ("POSITA") in a

particular field, can conduct searches in library online catalogs to discover works of interest, or seek assistance from reference librarians to locate items relevant to their interest. For security reasons, Library of Congress has a closed stack system, meaning users have to request materials to be brought to their desks in the reading rooms.

43. This MARC record (**Appendix 1014B**) shows that the Library of Congress created a CIP record for *Kerman* on September 15, 2000, making it discoverable from that date on in their online catalog. Users would have been able to conduct subject searches by the LCC number, DDC number and the Library of Congress subject headings in the MARC record to discover this book. They would also have been able to search for *Kerman* by the author, title, ISBN and LCCN. For security reasons, Library of Congress has a closed stack system, meaning users have to request materials to be brought to their desks in the reading rooms. In this case, after POSITA has discovered *Kerman* in the online catalog, the user will ask for the book to be delivered to their reading room. If the user has any questions about access, the staff at the reading room will be able to provide assistance.

44. As evidence of the public accessibility of the online catalog of the Library of Congress, I located a library record from a directory and an archived copy of this online catalog from the Internet. **Appendix 1014C** is a true and correct copy

of an entry for the Library of Congress obtained from "libraries.org" (a web directory of libraries throughout the world) and an Internet Archive copy of the library catalog. These are the type of material experts in my field would reasonably rely upon when forming their opinions.

45. Pages 1 and 2 of **Appendix 1014C** are the entry for Library of Congress from "libraries.org" that shows the current automation system at the library is Voyager, and it has been in operation since 1998. The record history on page 2 shows that "[t]his listing was created on Aug 9, 1998 and was last modified on Nov 5, 2022." Pages 3 and 4 are the Internet Archive records of the online catalog of the Library of Congress. The top of page 3 is a Wayback Machine calendar that shows the URL of the archived website and indicates this website was archived 4,781 times during the period of "12 Oct 1999 – 24 Mar 2023." Page 3 shows this particular copy was archived on "OCT 12 1999" and is available at https://web.archive.org/web/19991012141847/http://catalog.loc.gov/. I have used the Internet Archive many times and am familiar with their practice of assigning an extended URL that indicates the date a web resource is archived. The "19991012141847" portion of the URL above indicates that this copy of the Library of Congress online catalog was archived on October 12, 1999, 2 p.m., 18 minutes, 47 seconds. These two records in **Appendix 1014C** show that the online catalog of

the Library of Congress has been in operation since 1998, and the search system was available on the Internet no later than October 12, 1999, when it was archived by Internet Archive.

D. Copyright Record

46. **Appendix 1014D** is a true and correct copy of the copyright registration record for *Kerman*. I obtained this record from the public catalog of the U.S. Copyright Office by first accessing their website at http://www.copyright.gov/, selecting the "Learn More" link under "Search Copyright Records," and following the link for "Official Public Catalog (1978 to present)" under "Online Records Collections" to perform a search by the author to retrieve the record. I personally located, identified and obtained this record. This is the type of record experts in my field would reasonably rely upon when forming their opinions.

47. The "Type of Work" field identifies the publication as "Text." The title, imprint, description, and copyright claimant information match the information presented in **Ex. 1014** and **Appendix 1014A**. The "Date of Creation" field shows this work was created in 2001, and the "Date of Publication" field shows "2001-04-09" (i.e., April 9, 2001). The "Registration Number / Date" field shows "TX0005377798 / 2001-09-04" indicating *Kerman* was registered with the Copyright Office on September 4, 2001. The copyright registration number and the

date of registration match the number and the date stamp on the title page in the Library of Congress copy (**Ex. 1014**). This record informs my opinion that *Kerman* was published on April 9, 2001, and registered with the U.S. Copyright Office on September 4, 2001.

E. Loyola University Chicago Library Records

48. My research on WorldCat for *Kerman* also identified the library of Loyola University Chicago as a holding library, so I searched their online catalog for records, and the search results confirmed the holding information. **Appendix 1014E** is a true and correct copy of the bibliographic record and MARC record for *Kerman*. I personally located, identified and obtained these records from the online catalog of the library of Loyola University Chicago. These are the type of record experts in my field would reasonably rely upon when forming their opinions. I captured the images of the bibliographic record and the MARC record to produce **Appendix 1014E**.

49. Pages 1 and 2 of **Appendix 1014E** are the bibliographic record for *Kerman* that shows the title, author, subject, publisher, creation date, format, language, imprint (publication place, publisher and copyright date), and physical description of this book. These data match the information presented in **Ex. 1014** and confirm this record represents *Kerman*. The "Location Items" area shows this

book has a call number of "TR897.7 .K468 2001," and the copy is available for request.

50. Page 3 is the MARC record. The first six digits of Field 008 show the record was created on "000915" (*i.e.*, September 15, 2000), and the "s2001" code following this record creation date shows that this book has a "2001" publication date. Subfields "a" and "c" of Field 040 shows that "DLC" (*i.e.*, the Library of Congress) created this record. Field 005 is a machine-generated field that automatically records the last time this MARC record was modified by the library, and it shows "20020430" (i.e., April 30, 2002) was the last time the library of Loyola University Chicago modified this MARC record. Data from Fields 008, 040, and 005 inform my opinion that the library of Loyola University Chicago used the Library of Congress MARC record, which is a Library of Congress CIP record created on September 15, 2000, to create their copy cataloging record, and their record was last modified on April 30, 2002, meaning *Kerman* would have been discoverable in their online catalog by at least April 30, 2002.

51. Because the Loyola MARC record (**Appendix 1014E**) was based on the Library of Congress CIP MARC record (**Appendix 1014B**), it has the same data for critical fields, including the LCCN in Field 010, ISBN in Field 020, author in Field 100, title in Field 245, publisher information in Field 260, physical description

in Field 300, and an index note in Field 500. This record also has the same subject representations as the Library of Congress record. The only difference is the Dewey Decimal Classification number is encoded in Field 092. This MARC record (**Appendix 1014E**) informs my opinion that by April 30, 2002, *Kerman* would have become discoverable in the online catalog of the library of Loyola University Chicago. Users would have been able to discover this book by the LCCN, ISBN, author, title, LCC number, DDC number, and Library of Congress subject headings.

F. Indiana University of Pennsylvania Library Records

52. Using the WorldCat record for *Kerman*, I also located the library records for *Kerman* from the library of Indiana University of Pennsylvania. **Appendix 1014F** is a true and correct copy of bibliographic and MARC records for *Kerman*. I personally located, identified and obtained these records from the online catalog of the library of Indiana University of Pennsylvania. I used screenshots to capture the files. This is the type of record experts in my field would reasonably rely upon when forming their opinions.

53. Pages 1 and 2 of **Appendix 1014F** are the bibliographic record for *Kerman* that shows the title, author, subject, publisher, creation date, format, language, and identifiers for this book. These data match the information presented in **Ex. 1014** and confirm this record represents *Kerman*. The record also shows this

book has a call number of "TR897.7 .K468 2001," is in the stacks of the Stapleton Library, and one copy is available for request.

54. Page 3 is the MARC record. The first six digits of Field 008 show the record was created on "000915" (*i.e.*, September 15, 2000), and the "s2001" code following this record creation date shows that this book has a "2001" publication date. Subfields "a" and "c" of Field 040 shows that "DLC" (*i.e.*, the Library of Congress) created this record. Field 005 is a machine-generated field that automatically records the last time this MARC record was modified by the library, and it shows "20020811" (i.e., August 11, 2002) was the last time the library of Indiana University of Pennsylvania modified this MARC record. Data from Fields 008, 040, and 005 inform my opinion that the library of Indiana University of Pennsylvania used the Library of Congress MARC record, which is a Library of Congress CIP record created on September 15, 2000, to create their copy cataloging record, and their record was last modified on August 11, 2002, meaning *Kerman* would have been discoverable in their online catalog by at least August 11, 2002.

55. Because the Indiana MARC record (**Appendix 1014F**) was based on the Library of Congress CIP MARC record (**Appendix 1014B**), it has the same data for critical fields, including the LCCN in Field 010, ISBN in Field 020, author in Field 100, title in Field 245, publisher information in Field 260, physical description

in Field 300, and an index note in Field 500. This record also has the same subject representations as the Library of Congress record. The only difference is that the Library of Congress Classification number is encoded in Field 090 in this copy cataloging record. This MARC record (**Appendix 1014F**) informs my opinion that by August 11, 2002, *Kerman* would have become discoverable in the online catalog of the library of Indiana University of Pennsylvania. Users would have been able to discover this book by the LCCN, ISBN, author, title, LCC number, DDC number, and Library of Congress subject headings.

G. Summary of My Opinion on *Kerman*

56. The excerpt of *Kerman* obtained from the Library of Congress (**Ex. 1014**) shows a date stamp of "SEP 04 2001" and the copyright registration record (**Appendix 1014D**) shows *Kerman* was published on April 9, 2001, and registered with the US Copyright Office on September 4, 2001. The MARC record of the Library of Congress (**Appendix 1014B**) shows that a CIP record was created on September 15, 2000, making this book discoverable from that date on in their online catalog. The Library of Congress completed their catalog record on May 3, 2002, so it is my opinion that the copy of *Kerman* would have become publicly accessible between June 3, 2002, and August 3, 2002. Internet Archive records of the online catalog of the Library of Congress (**Appendix 1014C**) shows that their

search system was available on the Web no later than October 12, 1999. The library of Loyola University Chicago used the MARC record created by the Library of Congress and last modified their record on April 30, 2002, and this means *Kerman* would have been discoverable in that library by April 2002. The library of Indiana University of Pennsylvania also used the MARC record created by the Library of Congress and last modified their record on August 11, 2002, and this means *Kerman* would have become discoverable in this library by August 11, 2002. Based on these data and my knowledge and experience with library cataloging and processing procedures and practices, it is my opinion that *Kerman* would have become discoverable at the Library of Congress since September 15, 2000, when the CIP record was created. This book became discoverable in April and August of 2002 at the library of Loyola University Chicago and the library of Indiana University of Pennsylvania, respectively. The published copy of *Kerman* would have become publicly accessible at the Library of Congress no later than August 3, 2002. It would also have become publicly accessible at the library of Loyola University Chicago no later than May 6, 2002, and at the library of Indiana University of Pennsylvania no later than August 18, 2002. In sum, every record I reviewed confirmed that *Kerman* was publicly accessible by an interested person of ordinary skill in the art no later than August 2002, and definitely before March 2003.

III. Authenticity and Public Availability of *Rosenzweig* (Ex. 1015)

A. Authentication

57. **Ex. 1015** is a true and correct copy of an excerpt of *Using Macromedia Director 8, Special Edition*, by G. Rosenzweig, published by Que in 2000 ("*Rosenzweig*"), that I obtained from counsel. When I was asked to prepare this declaration, I searched WorldCat by the title of this book for records, and found a record at https://www.worldcat.org/title/47054480 that informed me the Library of Congress is one of the holding libraries. I then verified the holdings information in the online catalog of the Library of Congress. Counsel then obtained a copy of an excerpt from the Library of Congress for me. This copy is presented as **Ex. 1015** in this declaration.

58. The scanned copy from the Library of Congress (**Ex. 1015**) presents the content of this book two pages at a time. Page 1 shows the covers and spine of a book entitled "Using Macromedia Director 8, Special Edition" that has Gary Rosenzweig as the author and Que as the publisher. The cover also shows a CD-ROM is included with the book. The spine shows a label of "MRC QA 76.575 .R684 2000 Copy 1 FT MEADE MRC" and the back cover shows a Library of Congress barcode for this book and an ISBN of "0-7897-2334-4." Page 2 is the inside cover and the first page of the "Macromedia Director 8 Lingo Reference Card." Page 3

shows the second page of the Reference Card and the title page, which shows the same title, author, and publisher information as the cover. This title page indicates Que is a Division of Macmillan Publishing USA and is located in Indianapolis, Indiana. This page also shows the table of contents at a glance, indicating that the book has 11 parts, including 38 chapters and "The Lingo Reference Guide" as Part XI. Page 4 (internal page ii) is the copyright page that shows this book has a "2000" copyright date, and Que is the copyright holder. It also shows the ISBN of this book is "0-7897-2334-4," the Library of Congress Catalog Card Number is "00-100575" and this copy is of the "First Printing: July 2000." This page carries a date stamp of "Library of Congress JUL 26 2000 Copyright Office, indicating that the copy was received by the Library of Congress on July 26, 2000. Pages 4 to 9 (internal pages iii to xiii) are the table of contents that shows the topics covered by each chapter and the Lingo Reference Guide (Part XI). Pages 10 to 13 (internal pages [1] to 6) is the Introduction to this book. Pages 14 to 28 (internal pages [305] to 332) are the contents of Chapter 15 Graphic Interface Elements.

B. Amazon Copy of *Rosenzweig*

59. To authenticate **Ex. 1015**, I obtained a used copy of *Rosenzweig* from Amazon and scanned the pages corresponding to those included **Ex. 1015** from my Amazon copy. The pages from the Amazon copy are presented as **Appendix 1015A**

in this declaration. I have closely compared **Ex. 1015** with **Appendix 1015A** and concluded that these two copies have the same content. The only difference I noticed is that **Appendix 1015A** presents pages from *Rosenzweig* one at a time, includes additional pages of the front matter to show the front matter ends on page xvii, includes page 1047, which is the last numbered page of this book, and also includes a scan of the CD.

60. **Ex. 1015** is in a condition that creates no suspicion about its authenticity. Specifically, the portions included in **Ex. 1015** are not missing any intermediate pages (within their sequences), the text on each page appears to flow seamlessly from one page to the next, and there are no visible alterations to the document. Furthermore, **Ex. 1015** has the same content as the Amazon copy I scanned from a used copy (**Appendix 1015A)**, confirming that the pages in **Ex. 1015** are from a published and publicly accessible book. I therefore see no reason to question the authenticity of **Ex. 1015**.

C. Library of Congress Records

61. **Appendix 1015B** is a true and correct copy of the bibliographic and MARC records for *Rosenzweig*. I personally identified, located, and obtained these records from the online catalog of the Library of Congress. They are the type of records experts in my field would reasonably rely upon when forming their opinion.

62. Pages 1 and 2 of **Appendix 1015B** are the bibliographic record that shows the author, title, edition statement, publisher, and physical description of *Rosenzweig*. It also shows the ISBN of this book is "0789723344" and the LCCN (Library of Congress control number) for this book is "00100575." The "Item Available" area shows the book's call number is "QA76.575 .R684 2000 FT MEADE Copy 1" and it can be requested at the "Machine Readable Collections." This book is "STORED OFFSITE" and the status of copy 1 is "Not Charged" (meaning available for user request).

63. Page 3 is the MARC record that shows the date of record creation and revision. The first six digits of Field 008 show the MARC record was created on "000131" (*i.e.*, January 31, 2000), and the "s2000" code following this creation date indicates this publication has a single publication date of 2000. Field 955 is a field for recording local cataloging activities, and it shows the MARC record was created on "01-31-00," a copy of the published work was sent to ASCD (Arts & Sciences Cataloging Division) on "02-21-01" (*i.e.*, February 21, 2001), and the book was sent to the Dewey program on "05-31-01" (*i.e.*, May 31, 2001) for the assignment of a Dewey Decimal Classification (DDC) number. Subfields "a" and "c" of Field 040 show that "DLC" created the original record. According to the *Directory of OCLC Members* (https://www.oclc.org/en/contacts/libraries.html), "DLC" is the OCLC

library symbol for the Library of Congress. Fields 008, 040 and 955 inform my opinion that the original MARC record was a Cataloging-in-Publication (CIP) record based on the information supplied by the publisher. After the CIP record was completed, the MARC record was added to the library catalog to make this book discoverable in the library's catalog. After the book was published, catalogers at the Library of Congress completed the cataloging work on May 31, 2001, or soon after this date, because the assignment of a DDC number is usually the last step of cataloging at the Library of Congress, and this task usually does not take more than a day to complete. A customary practice of libraries is to process a newly cataloged monograph quickly to make it publicly accessible, usually within a week after the cataloging work is completed. Considering the large amount of works processed by the Library of Congress and my personal experience with their collections, it is my opinion that *Rosenzweig* would have been publicly accessible one to three months after the cataloging work was completed by early June of 2001. This means that *Rosenzweig* would have been publicly accessible at the Library of Congress as early as June 2001, and no later than August 2001.

64. The Cataloging in Publication Program (CIP) of the Library of Congress is responsible for cataloging books in advance of publication to alert the library community to forthcoming publications and facilitate acquisition. The initial

CIP record is created based on information submitted by the publisher. The Library of Congress distributes CIP records in MARC format to large libraries, bibliographic services, and book vendors worldwide.

65. CIP records are also entered into OCLC Connexion system, the cataloging system containing more than 554 million MARC records, for OCLC member libraries to use it for copy cataloging and acquisition purposes. After the book is published, the publisher is obligated to send a copy to the CIP Program, and Library of Congress catalogers use the newly received publication to verify and update data in the original CIP record.

66. Field 008 of the MARC record for *Rosenzweig* (**Appendix 1015B**) shows that the record was first created on January 31, 2000. Field 955 shows a copy of the book was sent to the cataloging unit on February 21, 2001, and catalogers completed the cataloging work soon after May 31, 2001. The time gap between 2000 and 2001 is due to the fact that the initial record was a CIP record, which was created based on information provided the publisher. In other words, a CIP record is a pre-publication record. The copyright registration record for this book (**Appendix 1015E**), to be discussed below, will show that the official publication date of *Rosenzweig* was July 19, 2000. After the book was published, the publisher then registered it with the copyright office on July 26, 2000. A few months later, on

February 21, 2001, a copy of *Rosenzweig* was sent to the cataloging unit, and catalogers used it to update and enrich the original CIP record, and completed their work soon after May 31, 2001. After processing, the Library of Congress copy of *Rosenzweig* would have become publicly accessible no later than August 2001.

67. Field 010 of the MARC record (**Appendix 1015B**) shows the Library of Congress control number assigned to *Rosenzweig* and Field 020 shows the book's ISBN. Field 100 shows the author and Field 245 shows the title and author statement. Field 260 shows Que of Indianapolis, Indiana, published this book with a 2000 copyright date. Field 300 shows the front matter of *Rosenzweig* ends on page xii, the last numbered page is page 1047, and the book includes illustrations and a CD. The LCCN, ISBN, author, title, edition statement, publisher, publication place, publication date, and pagination match the information presented in **Ex. 1015** and **Appendix 1015A** and confirm this MARC record represents *Rosenzweig*.

68. Subjects of this book are represented in Field 050 by a Library of Congress Classification (LCC) number of "QA76.575" that represents "Multimedia systems" (source: Library of Congress Classification – Class Q, https://www.loc.gov/aba/publications/FreeLCC/LCC_Q2019TEXT.pdf) and in Field 630 by a subject heading for "Director (Computer file)." These subject representations enable users to conduct subject searches to find this book by

Rosenzweig.

69. This MARC record (**Appendix 1015B**) shows that the Library of Congress created a CIP record *Rosenzweig* on January 31, 2000, making it discoverable in their online catalog from that date on. As discussed above in Paragraph 45, an Internet Archive record shows the online catalog of the Library of Congress was available on the Internet by October 12, 1999 (**Appendix 1014C**). This means users would have been able to use the online catalog on the Internet to explore the collection at the Library of Congress no later than October 12, 1999. As of 2000 and 2001, interested users would have been able to conduct subject searches by the LCC number and the Library of Congress subject headings in the MARC record to discover this book. They would also have been able to search for *Rosenzweig* by the author, title, ISBN, and LCCN.

D. Mississippi State University Library Records

70. As evidence of the availability of *Rosenzweig* in other libraries, I used the WorldCat record for this book to identify another holding library. **Appendix 1015C** is a true and correct copy of the bibliographic and MARC records for this book that I obtained from the online catalog of the libraries of Mississippi State University. I personally located, identified and obtained these records. They are the type of record experts in my field would reasonably rely upon when forming their

opinions.

71. Page 1 of **Appendix 1015C** is the bibliographic record for *Rosenzweig* that shows the title, author, edition, publisher, copyright date, ISBN, language and subjects of this book. The bottom of the page shows this book is available from the MSU-Mitchell Memorial Library and its call number is QA76.575 .R684 2000.

72. Page 2 is the MARC record that shows the dates when the record was created and modified. The first six digits of Field 008 show the original MARC record was created on "000131" (*i.e.*, January 31, 2000), and Field 040 shows "DLC" (*i.e.*, Library of Congress) created the original MARC record on this date. Field 005 is a machine-generated field that automatically records the last time the MARC record was modified in this library, and it shows the last record modification date was "20010531" (*i.e.*, May 31, 2001). Data in Fields 008, 040, and 005 inform my opinion that the library system of Mississippi State University used the Library of Congress MARC record for copy cataloging, and they last modified their MARC record on May 31, 2001, meaning this MARC record would have been available to help users discover *Rosenzweig* by May 31, 2001.

73. The Mississippi State University MARC record (**Appendix 1015C**) is based on the Library of Congress MARC record (**Appendix 1015B**), except for some local information. The former uses the same bibliographic data as the latter,

including the LCCN in Field 010, ISBN in Field 020, LC Classification number in Field 050, author in Field 100, title in Field 245, edition statement in Field 250, publisher and copyright date in Field 264, physical description in Field 300, and subject headings in Field 630 and Field 650.

74. This MARC record (**Appendix 1015C**) made *Rosenzweig* discoverable in the online catalog of the library by at least May 31, 2001. Interested users would have been able to locate this book by subject and by author, title, LCCN, and ISBN. A common library practice is to process newly cataloged books quickly to make them available for public access, usually within a week after the cataloging work is completed. It is therefore my opinion that the copy of this book would have become publicly accessible at the library of Mississippi State University by June 7, 2001.

75. As evidence that the online catalog of the library of Mississippi State University was in operation in 2001, I obtained a directory entry for the library and an Internet Archive records of the library website. **Appendix 1015D** is a true and correct copy of these records. I personally located, identified, and obtained these records. They are the type of material experts in my field would reasonably rely upon when forming their opinions.

76. Pages 1 and 2 of **Appendix 1015D** are the entry for the Mississippi State University Libraries from “libraries.org” that shows that from 1993 to 2001,

the automation system at the library was DRA. The record history on page 2 shows that "[t]his listing was created on Jun 15, 1997 and was last modified on Jan 6, 2023." Pages 3 to 5 are the Internet Archive records of the library website. The top of page 3 is a Wayback Machine calendar that shows the URL of the archived website and indicates this website was archived 54 times during the period of "17 Oct 2000 – 3 Feb 2013." Page 3 shows this particular copy was archived on "APR 05 2001" and is available at https://web.archive.org/web/20010405184544/http://library.msstate.edu:80/collections/galaxy.asp. The "20010405184544" portion of the URL above indicates that this copy of the library website was archived on April 5, 2001, 6 p.m., 45 minutes, 44 seconds, and the archived copy of the website shows that by April 5, 2001, users would have been able to use their online catalog to conduct five types of searches.[1]

E. Copyright Record

77. **Appendix 1015E** is a true and correct copy of the copyright registration record for *Rosenzweig*. I obtained this record from the public catalog of the U.S. Copyright Office by first accessing their website at http://www.copyright.gov/,

[1] In fact, the Mississippi State University library catalog was available on the Web as early as October 2000. *See* October 17, 2000 Internet Archive record at https://web.archive.org/web/20001017195147/http://library.msstate.edu:80/collections/galaxy.asp.

selecting the "Learn More" link under "Search Copyright Records," and following the link for "Official Public Catalog (1978 to present)" under "Online Records Collections" to perform a title search for the record. I personally located, identified and obtained this record. This is the type of record experts in my field would reasonably rely upon when forming their opinions.

78. The "Type of Work" field identifies the publication as "Text." The application title, title, edition, imprint, description, and copyright claimant information match the information presented in **Ex. 1015** and **Appendix 1015A**. The "Date of Creation" field shows this work was created in 2000, and the "Date of Publication" field shows "2000-07-19" (i.e., July 19, 2000). The "Registration Number / Date" field shows "TX0005349335 / 2000-07-26" indicating *Rosenzweig* was registered with the Copyright Office on July 26, 2000. The date of registration match the date stamp date on the copyright page of the Library of Congress copy (**Ex. 1015**). This record informs my opinion that *Rosenzweig* was published on July 19, 2000, and registered with the U.S. Copyright Office on July 26, 2000.

F. Usage Record

79. Actual usage of a publication is reflected by the publications that make reference to it. **Appendix 1015F** is a list of four citations I compiled from Google Scholar to demonstrate usage of *Rosenzweig* in 2001. The earliest citation appeared

in May 2001, and a citation published by IEEE appeared in June 2001.

G. Summary of My Opinion on Rosenzweig

80. The excerpt of *Rosenzweig* obtained from the Library of Congress (**Ex. 1015**) shows a date stamp of "JUL 26 2000" and the copyright registration record (**Appendix 1015E**) shows *Rosenzweig* was published on July 19, 2000, and registered with the US Copyright Office on July 26, 2000. The MARC record of the Library of Congress (**Appendix 1015B**) shows that a CIP record was created on January 31, 2000, making this book discoverable from that date on in their online catalog. The Library of Congress completed their catalog record no later than early June of 2001, so it is my opinion that the copy of *Rosenzweig* would have become publicly accessible between June and August of 2001. The copy cataloging record from the library of Mississippi State University (**Appendix 1015C**) shows that their record would have been in their online catalog by May 31, 2001, and users would have been able to use it to discover *Rosenzweig* by that time. The physical copy of this book would have been publicly accessible no later than June 7, 2001. An Internet Archive record shows that the online catalog of this library was available on the Internet no later than April 5, 2001. Based on these data and my knowledge and experience with library cataloging and processing procedures and practices, it is my opinion that *Rosenzweig* would have become discoverable at the online catalog of

the Library of Congress since January 31, 2000, when the CIP record was created. The published copy of *Rosenzweig* would have become publicly accessible at the Library of Congress no later than August of 2001, and the copy would also have been publicly accessible by June 7, 2001, at the library of Mississippi State University. Early citations to *Rosenzweig* were published in May and June of 2001. In sum, every record and physical copy I reviewed confirmed that *Rosenzweig* was publicly accessible by an interested person of ordinary skill in the art no later than August 2001, and definitely before March 2003.

IV. Conclusion

82. In signing this declaration, I recognize that the declaration will be filed as evidence in a Petition for *Inter Partes* Review before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to deposition in the case. If deposition is required of me, I will appear at a reasonable time and place to be agreed upon.

83. I hereby declare that all statements made herein on my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Date: 3-27-2023 in Annapolis, MD

Executed: Ingrid Hsieh-Yee
Ingrid Hsieh-Yee, Ph.D.

APPENDIX A

Ingrid Hsieh-Yee

Professor Emerita
Dept. of Library and Information Science
Catholic University of America
Washington, D.C. 20064
E-mail: hsiehyee@cua.edu

Education

Ph.D. 1990. Library and Information Studies, University of Wisconsin-Madison
Minors: Sociology and Psychology

M.A. 1984. Library and Information Studies, University of Wisconsin-Madison.

M.A. 1983. Comparative Literature, University of Wisconsin-Madison.

B.A. 1980. Foreign Languages and Literature, National Taiwan University.

Work Experience

Professor, School/Dept. of Library and Information Science, Catholic University of America, 1990-2022 (Assistant Professor, 1990-1996; Associate Professor, 1997-2004; Professor 2004-2022; Professor Emerita 2022)

Co-Chair, Dept. of Library and Information Science, Catholic University of America, June 2015-August 2016.

Acting Dean, School of Library and Information Science, Catholic University of America, January 2010-June 2012.

Cataloger, Dept. of Legislative Reference Library, Annapolis, Maryland, 1989-1990.

Lecturer, School of Library and Information Studies, University of Wisconsin-Madison, 1988.

Teaching Assistant, School of Library and Information Studies, University of Wisconsin-Madison, 1986-1988.

Cataloger, Health Sciences Library, University of Wisconsin-Madison, 1984-1986.

Areas of Teaching and Research Interests

Information Organization and Access; Metadata; Cataloging & Classification; Information Architecture; Information Retrieval; Digital Collections; Social Media; Scholarly

Communication; Information Behavior; Health Informatics; Human Computer Interaction; Usability Studies

Grants & Honors

Cultural Heritage Information Management Project. IMLS grant. Amount: $498,741. Period: Aug. 2012 to July 2015. Co-PI with Dr. Youngok Choi.

D.C. Health Information Technology (HIT4): Building Capacity & Providing Access in Our Nation's Capital. Dept. of Labor H2B Training Grant. Grant amount: $4,175,500. Grant period: Nov. 2011 to Dec. 2015. Partner with the Metropolitan School of Professional Studies of the Catholic University of America, Children's National Medical Center, D.C. Department of Employment Services, Holy Cross Hospital, Howard University, Center for Urban Progress, Providence Hospital, and Sibley Memorial Hospital.

Capital Health Careers Project. Department of Labor Healthcare Sector and Other High Growth and Emerging Industries Grant. Grant amount: $4,953,999. Grant period: March 2010 – February 2013. Awarded to a group of healthcare organizations and educational institutions in Washington, D.C. Providence Health Foundation of Providence Hospital (Lead institution). Part of the grant supported the development of a Master's degree program in Information Technology with a concentration in Health Information Technology offered by the School of Library and Information Science.

The Washington D.C. School Librarians Project. IMLS grant. Grant amount: $412,660. Grant period: Aug. 2007 – June 2011. The School partnered with the District of Columbia Public Schools (DCPS) and the District of Columbia Library Association to educate and mentor school media specialists for the DCPS system. PI, Jan. 2010 to June 2011.

SIG Member of the Year, American Society for Information Science and Technology (2009).

Most Outstanding Paper of *OCLC Systems & Services* (2001).

ALISE Research Grant (2001).

Most Outstanding Paper of *OCLC Systems & Services* (2000).

Research Grant from ERIC (1999-2000).

Best Research Paper Award; Association for Library and Information Science Education (1998).

Research Grants, Catholic University of America. 1991, 1992, 1993, 1996, 1998, 1999, 2004, 2005, 2006, 2007, 2013-14.

Cooperative Faculty Research Grant, Consortium of Universities in the Washington Metropolitan Area (1993-1994).

Cooperative Research Grant, Council on Library Resources (1993-1994).

Journal of the American Society for Information Science Best Paper Award (1993).

ASIS/ISI Information Science Doctoral Dissertation Scholarship (1989).

HEA Title IIB Fellowship (Dept. of Education) (1989)

Chinese-American Librarians Association Scholarship (1987).

Beta Phi Mu (1985).

Vilas Fellowship, University of Wisconsin-Madison. 1984

Publications

Hsieh-Yee, I. (2021). Can We Trust Social Media? *Internet Reference Services Quarterly*, 1-15. Published online: 28 July, 2021. 10.1080/10875301.2021.1947433.

Bailey, T., & Hsieh-Yee, I. (2020). Combating the Sharing of False Information: History, Framework & Literacy Strategies, *Internet Reference Services Quarterly*, 24 (1-2), 9-30.

Choi, Y., and Hsieh-Yee, I. (2010). Finding Images in an OPAC: Analysis of User Queries, Subject Headings, and Description Notes. *Canadian Journal of Information and Library Science,* 34(3): 271 – 295.

Hsieh-Yee, I. (2008). Educating Cataloging Professionals in a Changing Information Environment. *Journal of Education for Library and Information Science,* 46(2): 93-106.

Vellucci, S. L., Hsieh-Yee, I., and Moen, W.E. (2007). The Metadata Education and Research Information Commons (MERIC): A Collaborative Teaching and Research Initiative. *Education for Information,* 25(3&4): 169-178.

NISO Framework for Guidelines for Building Good Digital Collections. 3rd ed. Baltimore, MD: National Information Standards Organization, 2007. Also available online: http://www.niso.org/framework/framework3.pdf (NISO Working Group members: Priscilla Caplan (chair), Grace Agnew, Murtha Baca, Tony Gill, Carl Fleischhauer, Ingrid Hsieh-Yee, Jill Koelling, and Christie Stephenson.)

Choi, Y., Hsieh-Yee, I., and Kules, B. (2007). Retrieval Effectiveness of TOC and LCSH. *Proceedings of the Joint Conference on Digital Libraries*, pp. 233-234.

Vellucci, S., and Hsieh-Yee, I. (2007). They Didn't Teach Me That in Library School! Building

a Digital Teaching Commons to Enhance Metadata Teaching, Learning and Research. *Proceedings of the National Conference of the Association of College and Research Libraries, Baltimore, MD*, pp. 26-31.

Mitchell, Vanessa, and Ingrid Hsieh-Yee. (2007). Converting Ulrich's Subject Headings to FAST Headings: A Feasibility Study. *Cataloging & Classification Quarterly,* 45(1): 59-85.

Hsieh-Yee, I., Tang, R., and Zhang, S. (2007). User Perceptions of a Federated Search System. *IEEE Technical Committee on Digital Libraries Bulletin,* Summer 3(2) (URL = http://www.ieee-tcdl.org).

Tang, R., Hsieh-Yee, I., and Zhang, S. (2007). User Perceptions of MetaLib Combined Search: An Investigation of How Users Make Sense of Federated Searching." *Internet Reference Services Quarterly,* 12(12): 211-236.

Hsieh-Yee, I., Tang, R., and Zhang, S. (2006). User Perceptions of a Federated Search System. *Proceedings of the Joint Conference on Digital Libraries*, *June 11-15, 2006, Chapel Hill*, p. 338.

Hsieh-Yee, I. (2006). *Organizing Audiovisual and Electronic Resources for Access: A Cataloging Guide*. 2nd ed. Westport, Conn.: Libraries Unlimited.

NISO A Framework of Guidance for Building Good Digital Collections. 2nd ed. Bethesda, MD: National Information Standards Organization, 2004. Framework Advisory Group: Grace Agnew, Liz Bishoff, Priscilla Caplan (Chair), Rebecca Gunther and Ingrid Hsieh-Yee.

Hsieh-Yee, I. (2004). Cataloging and Metadata Education in North American LIS Programs. *Library Resources & Technical Services,* 48(1): 59-68.

Hsieh-Yee, I. (2004). Cataloging and Metadata Education. In Gary E. Gorman (Ed.), *International Yearbook of Library and Information Management 2003: Metadata Applications and Management*, (pp.204-234). London: Facet Publishing.

Yee, P. L., Hsieh-Yee, I., Pierce, G.R., Grome, R., and Schantz, L. (2004). Self-Evaluative Intrusive Thoughts Impede Successful Searching on the Internet. *Computers in Human Behavior,* 20(1): 85-101.

Hsieh-Yee, I. (2003). Cataloging and Metadata Education: A Proposal for Preparing Cataloging Professionals of the 21st Century. A report submitted to the ALCTS-Education Task Force in response to Action Item 5.1 of the *Bibliographic control of Web Resources: A Library of Congress Action Plan*. Approved by the Association for Library Collections and Technical Services. Web version available since April 2003 at http://lcweb.loc.gov/catdir/bibcontrol/CatalogingandMetadataEducation.pdf.

Hsieh-Yee, I. (2002). Cataloging and Metadata Education: Asserting a Central Role in Information Organization. *Cataloging & Classification Quarterly* 34(½): 203-222.

Hsieh-Yee, I., and Smith, M. (2001). The CORC Experience: Survey of Founding Libraries, Part I. *OCLC Systems & Services,* 17: 133-140. (Received "The Most Outstanding Paper of OCLC Systems & Services in 2001" award.)

Hsieh-Yee, I., and Smith, M. (2001). The CORC Experience: Survey of Founding Libraries, Part II, Automated Tools and Usage. *OCLC Systems & Services,* 17: 166-177. (Received "The Most Outstanding Paper of OCLC Systems & Services in 2001" award.)

Hsieh-Yee, I. (2001). ERIC User Services: Changes and Evaluation for the Future. *Government Information Quarterly,* 18: 31-42.

Hsieh-Yee, Ingrid. (2001). Research on Web Search Behavior. *Library and Information Science Research,* 23: 167-185.

Logan, E., and Hsieh-Yee, I. (2001). Library and Information Science Education in the Nineties. *Annual Review of Information Science and Technology,* 35: 425-477.

Hsieh-Yee, I. (Ed.) (2001). *Library and Information Science Research*, 23 (2). A special issue in honor of the retirement of Douglas L. Zweizig.

Hsieh-Yee, I. (2000). *ERIC User Services: Evaluation in a Decentralized Environment*. Washington, D.C.: Dept. of Education.

Hsieh-Yee, Ingrid. (2000). *Organizing Audiovisual and Electronic Resources for Access: A Cataloging Guide*. Littleton, CO: Libraries Unlimited.

Hsieh-Yee, I. (2000). Organizing Internet Resources: Teaching Cataloging Standards and Beyond. *OCLC Systems & Services,* 16: 130-143. (Received "The Most Outstanding Paper of OCLC Systems & Services in 2000" award.)

Hsieh-Yee, I. (1998). The Retrieval Power of Selected Search Engines: How Well Do They Address General Reference Questions and Subject Questions? *Reference Librarian,* 60: 27-47.

Hsieh-Yee, I. (1998). Search Tactics of Web Users in Searching for Texts, Graphics, Known Items and Subjects: A Search Simulation Study. *Reference Librarian,* 60: 61-85. (Received the 1997 Best ALISE Research Paper Award.)

Hsieh-Yee, I. (1997). Access to OCLC and Internet Resources: LIS Educators' Views and Teaching Practices. *RQ,* 36: 569-86.

Hsieh-Yee, I. (1997). Teaching Online and CD-ROM Resources: LIS Educators' Views and Practices. *Journal of Education for Library and Information Science,* 38: 14-34.

Hsieh-Yee, I. (1996). The Cataloging Practices of Special Libraries and Their Relationship with OCLC. *Special Libraries,* 87: 10-20.

Hsieh-Yee, I. (1996). Modifying Cataloging Practice and OCLC Infrastructure for Effective Organization of Internet Resources. In *Proceedings of the OCLC Internet Cataloging Colloquium*. [Online]. Available: http://www.oclc.org/oclc/man/colloq/hsieh.htm

Hsieh-Yee, I. (1996). Student Use of Online Catalogs and Other Information Channels. *College & Research Libraries,* 57: 161-175.

Hsieh-Yee, I. (1995). Ten entries in James S. C. Hu (Ed.), *Encyclopedia of Library & Information Science*, 913, 1028-29, 1036, 1037, 1145-46, 1514, 1575, 1763-64, 2216-27, 2378-79. Taipei, Taiwan: Sino-American Publishing. (Topics include "Advanced Technology/Libraries," "Information Ethics," "Instruction on Cataloging and Classification," "Instruction on Reference Services.")

Hsieh-Yee, I. (1993). Effects of Search Experience and Subject Knowledge on Online Search Behavior: Measuring the Search Tactics of Novice and Experienced Searchers. *Journal of the American Society for Information Science,* 44: 161-174. (Received the 1993 Best JASIS Paper Award.)

Works in Progress

Hsieh-Yee, I. Actionable lessons for debunking false information: A comparative analysis of literacy lessons in communication and library and information science

Hsieh-Yee, I. Representing diverse voices and communities in cataloging and indexing: The evolution of controlled vocabularies

Hsieh-Yee, I. Information Organization and Retrieval in Practice (a monograph scheduled for completion and publication in 2023)

Presentations

Bailey, T. & Hsieh-Yee, I. (February 2020). The Phenomena of Sharing Misinformation and the Need for Information Literacy. Presented at the 2020 Bridging the Spectrum Symposium, Washington, D.C., February 14, 2020.

Hsieh-Yee, I. and Fragan-Fly, J. (May 2018) Trends, Design & Strategies for Digital Scholarship Services. Presented at the 2018 Maryland/Delaware Library Association Conference, Cambridge, MD.

Hsieh-Yee, I. (February, 2018) Research Data Management: What It Takes to Succeed. Presented at the 10th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I. (February, 2017) *Research Data Management: New Competencies and Opportunities for Information Professionals*. Presented at the 9th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I. and Lawton, P. (February, 2017) *Enhancing Catholic Portal Searches with User Terms and LCSH*. Presented at the 9th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I. (2016, October) *Visualizing Data for Information*. Presented at the 2016 Virginia Library Association Conference, Hot Springs, VA.

Hsieh-Yee, I. (2016, August) *Religious Materials Toolbox for Archivists: Solutions to Problems Facing the Profession*. Presented at Archives * Records 2016, Atlanta, GA.

Hsieh-Yee, I. and Lawton, P. (2016, March) *Enhancing Retrieval of Catholic Materials with LCSH Knowledge Structure*. Presented at the 2016 Catholic Library Association Conference, San Diego, CA.

Fagan-Fry, J. and Hsieh-Yee, I. (2016, February) *Approaches to Digital Scholarship at Top Universities around the World: Scholarly Publishing in the Digital Age.* Presented at the 8th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee and Fagan-Fry, J. (2016, January) *Innovative Services for Digital Scholarship at Top 100 Research Libraries of the World*. Poster presented at the 2016 Annual Conference of the Association for Library and Information Science Education, Boston, Mass.

Hsieh-Yee, I. and Lawton, P. (2015, June). *Crowdsourcing terms for CRRA portal themes.* Poster presented at the third CRRA symposium and annual meeting, Bringing the created toward the Creator: Liturgical art and design since Vatican II. Catholic Theological Union, Chicago, Illinois.

Hsieh-Yee, I. and Lawton, P. (2015, February). *Crowdsourcing terms for thematic exploration in the Catholic Portal.* Poster presented at the 7th Annual Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I., James, R., and Fagan-Fry, J. (2015, February). *Support for digital scholarship at top university libraries of the world*. Poster presented at the 7th Annual Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I., Zhang, S., Lin, K., and Cherry, S. (2015, February). *Thus said the end users: Summon experience and support for research workflows*. Poster presented at the 7th Annual Bridging the Spectrum Symposium, Washington, D.C.

Yontz, E., Hsieh-Yee, I., & Houston, S. (2015, February). *Healthy Heroes Summer Reading Club: Developing healthy youth at public libraries.* 11th Annual Jean Mills Health Symposium, Greenville, North Carolina.

Yontz, E., Hsieh-Yee, I., and Houston, S. (2015, January). *Healthy youth and libraries: A pilot study.* Association for Library & Information Science Education (ALISE) Annual Conference, Chicago, Illinois.

Hsieh-Yee, I. (2014, May). *Linking CRRA resources to portal themes via authority files*. Presented at the Catholic Research Resources Alliance 2014 Membership Meeting, Marquette, WI.

Hsieh-Yee, I. (2014, April). *Enhancing subject access to CRRA resources*. Presented at the 2014 Catholic Library Association Conference, Pittsburgh, PA.

Hsieh-Yee, I. (2014, January). *Health Information Technology Program: Educational entrepreneurship in action*. Presented at the 2014 annual Conference of the Association for Library and Information Science Education, Philadelphia, PA

Hsieh-Yee, I., Zhang, S., Lin, K., and Cherry, S. (2014, January). *Discovering information through Summon: An analysis of user search strategies and search success*. Paper presented at the 6th Bridging the Spectrum Symposium, Washington, D.C.

Hsieh-Yee, I. (2012, December). *National Digital Stewardship Alliance and SLIS at CUA: An Educational Partnership*. Paper presented at Best Practices Exchange: Acquiring, Preserving, and Providing Access to Government Information in the Digital Era, Annapolis, MD

Choi, Y. and Hsieh-Yee, I. (2010, January). *Finding Images in an OPAC: Analysis of User Queries, Subject Headings, and Description Notes*. Paper presented at 2nd Annual Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. and Coogan, J. (2010, January). *Google Scholar vs. Academic Search Premier: What Libraries and Searchers Need to Know*. Paper presented at 2nd Annual Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. (2009, November). *Information Science Education: An LIS School's Perspective*. Paper presented at Annual Meeting of the American Society for Information Science and Technology, Vancouver, British Columbia, Canada.

Hsieh-Yee, I., Menard, E., Ya-Ning Chen, A., Shu-Jiun Chen, S., Kalfatovic, M. R., Wisser. K. M. (2009, November). *Information Organization in Libraries, Archives and Museums: Converging Practices and Collaboration Opportunities*. Presented at Annual Meeting of the American Society for Information Science and Technology, Vancouver, British Columbia, Canada. (Organizer and moderator of this panel.)

Hsieh-Yee, I. and Coogan, J. (2009, July). *Catching up to Google Scholar: The Retrieval Power of Academic Search Premier and Google Scholar*. Poster presented at American Library Association Conference, Chicago, Illinois.

Hsieh-Yee, I., with the CUA Scholarly Communications Project Team. (2009, January). *Digital Scholarship@CUA: Developing an Institutional Repository for CUA*. Poster presented at 1st Annual Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.

Wise, M., Cylke, K., and Hsieh-Yee, I. **(**2009, January). *Digital Talking Books: Meeting the Needs of the Blind and the Handicapped*. Paper presented at the Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. (2009, January). *User Expectations of MERIC*. Presented at the Information Organization Competencies for the 21st Century Discussion Session of the 2009 Conference of the Association for Library and Information Science Education, Denver, Colorado.

Choi, Y., and Hsieh-Yee, I. (2008,November). *Subject Access for Images in an OPAC.* Annual Meeting of the American Society for Information Science and Technology, Columbus, Ohio. (Also co-organized a panel on Retrieving and Using Visual Resources: Challenges and Opportunities for Research and Education.)

Hsieh-Yee, I. (2008, June). *Educating Cataloging Professionals in a Changing Information Environment*. National Taiwan University, Taipei, Taiwan.

Vellucci, S. L., Moen, W.E., Hsieh-Yee, I., Marson, B., and Wisser, K. (2008, January) *Building a Metadata Education and Research Community through MERIC (Metadata Education and Research Information Commons): Demo and Stakeholder Input*. A panel presented at the 2008 Conference of the Association for Library and Information Science Education, Philadelphia, Pennsylvania.

Hsieh-Yee, I., Choi,Y. and Kules, B. (2007, October). *Searching for Books and Images in OPAC: Effects of LCSH, TOC and Subject Domains.* A poster presented at the American Society for Information Science and Technology Annual Meeting, Milwaukee, Wisconsin.

Hsieh-Yee, I. and Coogan, J. (2007, August) *Google Scholar vs. Academic Search Premier: A Comparative Analysis.* Presented to the Faculty and Staff of the University of the District

of Columbia.

Hsieh-Yee, I. and Coogan, J. (2007, June). *Google Scholar vs. Academic Search Premier: A Comparative Analysis*. Presented to the Washington Research Library Consortium Community, Catholic University of America, Washington, D.C.

Hsieh-Yee, I., Choi, Y., and Kules, B.. (2007, June). *What Users Need for Subject Access: Table of Contents or Subject Headings?* A poster presented at the 2007 American Library Association Annual Conference, Washington, D.C., June 2007.

Choi, Y., Hsieh-Yee, I., and Kules, B. (2007, June). *Retrieval Effectiveness of TOC and LCSH.* A paper presented at the Joint Conference on Digital Libraries 2007, Vancouver, Canada.

Vellucci, S. L., Hsieh-Yee, I., and Moen, W.E. (2007, May). *If We Build It, Will They Come? Building a Community of Practice for Metadata Stakeholders.* A poster presented at the Rutgers University Research Day, Bridgeton, New Jersey.

Hsieh-Yee, I. (2007, May). *Federated Searching: User Experience & Perceptions.* International Conference on Information Organization & Retrieval, National Taiwan University, Taipei, Taiwan.

Hsieh-Yee, I. (2007, May). *Search Performance of Google Scholar and Academic Search Premier.* International Conference on Information Organization & Retrieval, National Taiwan University, Taipei, Taiwan.

Hsieh-Yee, I. (2007, May) *MERIC: Building a Digital Commons for Metadata Education & Research.* International Conference on Information Organization & Retrieval, National Taiwan University, Taipei, Taiwan.

Hsieh-Yee, I., and Coogan, J. (2007, March/April). *A Comparative Analysis of Google Scholar and Academic Search Premier.* Poster presented at the Association of College & Research Libraries 13th National Conference, Baltimore, Maryland.

Vellucci, S. L. and Hsieh-Yee, I. (2007, March/April) *They Didn't Teach Me That in Library School! Building a Digital Teaching Commons to Enhance Metadata Teaching, Learning and Research*. On-site presentation and Webcast by Elluminate. A contributed paper presented at the Association of College & Research Libraries 13th National Conference, Baltimore, Maryland. The acceptance rate for contributed paper was 20%. This paper was one of 10 conference papers chosen for live webcast during the conference.

Moen, W., Hsieh-Yee, I. and Vellucci, S.L. (2007, January) *A DSpace Foundation for a Teaching & Research Commons: The Metadata Education and Research Information Commons.* A poster session presented at the Open Repositories Conference 2007, San Antonio, Texas.

Tang, R., Hsieh-Yee, I., and Zhang, S. (2006, November) *User Perception of MetaLib Combined Search*. Paper presented at the Annual Meeting of the American Society for Information Science and Technology, Austin, Texas, Nov. 2006.

Hsieh-Yee, I. (2006, November). *Federated Searching: User Perceptions, System Design, and Library Instructions.* Paper presented at the Annual Meeting of the American Society for Information Science and Technology, Austin, Texas. (Panel organizer, moderator, presenter).

Hsieh-Yee, I. (2006, November). *Building a Digital Teaching Commons to Enhance Teaching and Learning: The MERIC Experience and Challenges.* Paper presented at the Annual Meeting of the American Society for Information Science and Technology, Austin, Texas. (Panel organizer, moderator, presenter)

Hsieh-Yee, I. (2006, September). *Search Performance of Google Scholar and Academic Search Premier.* Paper presented at the ERIC Publishers Meeting, Washington, D.C.

Hsieh-Yee, I., Zhang, S., and Rong Tang, R. (2006, June). *User Perceptions of a Federated Search System*. Poster presented at Joint Conference on Digital Libraries, Chapel Hill, North Carolina.

Hsieh-Yee, I. and Zhang, S. (2006, June). *Preparing Users for Federated Search: Implications of a MetaLib User Perceptions Study.* Paper presented at the 2006 Ex Libris User Groups of North America Conference, Knoxville, Tennessee.

Hsieh-Yee, I. (2006, January). *MERIC Organizations and Navigation*. Paper presented at the 2006 ALISE Annual Conference, San Antonio, Texas.

Hsieh-Yee, I. (2006, January). *Metadata and Cataloging Education: Recommended Competencies.* Paper presented at the 2006 ALISE Annual Conference, San Antonio, Texas.

Hsieh-Yee, I. (2005, November). *Digital Library Evaluation: Progress & Next Steps.* Presentation at the Annual Meeting of the American Society for Information Science & Technology, Charlotte, North Carolina.

Hsieh-Yee, I. (2005, August). *Providing Access to Digital Content: Issues for DL Managers.* Presentation at MDK12 Digital Library Steering Committee Meeting, Columbia, Maryland.

Hsieh-Yee, I. (2005, April). *Enhancing Teaching and Learning: The Role of School Library Media Specialists.* Presentation at Meeting of the Baltimore County Public School System School Media Specialists, Baltimore, Maryland.

Hsieh-Yee, I. (2005, January). *Subject Access and Users: Insights & Inspirations from Marcia J. Bates.* Paper presented at the Historical Perspectives SIG, 2005 Conference of the Association for Library and Information Science Education, Boston, Massachusetts.

Hsieh-Yee, I. (2005, January). *Electronic Resource Management: Practice, Employer Expectations, & CE Interests.* Paper presented at Technical Services Education SIG, 2005 Conference of the Association for Library and Information Science Education, Boston, Massachusetts.

Hsieh-Yee, I. (2004, October). *Library Professionals for the Digital Age: Competencies & Preparation.* Paper presented at Bibliographic Access Management Team meeting, Library of Congress, Washington, D.C.

Hsieh-Yee, I. (2004, January). *Cataloging and metadata expertise for the digital era.* Presented at Preparing 21st Century Cataloging and Metadata Professionals: A Workshop for Educators and Trainers, San Diego and sponsored by ALCTS, ALISE, LC, and OCLC.

Hsieh-Yee, I. (2004, January). *Educating catalogers for the digital era.* Paper presented at the Technical Services SIG, 2004 Conference of the Association for Library and Information Science Education, San Diego.

Hsieh-Yee, I. (2003, July). *Cataloging Education for the 21st Century.* A presentation at the Library of Congress, Washington, D.C.

Hsieh-Yee, I. (2002, January) *Metadata Education and Research Priorities: A Delphi Study of Metadata Experts.* Presentation at the 2002 Conference of the Association for Library and Information Science Education, New Orleans.

Hsieh-Yee, I. (2001, November). *A Delphi Study of Metadata: Preliminary Findings.* Poster session at the 2001 Annual Meeting of the American Society for Information Science & Technology, Washington, D.C.

Hsieh-Yee, I. (2001, June). *Resources on Asian American Children: Analysis of Retrieval by Search Engines and WorldCat.* Presentation at the National Conference on Asian Pacific American Librarians, San Francisco.

Hsieh-Yee, I. (2001, January). *Delphi Study on Metadata: Project Design.* Presentation at Research Awards Session, Association for Library & Information Science Education, Washington, D.C.

Hsieh-Yee, I. (2000, May). *Web Search Behavior Research: Progress and Implications.* Presentation at the Symposium on Evaluating Library and Information Science Research, University of Wisconsin-Madison, Madison, Wisconsin.

Hsieh-Yee, I. (2000, March). *ERIC User Services: Evaluation in a Decentralized Environment.* Presentation at the National ERIC Joint Directors/Technical Meeting, Arlington, Virginia.

Hsieh-Yee, I. (2000, January). *Enhancing Learning with Web Technology.* Presentation at Faculty Conversations, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. (2000, January). *From Surrogates to Objects: CUA's Approaches to Organizing Electronic Resources.* Paper presentation at the Annual Conference of the Association for Library and Information Science Education, San Antonio, Texas.

Yee, P., and Hsieh-Yee, I. (1997, November). *Individual Differences in Search Behavior on the WWW.* A poster session presented at the 38th Annual Meeting of the Psychonomic Society, Philadelphia, Pennsylvania.

Hsieh-Yee, I. (1997, April). *Research + Marketing + Preparation = Job!* Presented at the "Workshop on Resume and Interview Techniques," Special Libraries Association, Student Chapter, Catholic University of America, Washington, D.C.

Hsieh-Yee, I. (1997, February). *Creating CyberCatalogers: Education and Training.* Presentation at ALA's Midwinter Meeting, Washington, D.C.

Hsieh-Yee, I. (1997, February). *Search Tactics of Web Users in Searching for Texts, Graphics, Known Items and Subjects: A Search Simulation Study.* Presented at the Conference of the Association for Library and Information Science Education, Washington, D.C.

Hsieh-Yee, Ingrid. "Beginning Your Special Library/Information Center Career." Presented at SLA's "Career Day," Jan. 11, 1997, Catholic University of America.

Hsieh-Yee, I. (1996, September). *The Roles of Library and Information Scientists in Managing Electronic Information.* Presentation at Hamilton College, Clinton, New York.

Hsieh-Yee, I. (1996, May). *The Future of Cataloging as a Profession.* Presented at "The Cataloging Forum, Library of Congress, Washington, D.C.

Hsieh-Yee, I. (1994, October). *The Impact of the Internet on OPACs.* Presented at the Third Workshop on User Interfaces for OPACs, Library of Congress, Washington, D.C.

Reports

Hsieh-Yee, I., with Knowledge Management Competencies and Performance Action Group of the Federal Knowledge Management Initiative. "From Knowledge Management Competencies to Improved Organizational Performance." April 9, 2009.

Hsieh-Yee, I., with Knowledge Practices Action Group of the Federal Knowledge Management Initiative. "KM Practice in Government Agencies: Findings and Recommendations." April 9, 2009.

Hsieh-Yee, I. "Delphi Study on Metadata." 2001. Three quarterly reports submitted to the Association for Library and Information Science Education.

Hsieh-Yee, I. "College Students' Information Channels: Patterns of Use and Possible Factors in Channel Selection." 1995. Submitted to the Catholic University of America.

Hsieh-Yee, I. "The Information-Seeking Patterns of Scholars and Their Use of an Online Information System." 1994. Submitted to the Council on Library Resources.

Book Reviews

Review of *The Measurement and Evaluation of Library Services*, by Sharon L. Baker and F. Wilfrid Lancaster. *Information Processing and Management* 30 (1994): 450-52.

Review of *Subject Access to Films and Videos, by Sheila S. Intner and William E. Studwell; and Cataloging Unpublished Nonprint Materials*, by Verna Urbanski with Bao Chu Chang and Bernard L. Karon. *Information Processing and Management* 30 (1994): 449-50.

Review of *Automated Information Retrieval in Libraries: A Management Handbook*, by Vicki Anders. *Journal of Library and Information Science* 19 (1993): 98-100.

Review of *Full Text Databases*, by Carol Tenopir and Jung Soon Ro. *Information Processing and Management* 28 (1992): 667-68.

Review of *Descriptive Cataloging for the AACR2R And USMARC: A How-to-Do It Workbook*, by Larry Millsap and Terry Ellen Ferl. *Information Processing and Management* 28 (1992): 809-11.

Review of *MARC Manual: Understanding and Using MARC Records*, by Deborah J. Byrne. *Information Processing and Management* 28 (1992): 537-38.

Service

Professional Associations and Societies

- Library of Congress. RDA Training Program for the Profession. Co-authored with Tim Carlton. 2013-2014.
- 2014 Digital Preservation Outreach & Education Survey. Contributed to the design of the survey, 2014.
- National Digital Stewardship Alliance. Outreach Committee. 2011-2014.
- National Digital Stewardship Residency Program. Advisory Group, 2012-2013.
- FEDLINK Health Information Technology Advisory Council, 2011-2015.

- 2012 Joint Conference on Digital Libraries. Program Planning Committee, Pre-Conference Proposals Review Committee, 2012
- Catholic Research Resources Alliance. Five-Year Strategic Plan Task Force, 2011-2012
- Institute of Museum and Library Services. Grant reviewer. 2004, 2005, 2010.

- Association for Library and Information Science Education.
  * ALISE Bodan Wynar Research Paper Award Committee, 2015, 2016, 2017
  * ALISE Eugene Garfield Dissertation Award Competition, Jury, 2013, 2014
  * ALISE Research Grant Competition Committee. Chair, 2012
  * Pratt-Severn Faculty Innovation Award. Chair, 2009, 2010
  * ALISE Doctoral Poster Jury, 2012
  * "Information Organization Competencies for the 21st Century" Discussion session leader. 2009 Conference of the Association for Library and Information Science Education.
  * Assisted Technical Services SIG Convener in organizing a program, ""Building a Metadata Education and Research Community through MERIC (Metadata Education and Research Information Commons): Demo and Stakeholder Input" for the 2008 ALISE conference.
  * Association for Library Collections and Technical Services/Association for Library and Information Science Education (ALCTS/ALISE) Metadata Education and Research Information Center (MERIC) Advisory Board, Co-Chair (with Sherry Vellucci), 2005-2007. Chair, 2008-2009 (leading the effort to build MERIC, a repository and collaborative space for metadata educators, practitioners, and researchers)
  * Technical Services SIG, Convener, 2004-2005. Organized a program on "Electronic Resources Management: Current Practices, Employer Expectations, and Teaching Strategies" for the 2005 conference in Boston, Massachusetts.
  * Technical Services SIG, Convener, 2003-2004. Organized a program on "Organizing Information with Metadata: Desired Competencies and Teaching Innovations" for the 2004 conference.
  * Technical Services SIG, Convener, 1999-2000. Organized a program on "Teaching the Organization of Electronic Resources" for the 2000 conference.
  * Curriculum SIG, Co-convener (with Sibyl Moses), 1996-97. Organized a program on "Government Information Policy" for the 1997 conference.

- American Society for Information Science & Technology.
  * Reviewer, Conference program panel submissions and poster submissions, 2005, 2006, 2007, 2009, 2011, 2012, 2013, 2014, 2015, 2016, 2017
  * Nomination Committee, 2009-2011
  * Information Science Education Special Interest Group. American Society for Information Science and Technology. Chair-Elect, 2007-2008. Chair 2008-2009.
  * Committee on Information Science Education. 1999-2006.
  * Committee on Information Science Education. Organizing Committee for an orientation program for students at ASIS annual meetings, 1999-2001
  * Committee on Information Science Education. Sub-committee on Student Welfare (focusing on issues related to master's education), 1998-2001

* SIG ED. Organizing Committee for the "Seminar on Research and Career Development" for junior researchers. 1995-96 (chair), 1997-2001
* ISI Doctoral Dissertation Proposal Scholarship Jury, 1997; 2001, 2002
* Pratt-Severn Best Student Research Paper Award Jury. Chair. 1997
* 1998 Midyear Meeting (referee of contributed papers), 1997
* Organizer and moderator of the ASIS Doctoral Forum and the Doctoral Research Seminar 1994-1995
* SIG Human Computer Interaction. Chair-Elect, Chair, 1993-1995
* Doctoral Forum Award Jury, 1995
* Best Student Paper Award Jury, 1995

- American Library Association.
  * Committee on Accreditation, External Review Panelist, 2009- (site visiting team 2013-2014; site visiting team 2016-2017)
  * Association for Library Collections and Technical Services Task Force on Competencies and Education for a Career in Cataloging, member, 2008-2009
  * Facilitator for "What They Don't Teach in Library School: Competencies, Education and Employer Expectations for a Career in Cataloging," an Association for Library Collections and Technical Services Preconference, June 22, 2007 in Washington, D.C. Also a local liaison for bringing this program to the Catholic University of America.
  * Facilitator for a discussion on "Effect of Electronic Resources on Technical Services" at ALA's Midwinter Meeting held in Feb. 1997 in Washington, D.C.
  * International Relations Committee, Subcommittee Task Force for IFLA and China, 1994-1997

- Virginia Association of School Librarians. Scholarships and Awards Committee. 2010-2012

- Federal Knowledge Management Initiative, Knowledge Management Practices Action Group. Member. 2009 (leading the effort to build a knowledge management repository)
- Federal Knowledge Management Initiative, Knowledge Management Competencies & Learning Action Group. Member. 2009 (developing an action plan for helping government knowledge workers and government agencies to develop knowledge management competencies)
- National Center for Education Statistics. Technical Review Panel. 2008.
- External evaluator for a case of promotion to full professorship. University of Tennessee. 2008.
- National Information Standards Organization (NISO). Advisory Board, Revision of "IMLS Framework of Guidance for Building Good Digital Collections," 2004, 2007.
- Library of Congress, Bibliographic Control of Web Resources: A Library of Congress Action Plan. Principal Investigator of Action Item 5.1, focusing on cataloging and metadata education for students and new librarians, 2002-2003. (worked with the Association for Library Collections and Technical Services, Education Task Force)

- Chinese American Librarians Association
  * Chinese American Librarians Association Outstanding Library Leadership Award in Memory of Dr. Margaret Chang Fung, Award Committee, 2016-2017
  * Achievement Award Jury, 2000-2001
  * CALA Goal 2000 Task Force, 1997
  * Scholarship Committee, 1995, 1996-1997 (chair)
  * Board of Directors, 1994-1997
  * Publication Committee, 1993-1995
  * International Relations Committee, 1993-1996

- Sailor[SM] Assessment Advisory Group (An impact study of Sailor, Maryland's Public Information Network), 1995

- Editorial boards
  Journal of Library & Information Science. Editorial Board, 2012-
  Chinese American Librarians Association, *Occasional Papers Series*. Editorial Board, 2009-2016.
  Library Quarterly. Editorial Board, 2003-2008
  Bulletin of the Medical Library Association, 1994-97
  Newsletter editor for the Chinese American Librarians Association, 1989-92

- Referee for the following journals
  Information Processing and Management
  Journal of Digital Information
  Journal of Education for Library and Information Science
  Journal of Information Science
  Journal of Library & Information Science
  Journal of Library Metadata
  Journal of the American Society for Information Science & Technology
  Library and Information Science Research
  Library Quarterly

- Expert reviewer, "Digital Library" course, Evaluation module, University of North Carolina, Chapel Hill, 2007-2008.
- Expert reviewer, "Information Organization" course, University of Michigan, Ann Arbor. 2007.

Catholic University of America

- Academic Senate Committee on Appointments and Promotions, 2005-2008
- Academic Senate Committee on Committees and Rules, 2009-2012, 2019-2022
- Academic Senate representative (for School of Arts & Sciences), 2017-2020

- Academic Senate representative (for School of Library & Information Science) 2003-2012
- Academic Senate Library Committee, Interim Chair (2007), Member, 2008-2012
- Academic Senate Committee on Computing, 1995-2003

- President's Administrative Council, 2010-2012
- Deans' Council, 2010-2012
- Academic Leadership Group, 2010-2012
- Graduate Board, 2010-2012

- School of Arts & Sciences, Committee on Appointments and Promotions, 2015-2019
- School of Arts & Sciences, Academic Council, 2015-2016.
- School of Arts & Sciences, Ordinary Professor Group, 2013-
- Doctoral Dissertation Defense Committee, Chair, Dept. of Psychology, 2016, 2017, 2018, 2019, 2020, 2021, 2022
- Doctoral Dissertation Defense Committee, Chair, Dept. of Education, 2014, 2015, 2017, 2018, 2019
- Doctoral Dissertation Defense Committee, Chair, National Catholic School of Social Services, 2019, 2020, 2021
- Doctoral Dissertation Defense Committee, Chair, School of Nursing, 2006, 2008
- CUA Scholarly Communication Project Team, Member (2007), Chair, 2008-2009
- Dean Search Committee, 1992-1994, 1998-1999, 2002-2003, 2006-2007
- Fulbright Review Panel, 2006
- CUA Service Learning Advisory Board, 2001-2002
- CUA Faculty Conversations on Enhancing Teaching and Learning through Technology, Planning Group, 1999-2001
- CUA Initiative on Technology and Teaching, 1998-2001

Dept. of Library and Information Science

- Admissions Committee, 2007-2009, Chair 2010-2012, Member 2013-2015, Member 2018-2022
- Appointments and Promotions Committee, 1991-2022
- Student advisement, 1990-2022
- 4+1 Bachelor to Master program, Program Advisor, 2014-2022
- LIS Minor, Advisor, 2017-2022
- Curriculum Committee, 2020-21 (Member), 2021-2022 (Chair)
- Scholarship & Awards Committee, 2016-2018, 2020-21
- Comprehensive examination editor, 2020-21
- Symposium, Colloquium, Lecture Series Committee, May 2018-May 2020 (Chair)
- Symposium and Colloquium Committee, fall 2016-May 2018
- Bridging the Spectrum Symposium session moderator, 2018, 2019, 2020, 2021

- Organization of Information Core Course Outcomes Assessment, 2019, 2020
- The Information Professions in Society Core Course Outcomes Assessment, 2019, 2020
- Community Services Librarianship, Course of Study Review and Revision, 2020
- Organization of Information, Course of Study Review and Revision, 2019
- Organization of Information, Competency and Course Offerings Review, 2018-2019
- Accreditation presentation, Chair, June 2015-August 2016
- Interim Co-Chair, June 2015-August 2016.
- Blended/OWL Learning Committee, spring 2016-2018
- Technology Committee, fall 2016-2017
- Comprehensive examination editor, 2016-2017, reader (every year since 1990)
- LIS Advisory Board, 2015-2016 (chair); fall 2016- May 2018 (member)
- Committee on Planning and Assessment, 2015-2016 (chair)
- Senior Faculty Committee, 2014-2016.
- Accreditation Steering Committee, 2014-2016 (Chair, 2015-2016)
- Accreditation Students Standard Committee, co-chair, 2014-2016
- Accreditation Mission, Goals, and Objectives Standard Committee, co-chair, 2014-2016
- Accreditation Curriculum Standard, member 2014-2-16
- Accreditation Administration and Finance Standard, member 2014-2016
- Cultural Heritage Information Management Project (IMLS-funded), Co-PI, 2012-2015
- Cultural Heritage Information Management Forum (scheduled for June 2015), Co-Organizer, 2013-2015
- Health Information Technology Interim Review Committee, 2015 (chair)
- Health Sciences Librarianship Advisory Group, 2015- (chair)
- Comprehensive examination editors, 2013-2014, 2016-2017
- National Digital Stewardship Alliance liaison, 2011-2014
- Advisory Board, Chair 2010-2012
- Academic Honesty Committee, Chair, 2008-2012
- Blended Learning Committee, 2010-2012
- Colloquium Committee, 2010-2012
- Comprehensive Examination Administration, 2010-2012
- Cultural Heritage Information Management Advisory Committee, 2010-2012 (chair), 2013-
- Curriculum Committee, 1991-2003, 2007-2009, Chair 2010-2012, member 2013-
- Curriculum Subcommittee on Comprehensive Examination, Chair 2009-2012
- Health Information Technology Advisory Board, Chair 2010-2012. Member 2013-
- Health Sciences Advisory Committee, 2009, Chair 2010-2012. Member 2013-
- HIT Expert Forum, Chair 2012. Member 2013-2015
- Health Information Technology Student Group Advisor, 2011-2012
- State Council for Higher Education of Virginia, SLIS Representative, 2010-2012
- Symposium Planning Committee, 2010-2012
- Website Management Team, Chair, 2010-2012
- Urban School Librarianship Project (IMLS-Funded), PI, 2007-2011 (chair, 2010-11)

- Failing Grades Committee, 1995-1997 (chair), 2000-2001 (chair), 2004-2005 (chair), 2007 (chair)-2011
- Faculty Search Committee, 1994-1998, 2002-2004, 2006 (chair), Fall 2007-2009, Chair fall 2009-2012
- Recruitment Committee, Chair 2010-2012
- Strategic Planning Committee, Chair 2010-2012
- Technology Committee, 2010-2012
- Accreditation Advisory Committee, 2007-2009
- Accreditation Coordinating Committee, 2007-2009
- Accreditation Steering Committee, 2007-2009
- SLIS Advisory Group, 2007-2009
- Accreditation Curriculum Standard Committee, Co-chair, 2007-2009
- Accreditation Faculty Standard Committee, Co-chair, 2007-2009
- LSC 551 Information Organization Review Team, Co-chair, 2008-2009, 2015-2016.
- Curriculum Subcommittee on Portfolios, 2009
- LSC 555 Information Systems in Libraries and Information Centers Review Team, contributor, 2008-2009
- Redesign of LSC 730 Use and Users of Libraries and Information. 2009-
- Development of a metadata institute that was taught as LSC 715 Organization of Internet Resources in 2008. The institute is being revised and will be offered in 2010 under a new course title.
- Development of lesson plans, assignments, and evaluation rubrics for LSC 606, Cataloging and Classification, for the School's NCATE accreditation. 2008
- Howard and Mathilde Rovelstad Scholarship Committee, Chair, 2004-2007
- Assistant Dean Search Committee, Chair, Fall 2007
- Liaison to the Association for Library Collections and Technical Services to bring its preconference program, Cataloging Education and Employer Expectations, to CUA during the 2007 American Library Association Annual Meeting in Washington, D.C.
- Organizer of the colloquium presentation and reception for Tamar Sadeh of Ex Libris on PRIMO June 2007
- Practicum review and design (work with potential supervisors, such as the American Indian Museum internship description revision) 2006-
- Comprehensive examinations (edits, proctoring, and grading), 1990-
- SLIS Web site redesign: Comments and suggestions. Fall 2007
- Conducted surveys of current students and alumni in preparation for the 2005 re-accreditation, 2004-2005
- Technology Committee, 1992-1999 (chair, 1996-1998), 2002-2003 (member)
- Colloquia Committee 1997-1999, 2002-2003.
- Advisor of the CUA Student Chapter of the American Society for Information Science and Technology, 2002-2003
- Visiting Professor Search Committee, 1999, 2000, 2001
- Leader, Participation in the CORC experiment, 1999-2000
- Advisor of the Special Libraries Association Student Chapter, 1993-1999; the group was recognized for outstanding leadership by SLA in 1999.

- COA planning Committee, Task Force on Electronic Presentation of SLIS Reports (team leader) 1997-1998
- COA Planning Committee, Subcommittee on Technology 1996-1998
- NLM practicum coordinator, 1997-1998
- Computer Literacy Workshops: Assisted with the development and evaluation of the workshops, 1996-1998
- Leader, Participation in the InterCat project, 1995-1997

APPENDIX B

MARC 21 Reference Materials

Part VII: A Summary of Commonly Used MARC 21 Fields
Part VIII: A List of Other Fields Often Seen in MARC Records
Part IX: The Leader
Part X: Field 008 for Books

Part VII:

A Summary of Commonly Used MARC 21 Fields

This is a summary of the MARC 21 tags used most frequently by libraries in entering their own bibliographic records. For full listings of all MARC 21 tags, indicators, and subfield codes, see *MARC 21 Format for Bibliographic Data*.

In the explanations on these pages:

Tags -- The tags (3-digit numbers) are followed by the names of the fields they represent. In this summary, and in the *MARC 21 Format for Bibliographic Data*, if a tag can appear more than once in one bibliographic record, it is labeled repeatable (R). If it can only be used once, it is labeled non-repeatable (NR). For example, a catalog record can have several subjects, so the tags for subject added entries (6XX) are labeled repeatable (R).

Indicators -- The use of indicators is explained in fields where they are used. Indicators are one-digit numbers. Beginning with the 010 field, in every field -- following the tag -- are two character positions, one for Indicator 1 and one for Indicator 2. The indicators are not actually defined in all fields, however. And it is possible that a 2nd indicator will be used, while the 1st indicator remains undefined (or vice versa). When an indicator is undefined, the character position will be represented by the character # (for blank space).

Subfield codes -- All the data in each field (beginning with the 010 field) is divided into subfields, each of which is preceded by a delimiter-subfield code combination. The most common subfield codes used with each tag are shown. Each subfield code is preceded by the character $, signifying a delimiter. The name of the subfield follows the code.

In general, every field MUST have a subfield 'a' (**$a**). One exception that is often seen is in Field 020 (ISBN), when the ISBN information (subfield **$a**) is unavailable but the price (subfield **$c**) is known. Some subfields are repeatable. In this summary, repeatability is noted for only the more common repeatable subfields.

Examples: Examples follow the explanation for each field. For clarity, one space has been placed between the tag and the first indicator, one space has been placed between the second indicator and the first delimiter- subfield code, and one space has been inserted between the delimiter-subfield code and the subfield data.

010 Library of Congress Control Number -- (LCCN)
(NR, or Not Repeatable
Indicators undefined.

Nintendo Exhibit 1029
Page 0074

Subfield used most often:

$a -- Library of Congress control number

```
Example:    010 ##  $a ###86000988#
```

020 International Standard Book Number -- (ISBN)
(R, or Repeatable)

Indicators undefined.

Subfields used most often:

$a -- International Standard Book Number
$c -- Terms of availability (often a price)
$z -- Cancelled/invalid ISBN (R)

```
Example:    020 ##  $a 0877547637
```

040 Cataloging source -- (NR)

Indicators undefined.

Subfields used most often:

$a -- Original cataloging agency
$c -- Transcribing agency
$d -- Modifying agency (R)

```
Example:    040 ## $a DLC
                   $c DLC
                   $d gwhs
```

100 Main entry -- Personal name -- (primary author)
(NR; there can be only one main entry)

Indicator 1: Type of personal name entry element

0 -- Forename
1 -- Surname (this is the most common form)
3 -- Family name

Indicator 2 undefined.

Indicator 2 became obsolete in 1990. Older records may display 0 or 1

Subfields used most often:

$a -- Personal name
$b -- Numeration
$c -- Titles and other words associated with a name (R)
$q -- Fuller form of name
$d -- Dates associated with a name (generally, year of birth)

```
Example:    100 1# $a Gregory, Ruth W.
```


Nintendo Exhibit 1029
Page 0075

```
                    $q (Ruth Wilhelme),
                    $d 1910–
```

130 Main entry -- Uniform title -- (NR)

Indicator 1: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Indicator 2 undefined.

Indicator 2 became obsolete in 1990. (See 100 above.)

Subfields used most often:

$a -- Uniform title
$p -- Name of part/section of a work (R)
$l -- Language of a work
$s -- Version
$f -- Date of a work

```
Example:    130 0# $a Bible.
                   $p O.T.
                   $p Psalms.
```

240 Uniform title (NR)

Indicator 1: Uniform title printed or displayed

0 -- Not printed or displayed
1 -- Printed or displayed (most common)

Indicator 2: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Subfields used most often:

$a -- Uniform title
$l -- Language of a work
$f -- Date of a work

```
Example:    240 10 $a Ile mystérieuse.
                   $l English.
                   $f 1978
```

245 Title Statement (NR)

Indicator 1: Title added entry

(Should the title be indexed as a title added entry?)

0 -- No title added entry

(indicates a title main entry; i.e. no author is given)

1 -- Title added entry

(the proper indicator when an author given in 1XX; the most common situation)

Indicator 2: Nonfiling characters

Nintendo Exhibit 1029
Page 0076

0-9 -- Number of nonfiling characters present, including spaces; usually set at zero, except when the title begins with an article; e.g., for *The robe*, the second indicator would be set to 4. The letters *T*, *h*, *e*, and the space following them are then ignored in alphabetizing titles. The record will be automatically filed under "*r*" -- for *Robe*.

Subfields used most often:

$a -- Title proper
$h -- Medium (often used for non-book media)
$p -- Name of part/section of a work (R)
$b -- Reminder of title (subtitles, etc.)
$c -- Remainder of title page transcription/Statement of responsibility

```
Example:    245 14 $a The DNA story :
                   $b a documentary history of gene
                      cloning /
                   $c James D. Watson, John Tooze.
```

246 Varying form of title (R)

Indicator 1: Note/title added entry controller
1 -- Note, title added entry
3 -- No note, title added entry

Indicator 2: Type of title
-- No information provided
0 -- Portion of title
1 -- Parallel title
4 -- Cover title
8 -- Spine title

Subfield used most often:
$a -- Title proper

```
Example:    246 3# $a Four corners power review
```

250 Edition statement (NR)

Indicators undefined.

Subfield used most often:
$a -- Edition statement

```
Example:    250 ## $a 6th ed.
```

260 Publication, distribution, etc. (Imprint) (R)

Indicator 1: Sequence of publishing statements
-- No information provided

Indicator 2: Undefined

Nintendo Exhibit 1029
Page 0077

Subfields used most often:

$a -- Place of publication, distribution, etc. (R)
$b -- Name of publisher, distributor, etc. (R)
$c -- Date of publication, distribution, etc. (R)

```
Example:    260 ## $a New York :
                   $b Chelsea House,
                   $c 1986.
```

300 Physical description (R)

Indicators undefined.

Subfields used most often:

$a -- Extent (number of pages) (R)
$b -- Other physical details (usually illustration information)
$c -- Dimensions (cm.) (R)
$e -- Accompanying material (for example, "teacher's guide" or "manual")

```
Example:    300 ## $a 139 p. :
                   $b ill. ;
                   $c 24 cm.
```

440 Series statement / Added entry--Title

This field was made obsolete in 2008 to simplify the series statement. See 490 and 830.

490 Series statement (No added entry is traced from field) (R)

Indicator 1: Specifies whether series is traced (whether an 8XX tag is also present)

0 -- Series not traced
1 -- Series traced (8XX is in record)

Indicator 2 undefined.

Subfield used most often:

$a -- Series statement (R)
$v -- Volume number (R)

```
Example:    490 1# $a Colonial American craftsmen
```

500 General note (R)

Indicators undefined.

Subfield used most often:

$a -- General note (Used when no specialized note field has been defined for the information. Examples: Notes regarding the index; the source of the title; variations in title; descriptions of the nature, form, or scope of the item.)

Nintendo Exhibit 1029
Page 0078

```
Example:    500 ##  $a Includes index.
```

504 Bibliography, etc. note (R)

Indicators undefined.

Subfield used most often:

$a -- Bibliography, etc. note

```
Example:    504 ## $a Includes bibliographical
                     references.
```

505 Formatted contents note (R)

Indicator 1: Type of contents note

0 -- Complete contents

1 -- Incomplete contents (used with multivolume set when some volumes are not yet published)

2 -- Partial contents

Indicator 2: Level of content designation

-- Basic

Subfield used most often:

$a -- Formatted contents note

```
Example:    505 0# $a Pride and prejudice -- Emma
                     -- Northanger Abbey.
```

520 Summary, etc. note (R)

Indicator 1: Display constant controller

-- Summary

1 -- Review

2 -- Scope and content

3 -- Abstract

Indicator 2 undefined

Subfields used most often

$a -- Summary, abstract, or annotation

$b -- Expansion of summary note

```
Example:    520 ## $a This basic guide to parliamentary
                     procedure tells how to conduct
                     and participate in a meeting
                     properly.
```

600 Subject added entry -- Personal name (R)

Indicator 1: Type of personal name entry element

0 -- Forename

Nintendo Exhibit 1029
Page 0079

1 -- Surname (this is the most common form)
3 -- Family name

Indicator 2: Subject heading system/thesaurus (identifies the specific list or file which was used)

0 -- Library of Congress Subject Headings
1 -- LC subject headings for children's literature
2 -- Medical Subject Headings
3 -- National Agricultural Library subject authority file
4 -- Source not specified
5 -- Canadian Subject Headings
6 -- Répertoire de vedettes-matière
7 -- Source specified in subfield $2

(Note regarding Sears subject headings: The MARC 21 format does not provide an assigned indicator for Sears subject headings. Therefore, an indicator of 7 is used, and the MARC defined code "sears" is placed in subfield $2.)

Subfields used most often:

$a -- Personal name (surname and forename)
$b -- Numeration
$c -- Titles and other words associated with a name (R)
$q -- Fuller form of name
$d -- Dates associated with a name (generally, year of birth)
$t -- Title of a work
$v -- Form subdivision (R)
$x -- General subdivision (R)
$y -- Chronological subdivision (R)
$z -- Geographic subdivision (R)
$2 -- Source of heading or term (used with 2nd indicator of 7)

```
Example:    600 10 $a Shakespeare, William,
                   $d 1564-1616
                   $x Comedies
                   $x Stage history.
```

```
Example:    600 10 $a Shakespeare, William,
                   $d 1564-1616
                   $x Knowledge
                   $z Rome
                   $v Congresses.
```

Notice that subfields $v, $x, and $z in the 600 field are repeatable. Subfields $v, $x, $y, and $z do not have to be in alphabetical order. They will be in the order prescribed by the instructions given by the subject heading system.

610 Subject added entry -- Corporate name (R)

*Indicator 1:*Type of corporate name entry element

Nintendo Exhibit 1029
Page 0080

0 -- Inverted name (not used with AACR2)
1 -- Jurisdiction name
2 -- Name in direct order

Indicator 2: Subject heading system/thesaurus.
See indicator 2 under 600

Subfields used most often:
$a -- Corporate name or jurisdiction name as entry element
$b -- Subordinate unit (R)
$v -- Form subdivision (R)
$x -- General subdivision (R)
$y -- Chronological subdivision (R)
$z -- Geographic subdivision (R)
$2 -- Source of heading or term (used with 2nd indicator of 7)

```
Example:    610 10 $a United States.
                   $b Army Air Forces
                   $v Biography.
```

650 Subject added entry -- Topical term (Most subject headings fit here.) (R)

Indicator 1: Level of subject
-- No information provided

Indicator 2: Subject heading system/thesaurus
(identifies the specific list or file which was used)
0 -- Library of Congress Subject Headings
1 -- LC subject headings for children's literature
2 -- Medical Subject Headings
3 -- National Agricultural Library subject authority file
4 -- Source not specified
5 -- Canadian Subject Headings
6 -- Répertoire de vedettes-matière
7 -- Source specified in subfield $2

Note regarding Sears subject headings: The MARC 21 format does not provide an assigned indicator for Sears subject headings. Therefore, an indicator of 7 is used, and the MARC defined code "sears" is placed in subfield $2.)

Subfields used most often:
$a -- Topical term
$v -- Form subdivision (R)
$x -- General subdivision (R)
$y -- Chronological subdivision (R)
$z -- Geographic subdivision (R)
$2 -- Source of heading or term used with 2nd indicator of 7)

Nintendo Exhibit 1029
Page 0081

```
Example:    650 #0 $a Theater
                   $z United States
                   $v Biography
                   $v Dictionaries.
```

Notice that subfields $v, $x, and $z in the 650 field are repeatable. Subfields $v, $x, $y, and $z do not have to be in alphabetical order. They will be in the order prescribed by the instructions given by the subject heading system.

651 Subject added entry -- Geographic name (R)

Indicator 1: undefined.

Indicator 2: Subject heading system/thesaurus.

See indicator 2 under 600

Subfields used most often:

$a -- Geographic name
$v -- Form subdivision (R)
$x -- General subdivision (R)
$y -- Chronological subdivision (R)
$z -- Geographic subdivision (R)
$2 -- Source of heading or term (used with 2nd indicator of 7)

```
Example:    651 #0 $a United States
                   $x History
                   $v Chronology.
```

Notice that subfields $v, $x, and $z in the 651 field are repeatable. Subfields $v, $x, $y, and $z do not have to be in alphabetical order. They will be in the order prescribed by the instructions given by the subject heading system.

700 Added entry -- Personal name (R)

Indicator 1: Type of personal name entry element

0 -- Forename
1 -- Surname (this is the most common form)
3 -- Family name

Indicator 2: Type of added entry

-- No information provided (most common; co-authors, editors, etc.)
2 -- Analytical entry (The values for Indicator 2 changed in 1994 with Format Integration, and older records may display additional values. An analytical entry involves an author/title of an item contained in a work.)

Subfields used most often:

$a -- Personal name
$b -- Numeration
$c -- Titles and other words associated with a name (R)
$q -- Fuller form of name
$d -- Dates associated with a name (generally, year of birth)
$e -- Relator term (such as ill.) (R)
$4 -- Relator code (R)

Nintendo Exhibit 1029
Page 0082

```
Example:    700 1# $a Baldridge, Letitia.
```

710 Added entry -- Corporate name (R)

Indicator 1: Type of corporate name entry element
- 0 -- Inverted name (not used with AACR2)
- 1 -- Jurisdiction name
- 2 -- Name in direct order

Indicator 2: Type of added entry.
- See Indicator 2 under 700
- # -- No information provided
- 2 -- Analytical entry

Subfields used most often:
- **$a** -- Corporate name or jurisdiction name as entry element
- **$b** -- Subordinate unit (R)

```
Example:    710 2# $a Sunburst Communications (Firm)
```

740 Added entry -- Uncontrolled related/analytical title (R)

Indicator 1: Nonfiling characters
- 0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

*Indicator 2:*Type of added entry. See Indicator 2 under 700
- # -- No information provided
- 2 -- Analytical entry
- (This field was redefined in 1994 with Format Integration. Prior to 1994, the field was also used for variant titles, such as a different wording on a spine title. In records created since Format Integration, those variant tiles appear in a 246 field.)

Subfield used most often:
- **$a** -- Title

```
Example:    740 02 $a Uncle Vanya.
```

800 Series added entry -- Personal name (R)

Indicator 1: Type of personal name entry element
- 0 -- Forename
- 1 -- Surname
- 3 -- Family name

Indicator 2 undefined.

Subfields used most often:
- **$a** -- Personal name
- **$b** -- Numeration
- **$c** -- Titles and other words associated with a name (R)
- **$q** -- Fuller form of name

Nintendo Exhibit 1029
Page 0083

$d -- Dates associated with a name (generally, year of birth)
$t -- Title of a work (the series)
$v -- Volume number

```
Example:    800 1# $a Fisher, Leonard Everett.
                   $t Colonial American craftsmen.
```

830 Series added entry -- Uniform title (R)

Indicator 1 undefined.

Indicator 2: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Subfield used most often:

$a -- Uniform title
$v -- Volume number

```
Example:    830 #0 $a Railroads of America (Macmillan)
```

[Back to Top of Page]

Part VIII:

A List of Other Fields Often Seen in MARC Records

001	Control number
003	Control number identifier
005	Date and time of latest transaction
006	Fixed-length data elements -- additional material characteristics
007	Physical description fixed field
008	Fixed length data elements (See Part X)
022	International Standard Serial Number (ISSN)
037	Source of acquisition
041	Language code
043	Geographic area code
050	Library of Congress call number
060	National Library of Medicine call number
082	Dewey Decimal classification number (the one recommended by the Library of Congress; locally-assigned call numbers may appear elsewhere)
110	Main entry -- Corporate name (less frequent under AACR2 rules)
256	Computer file characteristics
263	Projected publication date (indicates a CIP -- Cataloging in Publication -- record)

Nintendo Exhibit 1029
Page 0084

306 Playing time
508 Creation/production credits note
510 Citation/references note (review sources)
511 Participant or performer note
521 Target audience note (first indicator: 0 = reading grade level, 1 = interest age level, 2 = interest grade level, 3 = special audience characteristics, 4 = motivation interest level)
530 Additional physical form available note
538 System details note
586 Awards note
656 Index term -- Occupation
730 Added entry -- Uniform title
852 Location
856 Electronic location and access
9XX Reserved for local use. (They are used by vendors, systems, or individual libraries to exchange additional data)

[Back to Top of Page]

Part IX:

The Leader

There are 24 positions in the Leader, numbered from 00 to 23. For fuller explanation, see the *MARC 21 Format for Bibliographic Data*.

00-04 Record length (calculated by the computer for each record)
05 Record status
- a = increase in encoding level
- c = corrected or revised
- d = deleted
- n = new
- p = increase in encoding from prepublication (previous CIP)

06 Type of record
- a = language material
- c = printed music
- d = manuscript music
- e = cartographic material
- f = manuscript cartographic material
- g = projected medium
- i = nonmusical sound recording
- j = musical sound recording
- k = 2-dimensional nonprojectable graphic
- m = computer file
- o = kit
- p = mixed materials
- r = 3-dimensional artifact or naturally occurring object
- t = manuscript language material

07 Bibliographic level
- a = monographic component part
- b = serial component part

Nintendo Exhibit 1029
Page 0085

c = collection
d = subunit
i = integrating resource
m = monograph/item
s = serial

08 Type of control
= no specified type
a = archival

09 Character coding scheme
= MARC-8
a = UCS/Unicode

10 Indicator count (always "2")

11 Subfield code count (always "2")

12-16 Base address of data (calculated by the computer for each record)

17 Encoding level
= full level
1 = full level, material not examined
2 = less-than-full level, material not examined
3 = abbreviated level
4 = core level
5 = partial (preliminary) level
7 = minimal level
8 = prepublication level (CIP)
u = unknown
z = not applicable

18 Descriptive cataloging form
= non-ISBD
a = AACR2
i = ISBD
u = unknown

19 Multipart resource record level
= Not specified or not applicable
a = Set
b = Part with independent title
c = Part with dependent title

20 Length of the length-of-field portion (always "4")

21 Length of the starting-character-position portion (always "5")

22 Length of the implementation-defined portion (always "0")

23 Undefined (always "0")

[Back to Top of Page]

Part X:

Field 008 for Books

Field 008 is used for Fixed Length Data Elements ("Fixed Field Codes"). There are 40 character positions in field 008, numbered from 00-39. Undefined positions must contain either a blank (#) or a fill character (|). Positions 00-17 and 35-39 are defined the same way for all media.

Nintendo Exhibit 1029
Page 0086

The information shown here for positions 18-34 applies only to books. For explanation of all the positions below and for positions 18-34 for other media, see the *MARC 21 Format for Bibliographic Data*.

Note that field 008 has no indicators or subfield codes.

00-05 Date entered on file (YYMMDD),
where Y=year, M=month, and D=day

06 Type of date/publication status:
- b = no dates given; B.C. date involved
- e = detailed date
- s = single known date/probable date
- m = multiple dates
- r = reprint/reissue date (Date 1) and original date (Date 2)
- n = dates unknown
- q = questionable date
- t = publication date and copyright date
- | = no attempt to code

07-10 Date 1/beginning date of publication

11-14 Date 2/ending date of publication

Date fields contain the year(s) of publication. The type of date(s) in these elements are specified in fixed field element 06: Type of date/publication status. (For further details, see the field 008 description in the *MARC 21 Format for Bibliographic Data*.)

15-17 Place of publication, production, or execution
For example:
- pk# = Pakistan
- cau = California (US)

(For a full list of codes used in these positions, see the *MARC Code List for Countries*.)

18-21 Illustrations (up to 4 codes):
- # = no illustrations
- a = illustrations
- b = maps
- c = portraits
- d = charts
- e = plans
- f = plates
- g = music
- h = facsimiles
- i = coats of arms
- j = genealogical tables
- k = forms
- l = samples
- m = phonodisc, phonowire, etc.
- o = photographs
- p = illuminations
- | = no attempt to code

22 Target audience:
- # = unknown or not specified
- a = preschool

Nintendo Exhibit 1029
Page 0087

b = primary
c = pre-adolescent
d = adolescent
e = adult
f = specialized
g = general
j = juvenile
| = no attempt to code

23 Form of item:
= none of the following
a = microfilm
b = microfiche
c = microopaque
d = large print
f = braille
r = regular print reproduction
s = electronic
| = no attempt to code

24-27 Nature of contents (up to 4):
= no specified nature of contents
a = abstracts/summaries
b = bibliographies (is one or contains one)
c = catalogs
d = dictionaries
e = encyclopedias
f = handbooks
g = legal articles
i = indexes
j = patent document
k = discographies
l = legislation
m = theses
n = surveys of literature
o = reviews
p = programmed texts
q = filmographies
r = directories
s = statistics
t = technical reports
u = standards/specifications
v = legal cases and notes
w = law reports and digests
z = treaties
| = no attempt to code

28 Government publication:
= not a government publication
i = international intergovernmental
f = federal/national
a = autonomous or semi-autonomous component
s = state, provincial, territorial, dependent, etc.
m = multistate
c = multilocal

Nintendo Exhibit 1029
Page 0088

l = local
z = other type of government publication
o = government publication -- level undetermined
u = unknown if item is government publication
| = no attempt to code

29 Conference publication:
0 = not a conference publication
1 = conference publication
| = no attempt to code

30 Festschrift:
0 = not a festschrift
1 = festschrift
| = no attempt to code

31 Index:
0 = no index
1 = index present
| = no attempt to code

32 Undefined (since 1990) (Earlier records may contain the values 0 or 1)
= Undefined
| = no attempt to code

33 Literary form:
0 = not fiction (not further specified)
1 = fiction (not further specified)
c = comic strips
d = dramas
e = essays
f = novels
h = humor, satires, etc.
i = letters
j = short stories
m = mixed forms
p = poetry
s = speeches
u = unknown
| = no attempt to code

34 Biography:
= no biographical material
a = autobiography
b = individual biography
c = collective biography
d = contains biographical information
| = no attempt to code

35-37 Language:
A three-letter code. For example: eng fre ger spa rus ita

(For a full list of codes used in these positions, see the *MARC Code List for Languages*.)

38 Modified record:
= not modified
x = missing characters (because of characters unavailable in MARC character set)
s = shortened
d = "dashed-on" information omitted

Nintendo Exhibit 1029
Page 0089

r = completely romanized/printed cards in script
o = completely romanized/printed cards romanized
| = no attempt to code

39 Cataloging source:
= national bibliographic agency
c = cooperative cataloging program
d = other sources
u = unknown
| = no attempt to code

[Back to Top of Page]

[Back to Table of Contents] -- [Continue to Part 11]



Library of Congress
Library of Congress Help Desk (10/27/2009)

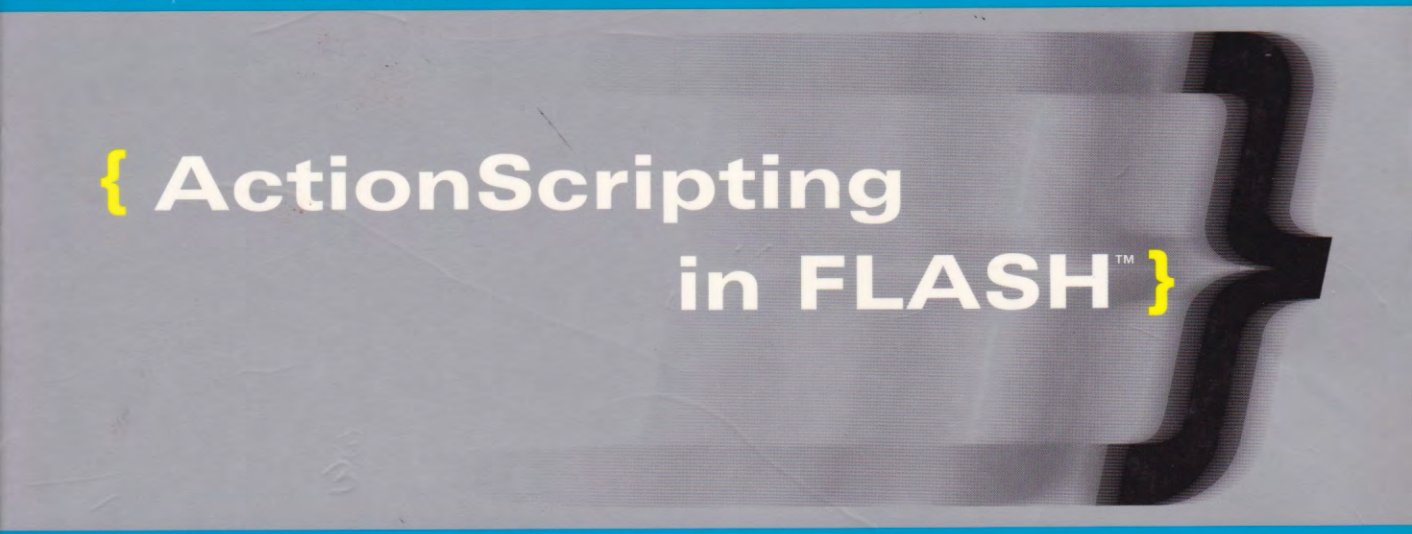
Nintendo Exhibit 1029
Page 0090

APPENDIX 1014A

Phillip Kerman

```
onClipEvent (load) {
        stop ();
        pageDisplay="Page "+1+" of "+ _totalFrames
}
onClipEvent (enterFrame) {
```

{ ActionScripting in FLASH™ }



```
        _alpha=_alpha+alphaChange;

        if (_alpha<1) {
                gotoAndStop (destinationFrame);
                alphaChange=5;
                pageDisplay="Page "+_currentFrame+" of "+_totalFrames;
        }

        if (_alpha>100) {
                alphaChange=0;
        }
}
```

SAMS

Nintendo Exhibit 1029
Page 0092

Phillip Kerman

{ ActionScripting in FLASH™ }

SAMS 201 West 103rd St. • Indianapolis, Indiana, 46290 USA

Nintendo Exhibit 1029
Page 0093

ActionScripting in Flash

Copyright © 2001 by Phillip Kerman

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-672-32078-9

Library of Congress Catalog Card Number: 00-108995

Printed in the United States of America

First Printing: April 2001

04 03 02 01 4 3 2

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Sams Publishing cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Executive Editor
Jeff Schultz

Development Editors
Kate Small
Alice Martina Smith

Managing Editor
Charlotte Clapp

Project Editor
Carol Bowers

Copy Editors
Michael Henry
Alice Martina Smith

Indexer
Sheila Schroeder

Proofreader
Daniel Ponder

Technical Editor
Lynn Baus

Team Coordinator
Amy Patton

Interior Designer
Alan Clements

Cover Designer
Alan Clements

Page Layout
Ayanna Lacey
Heather Hiatt Miller
Stacey Richwine DeRome

Overview

Part I Foundation 1

{ Foundation } Introduction **3**

1 Flash Basics **7**

2 What's New in Flash 5 **25**

3 The Programmer's Approach **37**

4 Basic Programming in Flash **47**

5 Programming Structures **73**

6 Debugging **113**

7 The Movie Clip Object **141**

8 Functions **161**

9 Selecting Text, Trapping Keys, and Manipulating Strings **185**

10 Arrays **211**

11 Objects **237**

12 Homemade Objects **261**

13 Smart Clips **283**

14 Interfacing with External Data **303**

Part II Workshop 351

{Workshop} Introduction **353**

Part IIA Beginning Workshops 357

1 Ensuring That Users Have the Flash 5 Player **359**

2 Faking Video **367**

3 Creating Custom Cursors **375**

4 Creating a Horizontal Slider **389**

5 Building a Slide Show **403**

6 Mapping **413**

7 Working with Odd-Shaped Clickable Areas **423**

8 Adapting Built-in Smart Clips **431**

Part IIB Intermediate Workshops 439

9 Creating a Currency-Exchange Calculator **441**

10 Creating a ToolTip Smart Clip **461**

11 Creating a Digital Timer **473**

12 Creating an Analog Timer **481**

13 Creating a Countdown Timer **489**

14 Using Math to Create a Circular Slider **497**

15 Developing Time-Based Animations **505**

16 Creating a Multistate Button **511**

Part IIC Advanced Workshops 521

17 Offline Production **523**

18 Creating a Dynamic Slide Presentation **529**

19 Creating JavaScript Cookies **543**

20 Writing JavaScript Inter-Movie Communications **551**

21 Fixing Broken Scripts **559**

Part III Appendices 573

A Equivalents **575**

B Making Flash Extensions for the Macromedia Exchange Web Site **583**

Index **589**

Contents

Part I Foundation 1

{ Foundation } Introduction 3

1 Flash Basics 7

Timeline Hierarchy 8
Implications for Animation and Filesize 8
Implications for Programming 10
Targeting 10
Script Locations 14
Always Movie Clips 17
Coordinate System 18
Tricks of the Trade 20
Invisible Buttons 20
Empty Movie Clips 21
Empty Layers and Keyframes 23
Summary 24

2 What's New in Flash 5 25

Advanced Features Making Programming Easy 25
Clip Events 26
Smart Clips 26
External Scripts 28
Custom Functions 29
Objects 29
HTML Text 30
Previous Knowledge Worth Forgetting 31
Deprecated ActionScript 31
Non-Deprecated ActionScripts That Should Be 33
Old Tricks That Are Old News 34
Summary 36

3 The Programmer's Approach 37

Specification 37
Prototyping 38
Hard Wiring 40
Pseudo-Code 40
Good Style 41
Less Is More 41
Comments 42
Magic Numbers, Constants, and Variables 44
Repeated Code 45

Nintendo Exhibit 1029
Page 0097

Code Data Separation 45
Summary 46

4 Basic Programming in Flash 47

Terminology, Special Characters, and Formatting 47
Events 48
Results of Events 49
Terms 50
Special Characters and Formatting 56
Data Types and Variables 58
Variables' Names and Values 58
String and Number Data Types 58
Other Data Types 61
Using Variables 65
Assigning and Accessing Variables 65
Comparing and Passing Values 66
Scope and Variable Collision 68
Dot Syntax 70
Summary 71

5 Programming Structures 73

Statements, Expressions, and Operators 73
Writing Expressions 74
Using Operators in Expressions 77
Types of Statements 84
Built-in Statements 87
Simple Objects in Statements 89
Using the Math Object 89
Using the Number Object 94
Conditional and Loop Statements 96
Conditional Statements: `if`, `if else`, `if else if` 97
Loop Statements: `for`, `for in` 101
`while` 105
Applied Expression Writing 109
Summary 112

6 Debugging 113

General Approaches to a Bug-Free Life 114
Finding and Defining Bugs 114
Fixing Bugs 119
Preventing Bugs 120
Using the Debugger 122
Viewing and Setting Properties and Variables 122
Watching Variables 125

Nintendo Exhibit 1029
Page 0098

Remote Debugging 126
Strategies of Debugging 128
Summary 138

7 The Movie Clip Object 141

Properties of Clips 142
You Can Get Them All, But You Can't Set Them All 143
Anonymous Targeting 144
Variables in Clips (or "Homemade Properties") 145
Methods of Clips 147
Referencing Clips and Targeting 151
Relative and Absolute Referencing 151
Dynamic Referencing 155
Summary 158

8 Functions 161

How to Use Functions 162
Using Built-in Functions 162
Using Homemade Functions 164
Creating Homemade Functions 166
Basics 166
Functions as Subroutines 167
Making Functions That Accept Parameters 170
Making Functions That Return Values 173
Using Functions as Methods 175
Local Variables 177
Applying Functions to Previous Knowledge 179
Review Built-in Functions 180
Things to Remember 181
Summary 184

9 Selecting Text, Trapping Keys, and Manipulating Strings 185

String Object Form 186
Esoteric String Object Details 187
Methods of the String Object Explored 189
Zero-Based System 189
Extracting Portions of Strings 190
Extracting Characters, Changing Case, and Searching 191
String Object Methods Applied 192
Using HTML Text 196
Selection Object 198
Getting and Setting Focus 200
Setting and Getting Selections 201

Nintendo Exhibit 1029
Page 0099

Key Object 202
Using the Key Object 202
Key Object Examples 205
Summary 210

10 Arrays 211

Array Overview 212
Array Creation and Manipulation 215
Creating and Populating Arrays 215
Accessing Array Contents 217
Array Object Methods 222
Associative Arrays 232
Summary 235

11 Objects 237

Formal Rules of Objects 238
Attach Sound 239
Sound Object Basics 239
Advanced Sound Controls 241
Controlling Multiple Sounds 243
Color 245
Simple Coloring 245
Using RGB Values 246
Using the Color Transform Method 248
Date 250
Instantiating a Date 251
Manipulating Dates 251
Attach Movie Clip 257
Summary 259

12 Homemade Objects 261

Basic Objects 262
Using Constructor Function 262
Making Methods 264
Inheritance 269
Practical Example of Homemade Objects 273
Summary 281

13 Smart Clips 283

Standard Smart Clips 284
Making Smart Clips 285
Advanced Applications for Standard Smart Clips 292
Replacing the Clip Parameters Panel with Custom UIs 294
Designing Custom UIs 295
Building Custom UIs 296
Summary 300

Nintendo Exhibit 1029
Page 00100

14 Interfacing with External Data 303
External Scripts 304
External Data Files 306
Waiting for Variables to Load 307
Data File Format 309
Examples of Using External Data Files 310
Server Scripts 314
XML 318
JavaScript 327
Quick and Dirty Method 328
JavaScript Talking to Flash 329
Targeting the Flash Object 333
Workarounds 335
Director and Authorware 338
Flash in Authorware 339
Flash in Director 344
Summary 349

Part II Workshop 351

{Workshop} Introduction 353

Part IIA Beginning Workshops 357

1 Ensuring That Users Have the Flash 5 Player 359

2 Faking Video 367

3 Creating Custom Cursors 375

4 Creating a Horizontal Slider 389
Creating a Quick and Dirty Slider 389
Converting the Slider into a Smart Clip 394
Summary 401

5 Building a Slide Show 403

6 Mapping 413

7 Working with Odd-Shaped Clickable Areas 423

8 Adapting Built-in Smart Clips 431

Part IIB Intermediate Workshops 439

9 Creating a Currency-Exchange Calculator 441

10 Creating a ToolTip Smart Clip 461

Nintendo Exhibit 1029
Page 00101

11 Creating a Digital Timer 473

12 Creating an Analog Timer 481

13 Creating a Countdown Timer 489

14 Using Math to Create a Circular Slider 497

15 Developing Time-Based Animations 505

16 Creating a Multistate Button 511

Part IIC Advanced Workshops 521

17 Offline Production 523

18 Creating a Dynamic Slide Presentation 529

19 Creating JavaScript Cookies 543

20 Writing JavaScript Inter-Movie Communications 551

21 Fixing Broken Scripts 559

Card Flip 560
Hint 560
Solution 561
Card Snap 561
Hint 561
Solution 562
Circle Move 562
Hint 562
Solution 563
Multiple Choice 563
Hint 564
Solution 564
Rotating Box 564
Hint 564
Solution 565
Word Float 565
Hint 566
Solution 566
Yellow Box 566
Hint 567
Solution 567
Click and Hold 567
Hint 568
Solution 568

Nintendo Exhibit 1029
Page 00102

Objects 569
Hint 569
Solution 570
Move Multiple. 570
Hint 571
Solution 571
Summary. 571

Part III Appendices 573

A Equivalents 575

Alternatives to Deprecated Code 575
`tellTarget()` 575
`call()` 576
`ifFrameLoaded()` 576
`toggleHighQuality()` 576
`eval()` 577
Logical Operators (`and`, `or`, and `not`) 577
String Comparison Operators (`eq`, `ge`, `gt`, `le`, `lt`, and `ne`) 577
String Functions. 577
`int()` 578
`random()` 578
No More Funkiness. 578
Two-Frame "Updater" Movie Clips 579
Drag Invisible Movie Clip to Ascertain Mouse Position 579
Pseudo Arrays 580
URL-Encoded Data Instead of XML-Structured. 580
The Oldest Ones in the Book. 580
You Can't Set `_currentframe`. 580
You Can Jump Only to Integer Frame Numbers 580
There's a Big Difference Between = and == 580
For-Loops Use Semicolons, Not Commas 581

B Making Flash Extensions for the Macromedia Exchange Web Site 583

How It Works 583
The MXI File Format 585

Index 589

Nintendo Exhibit 1029
Page 00103

{Part I }

Foundation

1 Flash Basics

2 What's New in Flash 5

3 The Programmer's Approach

4 Basic Programming in Flash

5 Programming Structures

6 Assigning Values

7 The Movie Clip Object

8 Functions

9 Selecting Text, Trapping Keys, and Manipulating Strings

10 Arrays

11 Objects

12 Homemade Objects

13 Smart Clips

14 Interfacing with External Data

Nintendo Exhibit 1029
Page 00104

{ Foundation } Introduction

The following 14 chapters cover practically every detail of ActionScript. Although the content is organized like a textbook, I've included several examples along the way. You are encouraged to break from the reading and try out any topic that interests you. Generally, however, this is the part of the book that you read. In the second half (the workshops), you can follow along with the 21 detailed tutorials.

Here's a quick rundown of the topics explored in this part of the book:

1. "Flash Basics" includes the prerequisites that I expect every reader to bring with him or her. Even if you consider yourself a Flash expert, you should read this chapter as both a review and an insight into some of the terms that I'll use throughout the book.
2. "What's New in Flash 5" introduces you to the key features in Flash 5 that pertain specifically to ActionScript. In addition to changes in the programming language, features that affect how you build a Flash movie are discussed. Finally, this chapter includes a quick rundown of previous knowledge (that you might have acquired in older versions of Flash) that is best forgotten.
3. "The Programmer's Approach" is a very general chapter that lays the groundwork for your programming career. Topics such as writing specifications, prototyping, and exactly what "good style" means are covered.
4. "Basic Programming in Flash" introduces you to the terminology and basic elements of ActionScript, such as data types and variables. It's impossible to discuss these elements without showing how they work, but the goal is to simply introduce all the pieces that will be incorporated in later chapters.

Nintendo Exhibit 1029
Page 00106

5. "Programming Structures" is a huge chapter that explains all the ways ActionScript is structured. If Chapter 4 was the building materials (like wood, bricks, and concrete), this chapter is the framework and architectural styles. You also get a peek at both the Math and Number objects because they're so integral to the structural elements covered.
6. "Assigning Values" offers a chance to catch your breath (after Chapter 5) and take the time to learn ways to ensure quality programming before you go too far in the wrong direction. The new Flash Debugger is explained, as well as some general programming techniques to avoid or remove bugs.
7. "The Movie Clip Object" introduces a familiar component of Flash, but in a way that will help you understand other "objects" that come up in later chapters. In this way, you can leverage your existing knowledge when learning advanced topics.
8. "Functions" shows you how to use the built-in functions as well as how to write your own functions. It turns out that homemade functions prove to be much more involved than the ones that come with Flash. This is possibly the most valuable chapter because it can save you a ton of time.
9. "Selecting Text, Trapping Keys, and Manipulating Strings" looks at ways to control text both on the screen (such as HTML text and input text) as well as off the screen (such as cleaning up strings before the user sees them). Also, you'll learn how to use the Key object to "trap" user interaction with the keyboard.
10. "Arrays" explores how to make, access, and manipulate arrays, which are simply a great way to organize complex information.
11. "Objects" introduces the general form of objects, shows you how to use the built-in objects Sound, Color, and Date, and teaches you new ways to use the familiar Movie Clip.
12. "Homemade Objects" shows you how to apply knowledge you already have to make complex objects in Flash. If arrays are a way to store complex information easily, objects are a way to store *really* complex information.
13. "Smart Clips" walks through all the ways to build and use Smart Clips, from ways of making standard Smart Clips work for you to creating Custom User Interfaces (Custom UIs).
14. "Interfacing with External Data" shows you many ways Flash can "talk" to outside applications. Topics include reading text files, interacting with server applications, exchanging XML-structured data, invoking and being controlled by JavaScript, and interacting with the host applications Authorware and Director. I didn't have time in this chapter to actually show you how to use all these other tools, but it shows you practically everything else.

Nintendo Exhibit 1029
Page 00107

It's amazing to think that there's so much to say about Flash and I don't even say it all. I could probably double the size of this book and there'd be still more! However, I'm sure that if you grasp all the content I've organized in these 14 chapters, you'll be able to adapt quickly to any new situation that arises.

Nintendo Exhibit 1029
Page 00108

{ Chapter 1 }

Flash Basics

Flash professionals are a strange breed. Some have traditional animation and graphics skills. Others come to Flash from a programming background. Still others are so young they don't have *any* professional background—they're straight out of college (or younger)! It doesn't matter where you come from because Flash is approachable and powerful.

For this book to serve as the bridge between intermediate Flash and advanced programming, it is best if you start at the same level. Intermediate Flash users, who are familiar with drawing, tweening, and sound effects, are about to embark on programming; already experienced programmers are about to apply their skills to Flash. But we all need to start at the same level.

This chapter is an important link to the material that follows. It isn't a recap of drawing, tweening, and sound effects; rather, it's a quick overview of foundational knowledge unique to Flash scripting. If this material looks familiar, good. If not, you should make sure that it all makes sense before attacking the rest of the book.

Specifically, this chapter covers

- The timeline hierarchy of nested Movie Clips
- The Stage coordinate system
- Traditional Flash tricks such as invisible Buttons and empty Movie Clips
- The places where scripting occurs

Nintendo Exhibit 1029
Page 00110

I'd like to think most of this chapter is a review, but it's alright if some material is new to you. If nothing else, you should begin to approach the concepts in the way they're presented here. For example, if you've never heard of invisible Buttons, that's fine—just try to start using them.

Timeline Hierarchy

A key concept that is critical to understanding Flash is nested timelines. Any time you select a shape and Insert, Convert to Symbol... (F8), the shape is placed in the Library and you're left with an instance of the symbol on stage. You can also select an instance of a symbol and Insert, Convert to Symbol..., which places an instance of the selected symbol inside a new symbol. The main thing to remember is that F8 takes whatever is selected and places it in the Library as a new symbol (even if you have a symbol instance selected). Such nesting of symbols has implications for both programming and animation structure.

Implications for Animation and Filesize

To see how nesting symbols applies to animation, consider how to create a symbol of a car containing moving wheels. First, draw a wheel and convert it to a Graphic Symbol. To create a rotating wheel symbol, you must convert an instance of the wheel Graphic Symbol to a Movie Clip. Then edit the contents of the Movie Clip by making a simple Motion Tween that rotates the wheel symbol. Because only symbol instances can be used in a Motion Tween, the extra step of placing the wheel symbol inside the "wheel in motion" symbol is necessary. Finally, use two instances of the "wheel in motion" clip inside a third clip, "car." Then you can animate the car across the main timeline.

In the end, your car is a symbol containing two instances of the "wheel in motion" clip, which both contain an instance of the plain wheel. When building such nested symbols, it's usually best to work from the "inside out" or "specific to general." However you approach it, be sure to monitor the address bar that appears above the stage as shown in Figure 1.1. The address bar begins with the name of the scene that you're currently editing, and shows the hierarchy of symbol nesting as well. In Figure 1.1, the address bar indicates that you're editing the "Wheel" symbol inside "Wheel in Motion," which in turn is nested inside "Car." Of course, you can make more complex nested symbols than a car with rotating wheels—you just have to keep track of what you're doing.

Nintendo Exhibit 1029
Page 00111

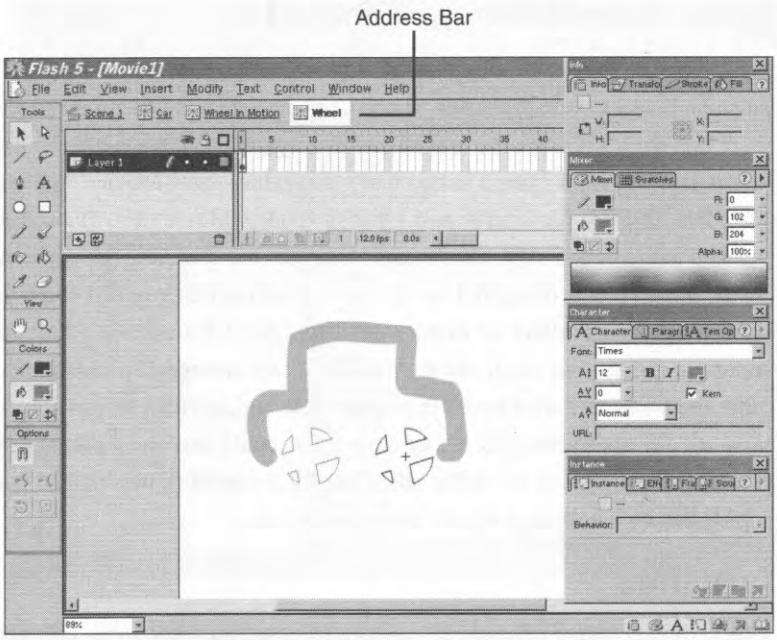


Figure 1.1 *The address bar shows the hierarchy of nested symbols.*

Besides enabling complex effects, nested symbols can reduce your movie's file-size by recycling graphic components. You can actually take this seeming advantage too far and it will begin to work against you. For example, take the absurd example of a single-pixel symbol that is recycled and nested to make a line…and then four lines are used to make a square…and so on. Generally, however, the benefits of recycling symbols are significant.

In addition to filesize savings, careful use of symbols generally (and Movie Clips specifically) can help your productivity. For example, when a change is made to the master symbol, that change is reflected in each instance already in use. Also, when you duplicate a shape that hasn't been converted to a symbol, you encounter two problems: the obvious filesize contribution and the fact that Flash might not render the "identical" shapes the same way. Because you can position graphics in fractional locations, Flash often needs to round off when drawing every pixel. The bottom line is that two different shapes of the same size can easily appear slightly differently whereas multiple symbols look the same (unless they're scaled differently).

Nintendo Exhibit 1029
Page 00112

Implications for Programming

Finally, nested symbols can be useful for programming tasks. Consider, for example, how a simple draggable Movie Clip is created. If you want to write a script that responds when the user begins to drag (mouse press), you can use a button. But you can't make a button that's draggable—only Movie Clips are draggable. Instead, you simply nest a button inside a Movie Clip. The script attached to the button says, effectively, "drag me" but it's really the clip containing the button that is dragged. It's not always required that you nest symbols to achieve a particular effect (as in the case of dragging), but often it's desirable because it makes your Flash file more modular and manageable. Imagine if our entire money supply were based in pennies. It would certainly be possible to carry out any transaction, but it'd be a pain. You could nest one hundred pennies in a "dollar" symbol. A ten-dollar bill is like 10 instances of the "dollar" symbol. Sometimes such nesting is simply more convenient.

Targeting

When creating nested symbols, it's important to pay attention to the address bar. When programming, it's just as important to understand the concept of *targeting*. That is, which clip you want to affect. In the case of a button inside a draggable Movie Clip (as earlier), the "drag me" instructions really target the clip in which it is contained. By default, if you don't indicate otherwise, the "current clip" (the one in which the script is contained) is targeted. If you want to target a different clip, you just need to clarify which clip.

One way to envision targeting is to think of the folder (and subfolder) structure on your computer. You can have folders inside folders just like symbols inside symbols. (I'll use this analogy throughout the description and tie it to Flash at the end of this section.) If you are browsing one folder and you want to open a file in a subfolder, you simply "target" the subfolder you want to browse. When you decide to target another file or folder, you have two basic ways to do so: relatively and absolutely. Take a quick look at Figure 1.2. Starting from the folder "Flash 5," if you want to go into a folder "Help" and then into a subfolder "UsingFlash," the relative target can be expressed as *go into the folder "Help" and then into the subfolder "UsingFlash."* Consider that you can only be "in" (browsing) one folder at a time, so when you begin by saying *go into the folder "Help,"* you're assuming there's a folder called "Help" present within the current folder. This is relative targeting because it's relative to where you are.

Nintendo Exhibit 1029
Page 00113

the tool tips. Ideally, you would keep *all* the text for all the tool tips in one location to make translation that much easier. The main idea is that you want to be able to make significant changes to either the code or data without affecting the other.

You can think of code data separation as a form of modularization. There are other forms of modularization—including Flash's `LoadMovie()`, which enables you to play separate .swf files within a larger movie. Modularization has many benefits in addition to those mentioned for code data separation. For one thing, by modularizing your Flash movie, users won't have to wait for the entire site to download. They can selectively download just the portions in which they are interested. Also, modularizing makes working with others easy and efficient. Consider that if you just had one master file for the entire site, only one person could work at a time. So, there are a ton of benefits to code data separation and other kinds of modularization. Without providing a lot of details now, just realize that throughout this book I'll try to emphasize solutions that exhibit such modular attributes.

Summary

This chapter explored the attributes that make up the "programmer way." In my experience, it seems that programmers tend to fit the same profile. For example, they often work in darkened offices that lack windows and subsist on soft drinks.

You don't have to become a geek to be a good programmer. Just concentrate on the approach discussed in this chapter. Try to develop a good style by striving to write concise code with lots of comments and avoid magic numbers and repeated code. Realize, too, that your programming style should continually improve. The best programmers in the world know they have room to improve further.

The process you undertake can also make programming easier (and better). Creating a specification and quickly producing prototypes might seem like additional up-front work, but they will save you time later. Finally, always try to separate code from data. In no time, you'll start "feeling it" and before you know it, you can call yourself a "programmer" with pride.

Nintendo Exhibit 1029
Page 00114

{ Chapter 4 }

Basic Programming in Flash

Although you don't have to know traditional programming concepts to use ActionScript, they will help you greatly in Flash. If you're experienced in another programming language, this chapter might look familiar to you. Flash's ActionScript language is based on the same standard used by JavaScript (called *ECMA-262**). As such, many aspects are similar or identical. This chapter is a good introduction for both experienced programmers as well as those with basic Flash skills. I'll explain some general programming concepts, but with specific attention to how they apply to Flash.

In this chapter, you will

- Learn basic programming terms
- Recognize built-in script elements
- Get an overview of traditional concepts such as variables and data types

Terminology, Special Characters, and Formatting

The terms and rules in programming are very strict. This can actually be helpful. Unlike in English where the meaning of a sentence can be ambiguous or vague,

*ECMA stands for The European Computers Manufacturers Association (see `www.ecma.ch`).

Nintendo Exhibit 1029
Page 00115

in programming there are absolutes. After you fully understand what a "property" is, for example, you never need to wonder what kind of property—as you would with the word *property* in English, which has multiple meanings. Not only do terms in programming have absolute meanings, but they're usually closely related to the word's meaning in English. Be careful though: Sometimes your first impression of a term could carry special meaning to you that is unrelated to the true meaning. Luckily, most terms are very easy to learn.

Events

Everything that happens in Flash is the result of an event that causes a script to execute. Even for such a simple example as the user clicking a button that causes the movie to skip ahead, the instructions (to skip ahead) resulted from the click "mouse event."

There are three types of events in Flash: keyframes, mouse events, and clip events. Notice that these correlate to the three places you can place scripts (keyframes, button instances, and clip instances). If a script is placed in a keyframe, the script will execute as soon as Flash reaches the keyframe. In the case of buttons, you must always specify the mouse event to which you want to respond. Similarly, scripts on clip instances need to include a clip event. Figure 4.1 shows the available mouse events and clip events.

I like to think of an event as the wrapper of the scripts that it contains. Consider that you can't just have a `stop()` Action attached to a button—it must be wrapped within a mouse event. I suppose that it makes sense to think the `stop()` Action *follows* the mouse event. But because you can have several Actions follow one event (maybe the button causes both `stop()` and `stopAllSounds()` to execute), you need some indication of where the event ends. In Flash, the event wrapper starts with an opening curly bracket ({) and ends with a closing curly bracket (}). Sort of like a "script sandwich" with two curly brackets for bread and scripts in the middle. Even if that corny analogy doesn't help, you should see that scripts are wrapped inside events. (See Figure 4.2.)

Nintendo Exhibit 1029
Page 00116

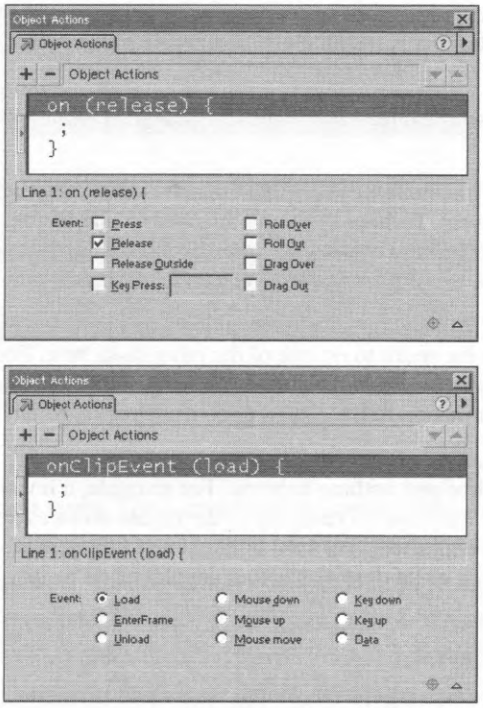


Figure 4.1 *The mouse events (for buttons) on the top and the clip events (for clips) on the bottom include all the events available in Flash.*

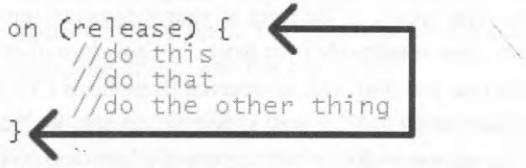


Figure 4.2 *Everything between the two curly brackets will be performed when the event occurs.*

Results of Events

After an event happens, a script is executed. The instructions that Flash follows can range from simple to complex. Let's start by thinking about the results of events in very general terms. First, we'll look at the five types of tasks that happen as the result of an event:

Nintendo Exhibit 1029
Page 00117

- One: A script can do something that is invisible or unimportant to the user (such as add to a counter that is tracking how many times a button is clicked).
- Two: The script can do something visual (such as change the rotation of a movie clip).
- Three: The script can trigger another script to execute—effectively behaving like a homemade event. Perhaps the script attached to one button causes another script to rotate a movie clip. In this way, several buttons can trigger the same "rotate the clip" script and your code can be modularized.
- Four: Scripts can "ask" other scripts for information (the part that's said to be "returned") and use the result to do one of the other tasks here. For example, one script could ascertain the current exchange rate from another script and use that information to translate a price from yen to dollars.
- Finally, the fifth result of an event is nothing. That might seem funny, but events happen all the time and nothing happens. For example, a linear movie with no buttons will have no response to the mouse down event (despite the fact that it happened). It's kind of like a tree falling in a deserted forest. That's an event despite whether anyone reacts to it.

Terms

We can't go much further without defining a few terms.

Syntax

Syntax is the way any sentence is organized. In programming, a "sentence of code" is called a *statement*. The syntax, or the form of your statements, must adhere to strict syntax rules. One simple rule (that works in English, too) is that any parenthetical statement that you start with an opening parenthesis ("(") must be followed by a closing parenthesis (")"). If your statement's syntax isn't correct, your code will perform unexpectedly (at best) or not at all (at worst). There are many ways that Flash helps you follow perfect syntax.

While in Normal Mode, the Actions panel will all but ensure that your syntax is perfect. Actually, as soon as your statement contains an error, it is highlighted in bright red. An explanatory message appears to guide you along. Figure 4.3 shows a statement that isn't complete. Remember that even after your syntax is perfected, it doesn't mean your Flash movie will play the way you want it to. The script could still contain logical errors. The first step is to get the syntax fixed, and then you can ensure that it performs correctly.

Nintendo Exhibit 1029
Page 00118

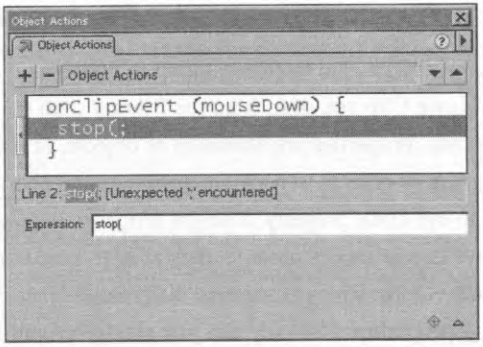


Figure 4.3 *While editing your script, errors are highlighted while typing.*

Another way that Flash helps you produce perfect syntax is by color-coding certain terms. The Colored Syntax option in the Action panel's Options menu is a great way to learn to type (and spell) terms correctly. Keywords and identifiers (which, for now, simply consider built-in terms of ActionScript) are colored blue. Properties (such as `_x` and `_y` or `_currentFrame`) are always green (not highlighted green the way deprecated ActionScript is, but the text itself is green). Comments (which are ignored by Flash to provide you a way to write notes to yourself or others reading your code) are always pink. Finally, anything you type between quotation marks is gray. This is particularly useful because you must remember to end any quotations you start (just like parenthetical statements). If you start a quotation, everything that follows will turn gray until you type another quotation mark. You'll find out more about keywords, properties, comments, and strings later this chapter.

I wanted to mention the Colored Syntax feature because it's a great way to learn. For example, when you look through someone else's finished code, it's easy to separate the elements that are built-in to Flash (they'll be blue) from the code that's created by the programmer (left in black). Additionally, if you want to type a particular property but are unsure of your spelling, you can tell you've typed it correctly when it turns green. Possibly the most frustrating syntax rule in Flash is that all keywords are case sensitive. For example, `goToAndPlay(2)` won't work (only `gotoAndPlay(2)` will work). Throughout this book you'll see many places where syntax is important. Colored Syntax is just one little helper available to you.

Nintendo Exhibit 1029
Page 00119

Objects, Instances, Instance Names, and Properties

In Flash, there are many types of objects, but the easiest kind to understand are Movie Clip instances. Every Movie Clip you drag from the Library onto the Stage becomes a separate instance. Properties are attributes of objects. In real life, you could compare properties of objects to personal attributes and characteristics of people. People have an eye color property, a height property, a weight property, and so on. If you have two instances onstage, they're both based on the original master symbol but each is most likely positioned differently—that is, they'll have different _x and _y properties. Not only can you change properties of individual clip instances while editing, but through scripting you can change each instance's properties at runtime. This is easiest if the clip instance has been given an instance name (via the Instance panel). In pseudo-code, you'd say something like this: *Set the _x property of the instance named "box_1."* Just remember that it's the instance name that matters (not the master name in the Library).

After you fully understand how to access clip instance properties, it will be easy to understand other objects. In Chapter 7, "The Movie Clip Object," you'll learn how to access instance properties. You will use the same techniques in Chapters 9, "Selecting Text, Trapping Keys, and Manipulating Strings," 10, "Arrays," and 11, "Objects," to control the sound object, string object, and color object (among others). Finally, in Chapter 12, "Homemade Objects," you'll create your own objects. The reason the Movie Clip is so much easier to understand is that it's the only object that you can see and touch. All the other objects are just as real, but you'll never grab one with your mouse or see it onscreen. It's sort of like how you can have a picture in your mind. Think of a physical printed photograph as the Movie Clip object—you can fold it, frame it, hang it on the wall. Now, consider a picture in your mind as any of the other objects. Even though there's no physical evidence, you can still manipulate the picture in your mind—picture it being folded, framed, and so on.

Statements, Expressions, and Operators

The entire block of text in the Actions panel is a script, where each line is a statement. As mentioned earlier, statements are "sentences of code." It's not so critical if you interchange the terms *script* and *statement*. But if a statement of code is like an entire sentence in speech, an expression is like a phrase. In speech, you might use the phrase *slow as molasses*. First of all, you'd create an entire sentence (a statement) such as *His computer is slow as molasses*. You can see that the expression (or phrase) is only part of the statement.

Nintendo Exhibit 1029
Page 00120

Expressions in Flash are evaluated. For example, an entire statement might read: `halfprice=price/2`. The part on the right of the equal sign is an expression and it will be evaluated (that is, a result will be placed in that spot). If it helps, think of evaluated expressions as having "math" done on them. Now, imagine that "slow as molasses" is an actual speed (say 1 mile-per-hour). If you evaluated the expression, you'd be left with the result "1 MPH." The original sentence would become *His computer is 1 MPH*. This analogy is beginning to fall apart, but it's a good way to learn about statements and expressions (plus the terms *evaluate* and *result*).

Finally, operators perform specific (usually math) operations. They always need at least one number, called an *operand*, on which to work. For instance, "+" is an operator that performs addition. The "+" addition operator requires two operands to work. Get it? There's the operator (doing the work) and the operands (getting work done on them). The thing to remember is that operators are used within expressions. Therefore, they're evaluated and their results appear in their place. For example, saying `whatever=2+2` is the same as saying `whatever=4`. The expression on the right is evaluated and turns into 4. This might seem simple. Just think of expressions morphing into their results.

Operators are pretty easy to understand when you consider familiar math operators (such as multiplication "*", division "/", subtraction "-", for example). Others aren't so easy to figure out and might even seem arbitrary. For example, one deceptively useful operator uses the percent sign ("%"), but not to calculate a percent. It's called the *modulo* operator and results in the remainder of two numbers. `10%2` is evaluated into `0`. Ten divided by 2 has 0 for a remainder. This can be useful to determine whether a number is odd or even (or evenly divisible by any other number, for that matter)—that is, if the number % 2 is 0, it's an even number because it's evenly divisible by 2. Anyway, I don't recommend memorizing all the operators. We'll use many of them in the Workshop section especially when we make sliders and the currency exchange calculator. However, after you understand the form—that operators create an expression that's evaluated and they always require at least one operand—you can learn them gradually.

Actions and Functions

In Flash, "Actions" have many meanings. I think it's best to think of Actions as Flash-unique commands that execute specific tasks. For example, the `stop()` Action is a command that causes the playback head to stop. The `gotoAndPlay("framelabel")` Action is another command that jumps to a frame

Nintendo Exhibit 1029
Page 00121

number or label and plays. These Actions do things specific to Flash. However, the Actions panel lists many "Actions" that are, technically, statements (or elements of statements). Figure 4.4 shows some Actions that are really statements. For example, `if` is a piece of code that creates a statement. You have to specify additional details (such as, "if *what*?"), but `if` was around way before Flash—and in my book, it's a statement. For the sake of understanding, I'll try to avoid the term "Actions" completely.

Keep in mind that although the term "Actions" is made up, Actions behave in a consistent manner. All the Actions in Flash are one of two types. They're either specific Flash commands (such as `stop`, `duplicateMovieClip`, and `play`), or they are statements (such as `if` or `break`). (Actually, you'll learn in the next chapter that a more common name for many actions is *method*.) Many Actions (of either type) need additional parameters provided—for instance, `gotoAndPlay()` requires a frame label or number. Another consistent attribute of all Actions is that they're never evaluated; they do things. Actions have consequences (and to avoid confusion, don't call them results). The script you create using Actions doesn't morph into a result the way expressions do. Actions simply "do things." Either they help construct a statement or they do a Flash-unique task.

Flash's built-in functions are easy to confuse with Actions, but they are different in a profound way. A function always returns a value. That is, a built-in function's job is to calculate an answer. Similar to how an expression is evaluated, a function "returns" a value which, in turn, is placed where the function was called. For example, the function `getTimer()` returns the total number of milliseconds elapsed since the movie started. It's almost like a "what time is it?" function. But consider if you asked someone, "What time is it?" That person might look at her watch, but unless she "returns" the answer, you might never know what time it is. Similarly, if you just wrote the script `getTimer()`, it would be meaningless. The `getTimer()` would turn into a number—maybe 10200, but so what? If, however, you wrote `elapsedTime=getTimer()/1000 + " seconds"`, the `getTimer()` part would turn into 10200 and the expression on the right side of the equation would evaluate as "10.2 seconds."

All the built-in functions behave this way. Consider these real-life activities as Actions: drive a car, stop the car, fill the car with gas. Now consider these as functions: get today's price of gas, determine how many liters fit in my 20-gallon gas tank, determine how many yen are equal to 10 dollars. The functions always result in an answer (which can be used within a bigger calculation). Notice, too, that some functions just provide an answer (such as "getTimer" or "get today's

Nintendo Exhibit 1029
Page 00122

price of gas"). However, other functions (such as "convert gallons to liters") require additional data to be provided (that is, "How many gallons?"). In these cases, you need to provide a parameter.



Figure 4.4 *All the "Actions" in Flash are actually a mix of statements and Flash-unique commands.*

Nintendo Exhibit 1029
Page 00123

In Chapter 8, "Functions," you'll learn all about functions—both the built-in functions and homemade functions. For now, just understand Flash Actions are either Flash-unique commands (also called *methods*) or statements, and that built-in functions always return values. The big difference is that an Action can be used by itself and a function is always used within a larger script.

Special Characters and Formatting

As part of the many syntax rules in Flash, certain characters have special meaning. Additionally, the layout or format of your scripts must follow some rules. Parentheses are a perfect example of this point. Any parenthetical statement must start with an opening parenthesis and must always end with a closing parenthesis. The reason this is important is that you can have nested parentheses like this: `((6+2)*3)/2`, which results in 12. That's different than `6+((2*3)/2)`, which is 9. The innermost expressions are evaluated first, followed by expressions in outer parentheses. The point is that anything you start, you have to finish. For example, `((6+2)*3/2` will cause an error because there's an extra open parenthesis.

There are identical rules for quotation marks (`"`), brackets (`[` and `]`), and curly brackets (`{` and `}`). Quotation marks are a little different because apostrophes will work the same as quotation marks. This way you can nest quotations in quotations. You can't mix and match, however. For example, all of these are legitimate:

```
"This is 'cool' "
'Flash version "5" is better than "4"'
"Phillip's dog is named Max."
```

But these won't work:

```
"What you "start" you have to "finish""
"Mixing will not work'
```

It's interesting because in the case of `"Phillip's dog..."`, you might think that the apostrophe would cause problems. It works fine, however. Also, nested quotation marks are different than nested parentheses. You can have two open parentheses in a row, and as long as you close what you open, they will work fine. However, two quotation marks in a row will cause Flash to think that the second one is closing what the first one opened. So, if you have to nest quotations in quotations, you must use quotations for one and apostrophes for the other. (By the way, the curly, or "smart," quotation marks " and " won't work in the Actions panel.)

Nintendo Exhibit 1029
Page 00124

Generally, spaces and return characters (at the end of a line) are ignored by Flash. If you start a parenthetical statement and close it five lines later, there's no problem. Because you can't simply create a new line to represent a new thought, every line of code can be terminated with a semicolon. For example, look at this piece of code:

```
if (age>18){
    voter=1;
    draftable=1;
}
```

Even though we haven't looked at the if-statement yet, you can see that the open curly bracket is followed (three lines later) by a closing curly bracket. Also, there are two separate lines of code, each ending with a semicolon. Actually, the following is totally legitimate:

```
if (age>18){voter=1;draftable=1;}
```

The first example was easier to read. In either case, the semicolon clearly ended individual statements. Occasionally you can be sloppy and, based on context, Flash will figure out what you intended, but it's best to deliberately use semicolons to end each line.

Finally, one of the best character sequences to learn is the double forward slash `//`, which creates a comment. Comments are ignored by Flash, so you can write anything you want. You can use comments as notes to yourself (or anyone else who might be viewing your code). The phrase "commenting your code" means that you include all the information necessary to make sense of your code through comments. Also, comments are useful to cause Flash to temporarily ignore part of your code—effectively removing the code, but leaving it so that you can restore it easily.

Everything that follows `//` is ignored. You'll see the commented text turn pink, which makes it very easy to recognize. The following code sample includes comments:

```
//This button makes the movie stop
on (release) {
    stop(); //don't go any further
}
```

Normally, comments automatically end at the end of a line. (Notice that there's no "closing" comment mark in the code.) However, you can use `/*` to start a block of comments that will continue until a `*/` is reached.

Nintendo Exhibit 1029
Page 00125

```
/* Comments:
    This is a comment.
    This is another comment.
*/
```

We'll encounter other special formatting and new terms throughout this book, but this should give you a good start. Now, with the terms and other technical matters out of the way, we can move on to variables and how to use them.

Data Types and Variables

It's difficult if not impossible to discuss either data types or variables without mentioning the other. We'll discuss both here.

Variables' Names and Values

Variables are a way that you can safely—yet temporarily—store data. Think of variables as individual whiteboards. You can write someone's phone number on one whiteboard and refer to it later. You can easily change the phone number on the whiteboard. Variables are similar in that you can store information in a variable for later reference or change it any time. You can have as many variables as you choose. To return to our whiteboard analogy, you could have one for phone number and one for address. Every variable has two parts: a *name*, so that you can keep track of which variable is which, and a *value*, or what is stored in the variable. For example, the name of our phone number whiteboard (or variable) could be "phonenumber," but the value might be "800-555-1212." To assign the value of `phonenumber`, you could write the script `phonenumber="800-555-1212"` (which, translated, reads "the variable named phonenumber is assigned the value "800-555-1212"). We'll discuss variables at length later in this chapter, but you really just need the concepts of value and name first.

String and Number Data Types

The type of data that is stored in a variable's value is important. The type of data that goes into a variable can be one of many types. Just as you can store paper in an envelope, you can also store paper clips, money, even sand. This concept is easiest to understand when you compare two common data types: string and number.

Nintendo Exhibit 1029
Page 00126

A string is always expressed between quotation marks. "Phillip", "David", and "Kerman" are all strings. (Remember that we're talking about the values contained in a variable—not the variable's name.) You can do interesting maneuvers with strings, such as converting them to all uppercase letters and determining the number of letters in a string (using the techniques you'll learn in Chapter 9.

Numbers (such as 13 and 35) are a different data type. They are as difficult to compare to strings as apples are to oranges. You can also do interesting things with numbers, such as add them together and find their difference. However, you can't mix them. "Phillip" plus 35 doesn't make sense.

The good news about Flash is that it's *untyped*—meaning that you don't have to decide ahead of time the data type for each variable's value. From context, Flash figures it out. If you said `username="Phillip"`, Flash would treat the value of `username` like a string because "Phillip" is contained within quotation marks. If you said `age=35`, Flash would treat `age` as a number. (No, Flash doesn't know what a username or age is—it just figures out the data type from how you've used the word.) Because ActionScript is untyped, you can change the data type in a variable. Maybe in the first frame you said `score="untested"` (at which point the value in the `score` variable would be a string:`"untested"`). Later, you could say `score=85` (and `score`'s value would become a number). Despite the freedom to change data types, it's still important to understand each data type.

More About Strings

In Chapter 9, you'll learn all about manipulating strings. There are two tricks you should understand now. Concatenating two strings is simply a matter of using the *concatenate* character +. For example, `"Phillip"+"Kerman"` would evaluate as `"PhillipKerman"`. A more practical situation might be that you have an input text field onscreen (with the variable name "username"), and after the user types her name, you could use the script `message="Welcome " + username + "!"` and make sure to display a dynamic field containing the variable `message`. Notice the variable `message` is having its value set to a string that combines the word "Welcome " (with an extra space) plus the value of `username` plus an exclamation point. Figure 4.5 shows part of such a "Hello World" exercise.

Nintendo Exhibit 1029
Page 00127



Figure 4.5 *Dynamic text (based on user input) can appear by modifying strings.*

Another interesting trick to understand is how to include characters that would otherwise be difficult or impossible to include inside a string. For example, what if you want your string to contain a double quotation mark, such as `"Phillip is "old""`? Forgetting for a moment the way you learned to nest quotation marks (inside apostrophes), the difficulty here is that the quotation mark right before the letter "o" would act like an end-quote for the quotation mark at the beginning. Instead of letting Flash get confused, you can use a backslash in front of any quotation mark you want to be used verbatim. So, `"Phillip is \"old\""` works fine. This is called an *escape sequence* and on page 43, of Macromedia's Flash 5 ActionScript Reference Guide, you'll find a table of escape sequences. For example, `\r` creates a return and `\t` makes a tab.

More About Numbers

Manipulating numbers will probably look more familiar to you because it works the same as traditional math. Something simple like `10-2` evaluates to 8. It might seem strange that `10+2` evaluates to 12 because we just learned the plus sign is the concatenation character for strings. The plus sign acts as an addition operator when the two operands are numbers. But, something odd like `"Phillip"+2` evaluates to "Phillip2", because one of the operands is a string, so the plus sign concatenates the two. Often, depending on context, Flash will use the same symbols differently.

Other number operators are pretty easy to figure out. Common operators such as / (divide) and * (multiply) are all listed in the operators section of the Action

Nintendo Exhibit 1029
Page 00128

{Chapter 4 Workshop }

Creating a Horizontal Slider

This workshop explores a popular user interface control: the slider. Many users prefer to specify a number by using the seemingly continuous scale offered by a slider rather than by typing into a field. Some tactile learners may respond better to the sense of touch provided by a slider. Another benefit of using sliders is that sliders can be adjusted without taking your hand off the mouse.

This workshop has two parts. First we make a perfectly acceptable plain slider and then we convert it into a more usable Smart Clip.

Creating a Quick and Dirty Slider

1. Create a button that will serve as the slider the user is supposed to grab. Perhaps you'll want to use a vertical rectangle with rounded corners like the one you made in Workshop 2, "Faking Video." If the shape is particularly small, consider creating a large "hit" area for the button so that it's not too difficult for the user to grab the slider.
2. To make the slider movable, it must be a Movie Clip. Select the button instance you just made and convert it to a Movie Clip symbol called "Simple Slider." Now you have a button inside a clip.
3. Double-click the Simple Slider to edit its contents. Create a Dynamic Text field that reads `100` using `_sans` font. Associate the variable `percent` with this block of text. Make sure "Selectable" is not checked and that "Border/BG" is checked. Position the text right above the button as shown in Figure W4.1.

Nintendo Exhibit 1029
Page 00129

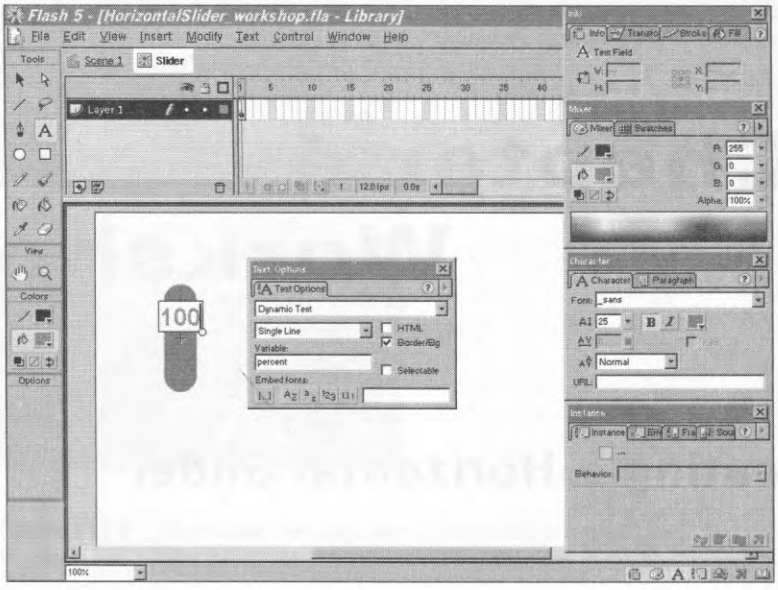


Figure W4.1 *The Dynamic Text field displays the slider's location as a percentage.*

4. On the instance of your button, place the following script:

```
on (press) {
  drag=1;
}
on (release, releaseOutside) {
  drag=0;
}
```

This script makes the button set the variable `drag` to `1` or `0`. The bulk of the code will be placed elsewhere; it's just that it's easiest to respond to mouse events using buttons. The only result of this script is that `drag` will change value, but that will trigger another script to kick in.

5. Return to the main timeline and select the instance of Simple Slider. Place the following script on the clip instance:

```
onClipEvent (mouseMove) {
  if (drag==1){
     _x=_root._xmouse;
  }
}
```

Translated, this script says that any time the user's mouse moves and the `drag` variable happens to be `1`, set the `_x` property of the clip to the location of the mouse (`_xmouse` within the `_root` timeline). Because we're setting

Nintendo Exhibit 1029
Page 00130

the _x property of the clip in the _root timeline, we want it to match the user's mouse position in the _root timeline.

Test the movie. It works! In the next steps, we're going to fix a few things: The movie should move smoother, there should be minimum and maximum bounds, it shouldn't jump when you click the slider's left or right edge the way it currently snaps itself to the center of the mouse pointer, and the percent has to display in the Dynamic Text field. These are all easy tasks!

6. Throw in an updateAfterEvent() right before the close (}) of the if statement to make the movie play smoother.

7. To add bounds, use the Info panel to determine the minimum and maximum locations for the slider in the main timeline. Draw some vertical lines in the main timeline to represent the bounds. After you have the x coordinate values, place the following script above the mouseMove event on your slider (actually, the following code works fine if you place it below the event, but it makes more sense to have the load event come first):

```
onClipEvent(load){
  min=1;     //or your preferred minimum
  max=550;   //or your maximum
}
```

In this code, replace the values for min and max with whatever the Info panel told you were your minimum and maximum values.

8. Inside the if statement of your mouseMove event and right above updateAfterEvent(), place this script:

```
if (_x>max){
  _x=max;
}
if (_x<min){
  _x=min;
}
```

These two if statements make sure that the _x property doesn't go below min or above max. If you test the movie now, you should see that the slider is smooth and doesn't go past the bounds specified.

9. Now that we know the bounds involved, we can calculate percentage. The formula for percentage is simply "current position / total distance." Total distance is max-min, and current position is the slider's _x property. If your minimum is greater than 0 (suppose that the leftmost position is 220), current position is really _x-min. If the _x is right at minimum, the current position should be 0, which it will be if you simply subtract min from the _x property. Here's the formula you can place above the updateAfterEvent() script:

```
percent = (_x-min)/(max-min)*100;
```

Nintendo Exhibit 1029
Page 00131

Actually, to find just the integer portion of percent, you can use the Math Object's `floor()` method, which was discussed in Chapter 5, "Programming Structures." Replace the preceding code with this version:

```
percent = Math.floor((_x-min)/(max-min)*100);
```

The two remaining issues to resolve are the fact the slider snaps to the center of the mouse and that the percentage doesn't start displaying until the slider is moved.

10. The snapping is caused by the fact that we're positioning the slider's `_x` to the exact location of the `_xmouse`. If the user clicks the left side of the slider, we really want the slider to move to the `_xmouse` plus how far off center the user clicked. If the user clicks the right side, we want to subtract the offset. It's a subtle issue, but one that comes up often, especially when you make a homemade draggable Movie Clip: You can't just move the clip to the mouse's location; you'll have to subtract an "offset." To address the snapping, first add this code to the press mouse event script on the button that you made in step 4, right above `drag=1`:

```
offsetX=_x-_root._xmouse;
```

11. In the `mouseMove` event on the clip, change the line `_x=_root._xmouse` to read as follows:

```
_x=_root._xmouse+offsetX;
```

When the user clicks, we calculate the difference between `_x` and `_root._xmouse` and then we use that value by adding it to the clip's `_x` position.

//

You may ask, "If we always *add* the offset, won't that make the slider appear to the right?" If the offset value is positive then, yes. Otherwise, when the offset is negative, adding it makes the slider appear to the left. Let's try some what-if numbers. Suppose that the slider is at `50`. If you start clicking at `45`, the slider should move to wherever the mouse is *plus* `5`. If you click at `55`, the slider should move with the mouse *minus* `5`. Actually, you can always add the offset because in the first case `50-45` is `5`. The second case `50-55` is `-5`. I always get messed up deciding whether to subtract `_xmouse` from `_x` or to subtract `_x` from `_xmouse`. Try a few what-if numbers, or use the trial-and-error method to work out the logic pretty quickly.

12. We have another step to complete before we can convert this slider to a Smart Clip and can really start using it. Place this code at the end of the `load` event on the slider instance, right before the closing curly bracket (}):

```
percent = Math.floor((_x-min)/(max-min)*100);
```

Nintendo Exhibit 1029
Page 00132

Although this statement is redundant (it also appears in the `mouseMove` event), it ensures that the percent is displayed at the very beginning.

13. We're about to convert this clip to a Smart Clip. In case you want to get off here and start using the clip as-is, realize that you can use the `percent` value anywhere in your movie. For example, if you have another clip on the Stage with an instance name `box`, you can add either or both of these lines of code right above `updateAfterEvent()` inside the slider clip's `mouseMove` event:

```
_root.box._alpha=percent;
_root.box._rotation=(percent/100)*360;
```

You can add this code right now because you have a perfectly suitable slider. Study Figure W4.2 for a full view of everything up to this point.

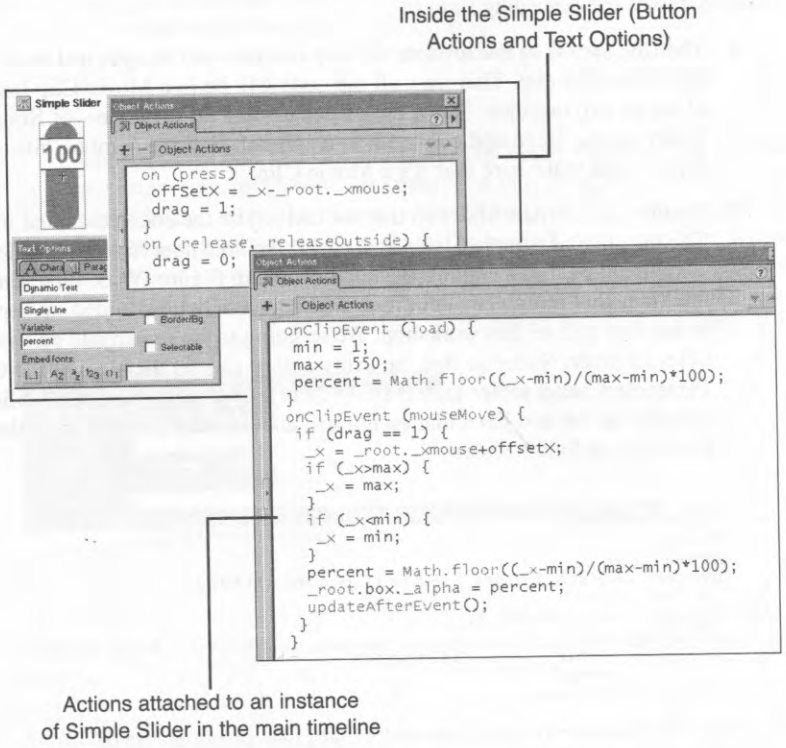


Figure W4.2 *The Simple Slider contains ActionScript both in the master symbol and attached to the instance.*

Nintendo Exhibit 1029
Page 00133

Converting the Slider into a Smart Clip

We're about to make our slider slightly better. The limiting issues with the slider we built in the first part of this workshop include the following:

- All the code is attached to the clip instance on the Stage instead of inside the clip itself. The problem is that we can't drag additional instances of this clip from the Library without copying and pasting the code. If there were a bug in the code, it would be duplicated in each instance and would have to be fixed in all individual instances.
- The `min` and `max` variables are hard wired. Our Smart Clip will allow each instance its own `min` and `max` values, plus an additional variable for the starting percentage.

Here are the steps for making our pretty good slider into a very good Smart Clip that addresses the preceding issues:

1. The first step is to encapsulate the clip instance and its attached code inside another clip. This way, all the code will be *in* a Movie Clip instead of *on* an clip instance. Select the instance—and all the code—of Simple Slider on the Stage and convert it to a symbol. Call the symbol "Smart Slider" and make sure that it's a Movie Clip.
2. Double-click Smart Slider so that we can adjust the code contained in it. The instance of Simple Slider now inside Smart Slider has some code that we can adjust. Let's explore the code shown in Figure W4.3. First remove any lines that reference `_root.box` which you may have added in step 13 in the first part of this workshop. We're going to make `min` and `max` variables of Smart Slider so that the using author can set them using the Clip Parameters panel rather than making them variables of the Simple Slider. Simply cut the two lines that assign `min` and `max` and paste them in the first keyframe of Smart Slider.

```
onClipEvent (load) {
    min=1;
    max=550;
    percent = Math.floor((_x-min)/(max-min)*100);
}
onClipEvent (mouseMove) {
    if (drag == 1) {
        _x = _root._xmouse+offsetX;
        if (_x>max) {
            _x = max;
        }
        if (_x<min) {
            _x = min;
        }
        percent = Math.floor((_x-min)/(max-min)*100);
        updateAfterEvent();
    }
}
```

Figure W4.3 *Here is the script that needs to be improved when it's converted to a Smart Clip.*

Nintendo Exhibit 1029
Page 00134

3. Notice that the two lines that assign `percent` are identical. Remove the line starting `percent=...` inside the `load` event. Later, we'll write a script that takes care of assigning percent in the first keyframe of Smart Slider. At this point, you can remove the now empty `load` event. If you study the `mouseMove` event, you'll notice several variables that have to be adjusted, or "re-scoped," to target the correct timeline, most notably every reference to `min` and `max` needs to be up one level in the Smart Slider and not part of the Simple Slider where they are now. Instead of fixing everything here by adding `_parent` in front of each reference, select all the text between `if (drag==1){` and `updateAfterEvent()` (Figure W4.4) and cut it (do not delete it). Replace the cut code with `_parent.moveSlider()`. We'll write this function in a minute. The modified script on the slider instance should look like this:

```
onClipEvent (mouseMove) {
  if (drag==1){
    _parent.moveSlider();
    updateAfterEvent();
  }
}
```

The variables `min` and `max` will be part of Smart Slider so that they can be set using the Clip Parameters panel. The only thing that "happens" in this script is that, when the mouse is moved and `drag` is `1`, the yet-to-be-written `moveSlider()` function is invoked. The `moveSlider()` function will reside in the first keyframe of Smart Slider.

```
onClipEvent (mouseMove) {
    if (drag == 1) {
        _x = _root._xmouse+offsetX;
        if (_x>max) {
            _x = max;
        }
        if (_x<min) {
            _x = min;
        }
        percent = Math.floor((_x-min)/(max-min)*100);
        updateAfterEvent();
    }
}
```

Figure W4.4 *Here is the portion of the script to be cut and moved to a keyframe function.*

4. To create the `moveSlider()` function, enter the following script underneath the two lines that assign `min` and `max` in the first keyframe of Smart Slider:

```
function moveSlider(){
}
```

Nintendo Exhibit 1029
Page 00135

Between the two curly brackets, paste everything from your clipboard to form the following complete script:

```
function moveSlider(){
  _x = _root._xmouse+offSetX;
  if (_x>max) {
    _x = max;
  }
  if (_x<min) {
    _x = min;
  }
  percent = Math.floor((_x-min)/(max-min)*100);
}
```

At this point, the slider should still work—sort of. We simply moved the bulk of code to a function. Luckily, all the variables here don't need any adjustment the way `min` had to change to `_parent.min` on the old instance of Slider in the first part of this workshop. However, our `offSetX` has stopped working properly. Actually, we never assigned an `offSetX` of the new Smart Slider clip.

5. You have to adjust the script attached to the button inside the Slider clip. The variable `drag` is fine because that is used by the Slider clip itself. However, `offSetX` must be part of the Smart Slider. But simply changing the assignment to read `_parent.offSetX=_x-_root._xmouse` won't work entirely. (Try it out if you want.) It's not the difference between the slider's `_x` and the main timeline's `_xmouse`, but rather the difference between `_x` and the `_xmouse` of the Smart Slider's timeline. This line of code (found inside the `press` mouse event attached to the button instance inside the Slider symbol) has to read as follows:

   ```
   _parent.offSetX=_x-_parent._xmouse;
   ```

   I suppose we could have seen this coming. In the first part of this workshop, we had the Slider clip that either used its own variables or targeted the `_root` timeline. But back then `_root` was only "one level up." Now, with the instance of Slider *inside* the Smart Slider clip, the hierarchy isn't much different but `_root` no longer means "one level up," and we should use `_parent` instead (Figure W4.5).

6. One last fix and we'll be ready to make this clip a true Smart Clip! The percent isn't displaying correctly; it's being calculated fine, but it's part of Smart Slider now. You can fix this by accessing the Dynamic Text field inside the Slider symbol. Change the percent variable (with which the text is currently associated) to read: `_parent.percent` (Figure W4.6).

Nintendo Exhibit 1029
Page 00136

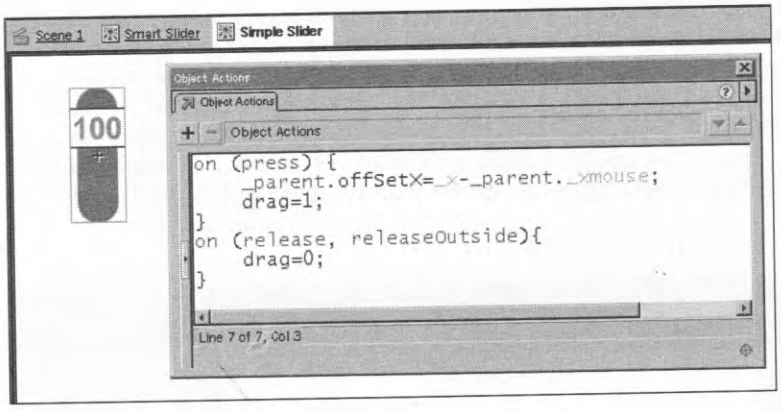


Figure W4.5 *The adjusted script on the button.*

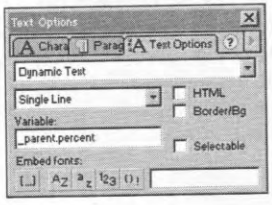


Figure W4.6 *One solution for displaying the correct* `percent` *is with a relative reference.*

7. We should make one more refinement before we convert to a Smart Clip. It would be nice to let the using author specify an initial value for `percent`. We should make sure that the slider not only displays the initial percent, but positions itself on the Stage accordingly. In the first keyframe of Smart Slider, type the following two lines of code directly after `min` and `max` are assigned:

```
percent=50;
_x= min+((max-min)*(percent/100));
```

   For now, `percent` is hard wired, as are `min` and `max`, for that matter. The second line sets Smart Slider's `_x` to `min` plus the `percent` multiplied by the range (`max-min`). We don't have to worry about making this value an integer using `Math.floor()` because there's no problem setting a clip's `_x` property to a float (a number that has decimal values).

8. After you test this out a few times to make sure that it works, we can turn Smart Slider into a Smart Clip so that the user can set `min`, `max`, and *initial* `percent`. Comment out the first three lines of code in frame 1 of Smart Slider to make the script appear like this:

Nintendo Exhibit 1029
Page 00137

```
//min = 1;
//max = 550;
//percent=50;
_x= min+((max-min)*(percent/100));
function moveSlider(){
  _x = _root._xmouse+offSetX;
  if (_x>max) {
    _x = max;
  }
  if (_x<min) {
    _x = min;
  }
  percent = Math.floor((_x-min)/(max-min)*100);
}
```

Commenting out these lines prevents us from overriding the values the using author supplies through the Clip Parameters panel.

9. Open the Library, select Smart Slider, and access the Define Clip Parameters dialog. Create three "parameters": `min`, `max`, and `percent`. Note that you can set initial values for each of these parameters as well (Figure W4.7).

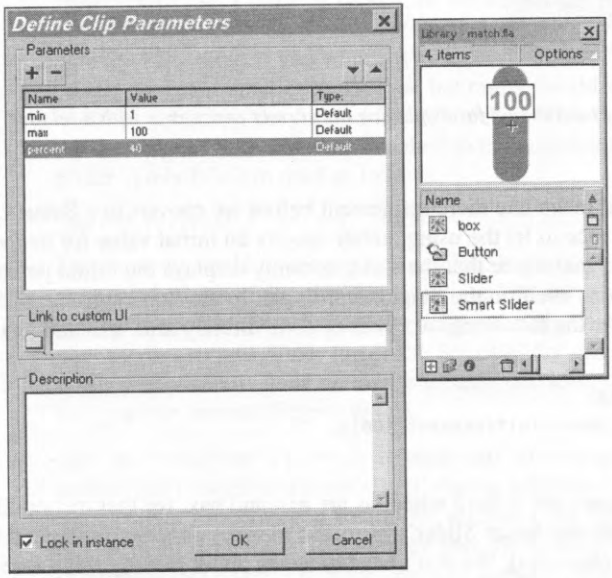


Figure W4.7 *Use the Define Clip Parameters dialog for the master symbol to enable the author to set these variables individually per instance.*

Nintendo Exhibit 1029
Page 00138

10. An overview of the main scripts is shown in Figure W4.8. The Smart Clip is complete. However, it lacks one tiny feature that could make the difference between something that's just cool looking and something that's very useful. We need a feature that links the slider to something onscreen. Although we can always ascertain the value of `percent` for any instance of this Smart Clip, it would be nicer if the slider itself sent a message to the Stage instead of us having to write a script that keeps asking the slider the value of its `percent`.

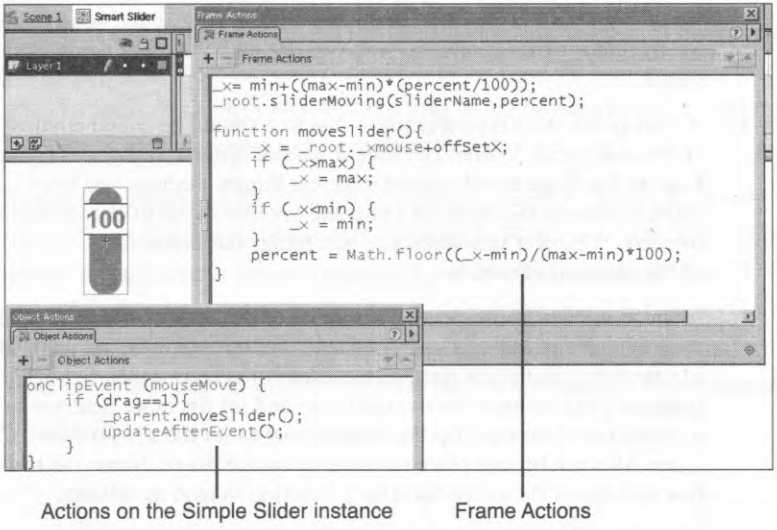


Figure W4.8 *The scripts for the first keyframe inside Smart Slider (top right) and the instance of Simple Slider (bottom left).*

To make this work, we'll need a function in the first keyframe of our main movie. No matter where we use the Smart Clip, it will send a message to this function in the `_root`. In the first keyframe of your main timeline, type this function:

```
function sliderMoving (whichSlider, howMuch){
}
```

The idea is that `sliderMoving()` can be invoked by any one slider, and there can be several. Passed as parameters will be a unique name `whichSlider` (that identifies which slider) and `howMuch` (the current percentage). Currently, this function doesn't do anything. If you want, you can add a line between the curly brackets that reads as follows:

```
trace("slider is moving "+whichSlider+" and "+howMuch);
```

For now, this statement tells us when a slider is invoking this function. We'll add more to this function later.

Nintendo Exhibit 1029
Page 00139

11. Making Smart Slider call this function is easy. Right before the close of the `mouseMove` event, type this line of code:

    ```
    _root.sliderMoving(sliderName,percent);
    ```

    This statement simply calls the `sliderMoving()` function in the main timeline. It sends as parameters the values of `sliderName` and `percent`. We haven't assigned `sliderName` yet, and we're not going to; the using author will do that using the Clip Parameters panel.

12. To let the using author assign sliderName, we just need to add that parameter through the Define Clip Parameters dialog. Add sliderName to the Smart Slider's parameters the way you did with the other parameters in step 9.

13. At this point, the `sliderMoving()` function should be called anytime the slider is dragged. It would be nice if several different instances of Smart Clip on the Stage could control different things. Perhaps you want one slider to change the alpha of a clip and another slider to change that clip's rotation. Whatever you want, just sort things out inside the `sliderMoving()` function.

    In the main timeline, delete all instances of the Smart Slider and start fresh by dragging two instances on the Stage. Set the `min`, `max`, `percent`, and `sliderName` parameters through the Clip Parameters panel. Set one instance's `sliderName` to `rotateSlider` and set the other instance to `alphaSlider` (use the Clip Parmeters panel to set the `sliderName`). Create a new Movie Clip and place an instance on the Stage. Name the instance box and adjust the `sliderMoving()` function to read as follows:

    ```
    function sliderMoving (whichSlider, howMuch) {
      if (whichSlider=="alphaSlider"){
        box._alpha=howMuch;
      }
      if (whichSlider=="rotateSlider"){
        box._rotation=(howMuch/100)*360;
      }
    }
    ```

    If you want to add more sliders, come up with a unique `sliderName` for each one and add another if-statement in the `sliderMoving()` function in order that uses the `percent` received as the value of `howMuch`.

14. We have one more step to make this Smart Clip complete. To make the initial percentage for any slider invoke the `sliderMoving()` function, even before the user even starts dragging, add the following script to the first keyframe in Smart Slider:

    ```
    _root.sliderMoving(sliderName,percent);
    ```

    Without this statement, my `alphaSlider` example could be given an initial percent of, say, 30, but the box instance wouldn't change its `_alpha` until I started dragging the slider.

Nintendo Exhibit 1029
Page 00140

Summary

In this workshop, you explored how to write a relatively simple script and then convert it into a Smart Clip. The process involved first creating a hard-wired script that worked to our satisfaction. Then we identified the aspects of the script that were not ideal—especially the fact that all the code was on the surface of the clip instance and that there were several hard-wired numbers. We also tried to think of generic applications for this Smart Clip, such as allowing the author to specify an initial percentage. All the variables that the using author would eventually access through the Clip Parameters panel must be "in" the main timeline of the clip itself. We simply moved the scripts to the first keyframe or used relative references such as `_parent` to make this possible.

For more information about relative references, refer to Chapter 1, "Flash Basics," and Chapter 4, "Basic Programming in Flash." General information about how to identify ways to improve scripts can be found in Chapter 3, "The Programmer's Approach." Writing functions was discussed in Chapter 8, "Functions." Finally, Chapter 5, "Programming Structures," covered if-statements, and how to write expressions such as our formula that determines percentage.

Nintendo Exhibit 1029
Page 00141

{Chapter 5 Workshop }

Building a Slide Show

After the last couple of workshops, this one will seem like a breeze. That doesn't mean that the results won't be impressive. In this workshop, you'll see how a little code can create a really nice effect. We're going to build a slide show that lets the user step through a series of images in a Movie Clip. Although the user can step forward and back in the slide show, instead of simply jumping to the next frame, we'll make the current page fade out and the next one fade up.

In addition to creating a slide show, you'll learn two general techniques: First, we'll use the `enterFrame` event to continuously make changes to the `_alpha` property of the current slide. We'll also look at a way to deactivate buttons so that when users view the first slide, they can't click the "page back" button. Similarly, when users are on the last page, they shouldn't be able to click "page forward." Although these two techniques are useful for *this* workshop, they'll also prove helpful in many of the workshops to come.

1. Create a Movie Clip with a separate image on each frame. Inside a new Movie Clip, insert blank keyframes as you import pictures or just draw something onto each frame. Consider creating a separate layer in which you draw a rectangle to serve as an outline (Figure W5.1).

Nintendo Exhibit 1029
Page 00143

within clips, 26
xmlObj, 321
var_tempVar, 178
verifying versions of Flash, 359
version control (documentation), 120-121
$version variable, 364
video
converting to bitmaps, 367
creating, 367-374
displaying, 339
sliders, 368-373
virtual key codes, 202-204
volume
adjusting, 241-243
determining, 241

W-Z

while loops, 105-109
Window menu, Common Libraries, Smart Clips command, 290, 431
windows, output windows
copying data, 525
errors, 117
with( ), 575
with Action, 33
Word Float script, repairing, 565-566
writing
expressions, 74
functions, 181-182
that accept parameters, 170-173
check( ), 378-379
homemade, 166-167
that return values, 173-175
selectPointer( ), 386-387
JavaScript inter-movie communications, 551-556, 558
scripts, 26
specifications, 38
statements, 87
conditional, 98-99
loop, 101-102
to text files, 338
x coordinate, 18
XML
data
format, 318-320
importing, 318
loading, 320-323, 531-534
modifying, 326
structured, 529
declarations, 320
nodes, 319
attributes, 325-326
<CATALOG>, 321
pseudo-code, 323-324
structure, 319-320
structuring, 530-531
XML Objects, 29
XML Socket objects, 326-327
XML-structured data versus URL-encoded data, 580
xmlObj variable, 321
Xtras, 338
y coordinate, 18
Yellow Box script, repairing, 566-567
zero-based systems, 189

Nintendo Exhibit 1029
Page 00144

{ ActionScripting in FLASH™ }

Building a linear animation or moderately interactive Flash movie is relatively simple. With the upgrade from Flash 4 to Flash 5, ActionScripting has become an even more crucial and powerful part of the product. Tapping the vast programming capabilities of Flash, however, requires more than brute programming skill—it requires a keen understanding of how to apply scripting to specific tasks. This book concentrates on practical applications for Flash programming.

Structured in two parts, this book begins with the Foundation section, which clarifies those fundamentals that a novice programmer may not be familiar with, including general programming theory. Then the book progresses through more complex Flash ActionScripting applications, including functions, arrays, objects, Smart Clips, and interfacing with external data.

Following the Foundation, the Workshop section includes 21 practical applications of ActionScripting, ranging from beginner to advanced levels. This organization allows the reader to apply what he has learned to any future challenges he might encounter.

Intermediate Flash users just embarking on scripting will learn the first steps necessary to recognize situations that call for scripting and make their hard-wired sites dynamic and extensible. Experienced programmers will learn how to apply their knowledge to specific tasks in Flash.

Phillip Kerman is an independent programmer, teacher, and writer specializing in Macromedia products. His degree in Imaging and Photographic Technology from the Rochester Institute of Technology was earned back when "multimedia" involved programming multiple slide projector presentations with dissolves synchronized to a sound track. In 1993 he found Macromedia Authorware a natural fit to his interest and skills. Since then, he's transitioned his expertise from Authorware to Director, and now, to Flash. During the last seven years he's had to adapt to a total of 13 version upgrades of these—Flash 5 being the most significant of them all! Phillip is also the best-selling author of *Sams Teach Yourself Flash 5 in 24 Hours*.

7 52063 32078 5

ISBN 0-672-32078-9

53999

9 780672 320781

$39.99 USA
$59.95 CAN
£28.99 Net UK

Category: Graphics/Multimedia/ Web Development/Internet
Covers: Macromedia® Flash™
User Level: Intermediate—Advanced

SAMS
www.samspublishing.com

InformIT
Find IT Solutions at
www.informit.com

Nintendo Exhibit 1029
Page 00145

APPENDIX 1014B

LIBRARY | CATALOG

1 of 1

BOOK

ActionScripting in Flash

Full Record MARC Tags

Personal name

Kerman, Phillip.

Main title

ActionScripting in Flash / Phillip Kerman.

Published/Created

Indianapolis, IN : SAMS, 2001.

Request this Item LC Find It

More Information

LCCN Permalink	https://lccn.loc.gov/00108995
Description	xi, 618 p. : ill. ; 24 cm.
ISBN	0672320789
LC classification	TR897.7 .K468 2001
LC Subjects	Flash (Computer file) Computer animation. ActionScript (Computer program language)

Web sites--Design.

Browse by shelf order	TR897.7
Notes	Includes index.
LCCN	00108995
Dewey class no.	006.6/96
Type of material	Book

Item Availability

CALL NUMBER	TR897.7 .K468 2001 FT MEADE Copy 2
Request in	Jefferson or Adams Building Reading Rooms - STORED OFFSITE
Status	c.2 Charged - Due - (Internal Loan)

LIBRARY | CATALOG

1 of 1

BOOK

ActionScripting in Flash

Full Record MARC Tags

000		01030cam a22002894a 4500
001		12174240
005		20170210085554.0
008		000915s2001 inua 001 0 eng
906	__	|**a** 7 |**b** cbc |**c** orignew |**d** 2 |**e** epcn |**f** 20 |**g** y-gencatlg
925	0_	|**a** acquire |**b** 1 shelf copy |**x** policy default
925	1_	|**a** acquire |**b** 2 shelf copies |**x** policy default
955	__	|**a** pc20 09-15-00; |**a** pv04 2001-08-09 to ASCD |**a** jf00 2001-08-14; |**c** jf03 2002-03-28 to Subj.; |**d** jf04 2002-04-10 to S.L. |**e** jf25 2002-04-11 2 copies to Dewey |**a** aa05 2002-05-03
010	__	|**a** 00108995
020	__	|**a** 0672320789
040	__	|**a** DLC |**c** DLC |**d** DLC
042	__	|**a** pcc
050	00	|**a** TR897.7 |**b** .K468 2001
082	00	|**a** 006.6/96 |**2** 21
100	1_	|**a** Kerman, Phillip.
245	10	|**a** ActionScripting in Flash / |**c** Phillip Kerman.
260	__	|**a** Indianapolis, IN : |**b** SAMS, |**c** 2001.
300	__	|**a** xi, 618 p. : |**b** ill. ; |**c** 24 cm.
500	__	|**a** Includes index.
650	_0	|**a** Computer animation.
630	00	|**a** Flash (Computer file)
650	_0	|**a** ActionScript (Computer program language)
650	_0	|**a** Web sites |**x** Design.

Request this Item | LC Find It

Item Availability

CALL NUMBER	TR897.7 .K468 2001 FT MEADE Copy 2
Request in	Jefferson or Adams Building Reading Rooms - STORED OFFSITE
Status	c.2 Charged - Due - (Internal Loan)

Nintendo Exhibit 1029
Page 00150

APPENDIX 1014C

libraries.org

A directory of libraries throughout the world

found **3** items where the item contains **Library of Congress**. Showing item **2** of **3**.

Library of Congress

United States -- Library of Congress

Address: 101 Independence Avenue SE
Washington, **District of Columbia**
20540
United States

Phone: 202-707-5000

Connect to: **Library Web Site**✔ **Online Catalog**

Library details: **Library of Congress** is a **National** library.
This library is affiliated with **Library of Congress** (**view map**) . The collection of the library contains **173,000,000** volumes. The library circulates **154,000** items per year.

Library Hours		
Day	**Open**	**Close**
Sunday	Closed	
Monday	8:30 am	8:30 pm
Tuesday	8:30 am	8:30 pm
Wednesday	8:30 am	8:30 pm
Thursday	8:30 am	8:30 pm
Friday	8:30 am	5:00 pm
Saturday	8:30 am	5:00 pm

Other Info: Went live with Voyager on August 15, 1999.

Permalink: **https://librarytechnology.org/library/2263**
(Use this link to refer back to this listing.)

Administration: The director of the library is **Dr. Carla Hayden** .

Organizational structure: The library is part of a **publicly-funded** organization. It operates on a **Non-profit** financial model.

Affiliations: The library is a member of the **Association of Research Libraries**

Collection size	173,000,000	**volumes**
Annual Circulation	154,000	**transactions**

Wireless: The library offers **wireless** access to the Internet. http://www.loc.gov/rr/wirelessaccess.html

Technology Profile		
	Product Name	**Year Contracted**
Current Automation System	**Voyager**	**1998**
Discovery Service (w/index)	**Ex Libris CDI**	
OpenURL Link resolver	SFX	2006
Electronic Resource Management	Electronic Resource Management	



Library of Congress



map location✔

Related Libraries

- All members of the **Association of Research Libraries** members
- **Directory of ARL Member Libraries**
- Libraries located in **Washington** (District of Columbia)
- **all National libs in District of Columbia**

About libraries.org

libraries.org is a free directory of libraries throughout the world.

This directory is a component of **Library Technology Guides**.

Please **contact Marshall Breeding** to report corrections about information listed for this library.

Follow Marshall on .

Item ID Type	Barcode	

This library is responsible for the procurement of the library automation system.

View Privacy and Security Report

Identifiers	
libraries.org ID	2263
OCLC Symbol	DLC
WorldCat Registry ID	45266

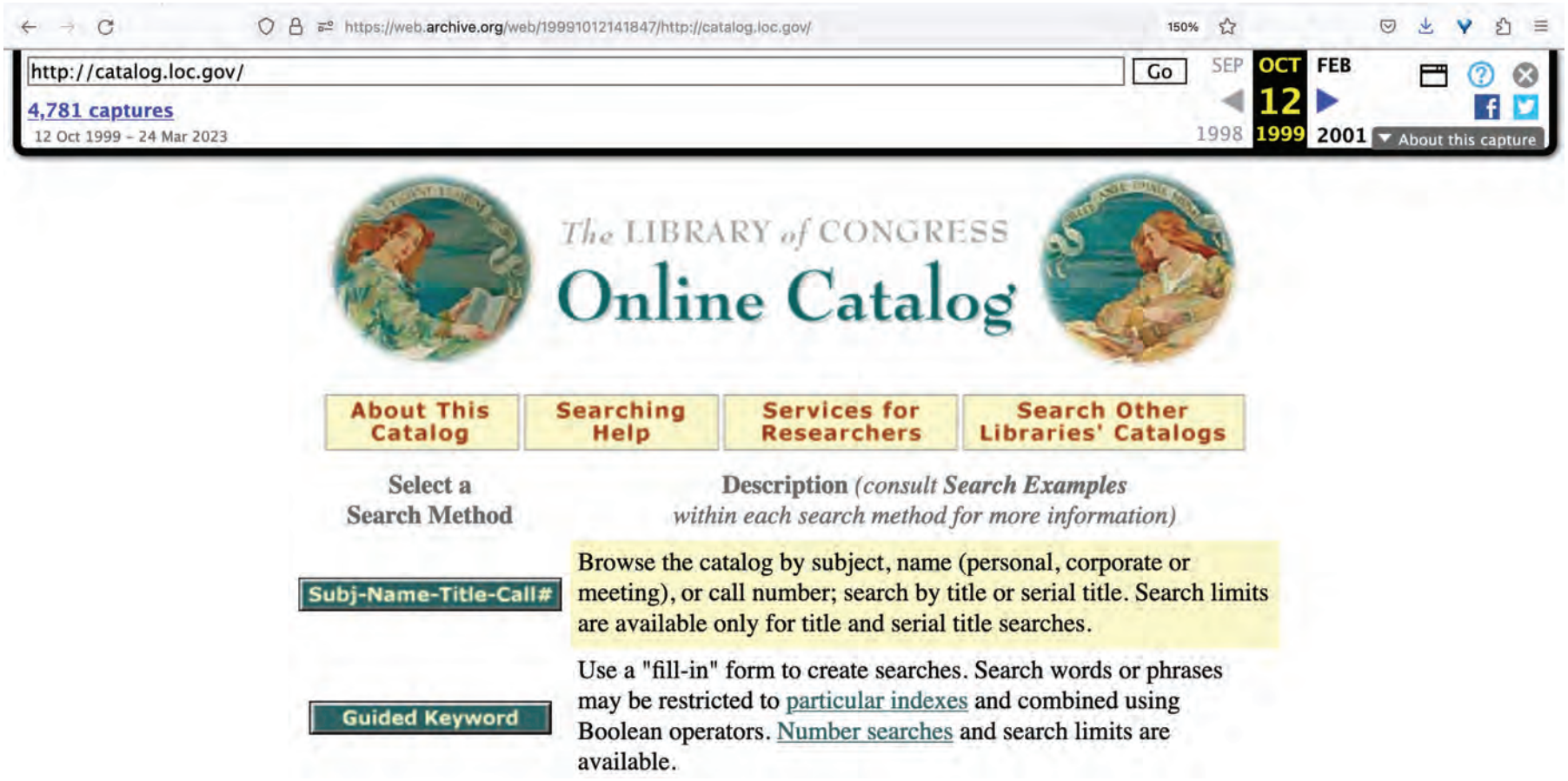
Record History: This listing was created on Aug 9, 1998 and was last modified on Nov 5, 2022.

Updates: Corrections or Updates? Registered members of Library Technology Guides can submit updates to library listings in libraries.org. **Registration** is free and easy. Already registered? **login**. Or, you can report corrections just by sending a message to **Marshall Breeding**.

libraries.org | Update this Entry | Library Technology Guides

Maintained by **Marshall Breeding**

terms of service
Copyright 1994-2022


https://web.archive.org/web/19991012141847/http://catalog.loc.gov/
150%
http://catalog.loc.gov/
Go
4,781 captures
12 Oct 1999 - 24 Mar 2023
SEP
OCT
12
FEB
1998
1999
2001
About this capture
The LIBRARY of CONGRESS
Online Catalog
About This Catalog
Searching Help
Services for Researchers
Search Other Libraries' Catalogs
Select a Search Method
Description (consult Search Examples within each search method for more information)
Subj-Name-Title-Call#
Browse the catalog by subject, name (personal, corporate or meeting), or call number; search by title or serial title. Search limits are available only for title and serial title searches.
Guided Keyword
Use a "fill-in" form to create searches. Search words or phrases may be restricted to particular indexes and combined using Boolean operators. Number searches and search limits are available.

The Wayback Machine - https://web.archive.org/web/19991012141847/http://catalog.loc.gov:80/



The LIBRARY of CONGRESS
Online Catalog



About This Catalog | Searching Help | Services for Researchers | Search Other Libraries' Catalogs

Select a Search Method	Description *(consult **Search Examples** within each search method for more information)*
Subj-Name-Title-Call#	Browse the catalog by subject, name (personal, corporate or meeting), or call number; search by title or serial title. Search limits are available only for title and serial title searches.
Guided Keyword	Use a "fill-in" form to create searches. Search words or phrases may be restricted to particular indexes and combined using Boolean operators. Number searches and search limits are available.
Command Keyword	Use Boolean operators and/or index codes in a command mode. Number searches and search limits are available.
Keyword	Search for words or exact phrases anywhere in the catalog record. Add punctuation for more precise results (see search examples). Results are relevance ranked. Search limits are available.

Set Search Limits	Limits restrict search results by language, date, type of material (e.g., book, music, etc.), location of material in the Library, or place of publication. Set search limits now, or prior to executing a search; they remain in effect until cleared or changed.



Library of Congress

Comments: lcweb@loc.gov (10/06/99)

APPENDIX 1014D

Copyright United States Copyright Office

Try the Copyright Public Records System (CPRS) pilot with enhanced search features and filters.

Help | Search | History | Titles | Start Over

Public Catalog

Copyright Catalog (1978 to present)
Search Request: Author = Kerman, Philip
Search Results: Displaying 2 of 9 entries

previous next

Labeled View

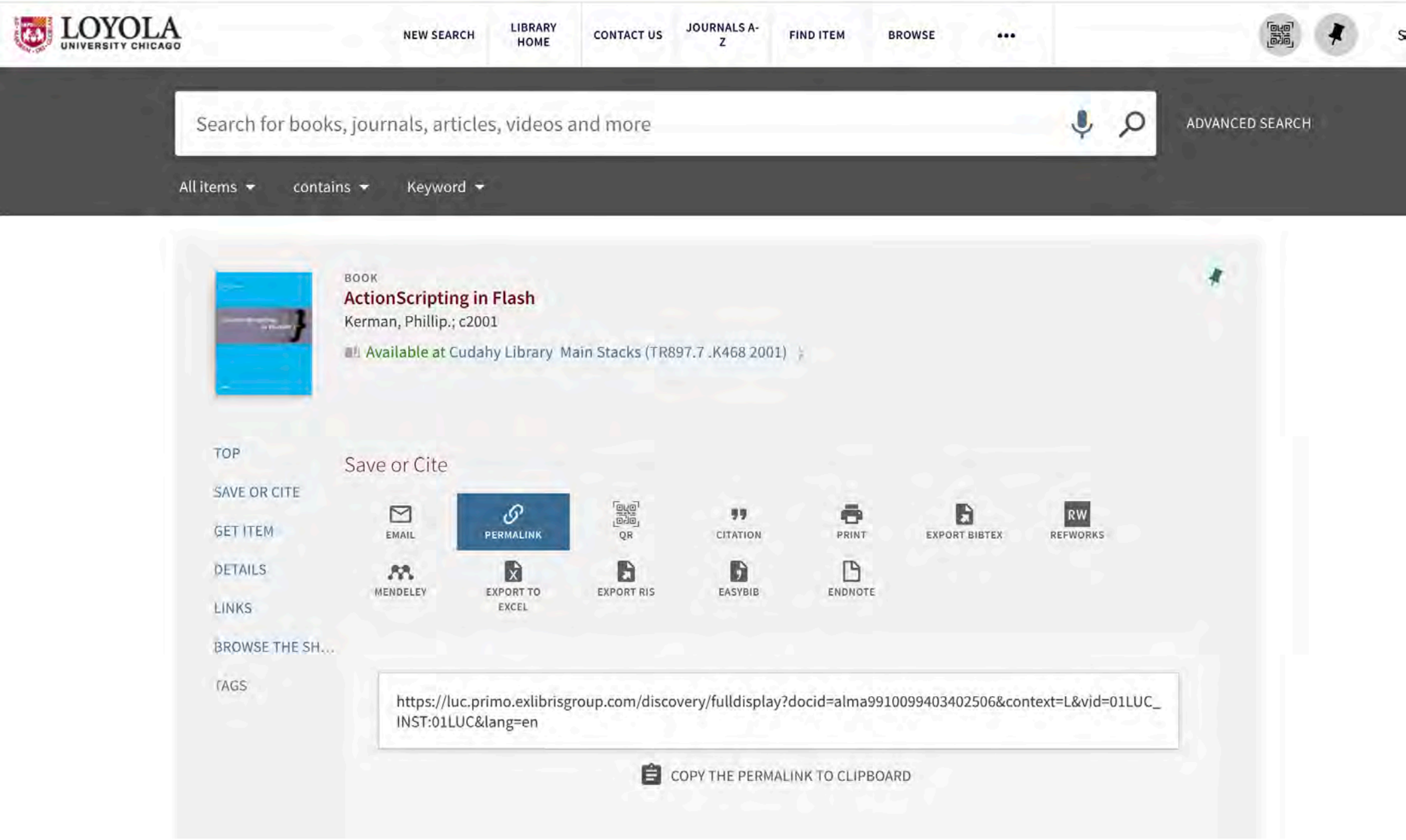
ActionScripting in FLASH / Phillip Kerman.

Type of Work: Text
Registration Number / Date: TX0005377798 / 2001-09-04
Application Title: Developing Flash action scripts.
Title: ActionScripting in FLASH / Phillip Kerman.
Imprint: Indianapolis : Sams, c2001.
Description: 618 p.
Copyright Claimant: Phillip Kerman
Date of Creation: 2001
Date of Publication: 2001-04-09
Names: Kerman, Phillip

previous next

Save, Print and Email (Help Page)	
Select Download Format	Full Record ▾ Format for Print/Save
Enter your email address:	Email

APPENDIX 1014E


LOYOLA UNIVERSITY CHICAGO
NEW SEARCH
LIBRARY HOME
CONTACT US
JOURNALS A-Z
FIND ITEM
BROWSE
Search for books, journals, articles, videos and more
ADVANCED SEARCH
All items
contains
Keyword
BOOK
ActionScripting in Flash
Kerman, Phillip.; c2001
Available at Cudahy Library Main Stacks (TR897.7 .K468 2001)
TOP
SAVE OR CITE
GET ITEM
DETAILS
LINKS
BROWSE THE SH...
TAGS
Save or Cite
EMAIL
PERMALINK
QR
CITATION
PRINT
EXPORT BIBTEX
REFWORKS
MENDELEY
EXPORT TO EXCEL
EXPORT RIS
EASYBIB
ENDNOTE
https://luc.primo.exlibrisgroup.com/discovery/fulldisplay?docid=alma9910099403402506&context=L&vid=01LUC_INST:01LUC&lang=en
COPY THE PERMALINK TO CLIPBOARD

LOCATION ITEMS

Cudahy Library
Available , Main Stacks ; TR897.7 .K468 2001
(1 copy, 1 available, 0 requests)

Item in place
Loanable

Details

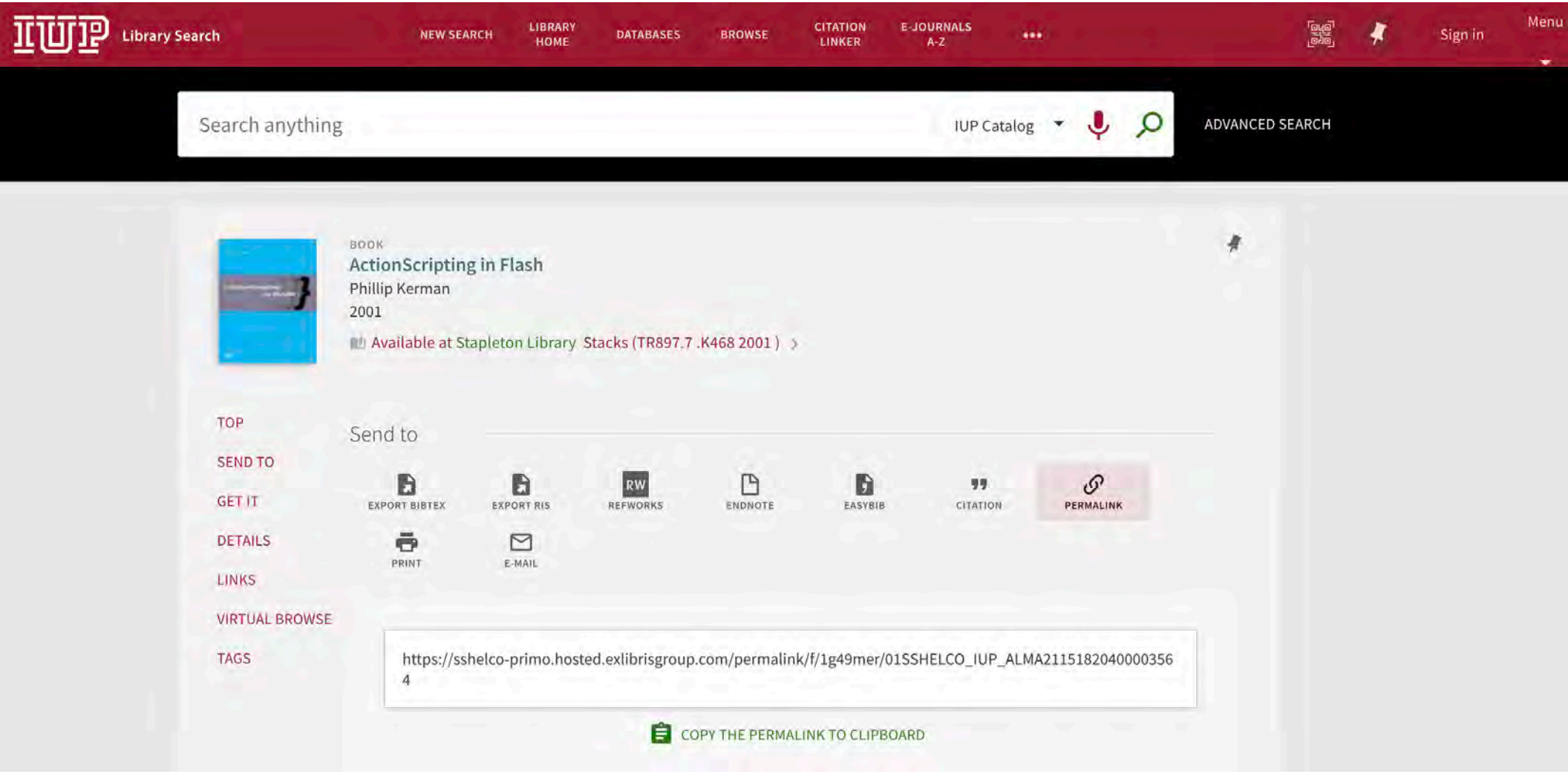
Title	ActionScripting in Flash
Author	Kerman, Phillip.
Subjects	Computer animation Flash (Computer file) ActionScript (Computer program language) Web sites -- Design Macromedia Flash (Computer file)
Publisher	Indianapolis, IN : SAMS
Creation Date	c2001
Format	xi, 618 p. : ill. ; 24 cm.
Language	English
Identifier	LC : 00108995 OCLC : (OCoLC)ocm47038202 ISBN : 0672320789
Source	Library Catalog
ALMA MMS ID	9910099403402506

Links

Report a Problem
Ask a Librarian
Get help with this item
Display Source Record

```
leader  00876cam a22003134a 4500
001     9910099403402506
005     20020430121230.0
008     000915s2001    inua           001 0 eng
010     ##$a   00108995
035     ##$a(OCoLC)ocm47038202
035     ##$a(ICL)1009940-lucdb-Voyager
035     ##$z(OCoLC)46393439
040     ##$aDLC $cDLC $dDPL $dUKM $dTXA
015     ##$aGBA1-Y9741
019     ##$a46393439
020     ##$a0672320789
042     ##$apcc
050     00$aTR897.7 $b.K468 2001
092     0#$a006.6/96 $221
049     ##$aIALA
100     1#$aKerman, Phillip.
245     10$aActionScripting in Flash / $cPhillip Kerman.
260     ##$aIndianapolis, IN : $bSAMS, $cc2001.
300     ##$axi, 618 p. : $bill. ; $c24 cm.
500     ##$aIncludes index.
650     #0$aComputer animation.
630     00$aFlash (Computer file)
650     #0$aActionScript (Computer program language)
650     #0$aWeb sites $xDesign.
630     04$aMacromedia Flash (Computer file)
994     ##$aE0 $bIAL
999     ##$aLakeside Libraries
```

APPENDIX 1014F


IUP Library Search
NEW SEARCH
LIBRARY HOME
DATABASES
BROWSE
CITATION LINKER
E-JOURNALS A-Z
Sign in
Menu
Search anything
IUP Catalog
ADVANCED SEARCH
BOOK
ActionScripting in Flash
Phillip Kerman
2001
Available at Stapleton Library Stacks (TR897.7 .K468 2001)
TOP
SEND TO
GET IT
DETAILS
LINKS
VIRTUAL BROWSE
TAGS
Send to
EXPORT BIBTEX
EXPORT RIS
REFWORKS
ENDNOTE
EASYBIB
CITATION
PERMALINK
PRINT
E-MAIL
https://sshelco-primo.hosted.exlibrisgroup.com/permalink/f/1g49mer/01SSHELCO_IUP_ALMA21151820400003564
COPY THE PERMALINK TO CLIPBOARD

Stapleton Library Stacks TR897.7 .K468 2001
Hide Details

(1 copy, 1 available, 0 requests)

1 - 1 of 1 Records

BARCODE	TYPE	POLICY	DESCRIPTION	STATUS
000007270710	Book	Loanable		Item in place

Details

Title	ActionScripting in Flash
Author	Phillip Kerman >
Subjects	Computer animation >
	ActionScript (Computer program language) >
	Web sites -- Design >
	Flash (Computer file) >
	Macromedia Flash (Computer file) >
Publisher	Indianapolis, IN : SAMS
Creation Date	2001
Format	xi, 618 p. : ill. ; 24 cm..
Language	English
Identifier	ISBN : 0672320789
Source	01SSHELCO_IUP_ALMA
Alma MMS ID	996474955003564
Call Number	TR897.7 .K468 2001

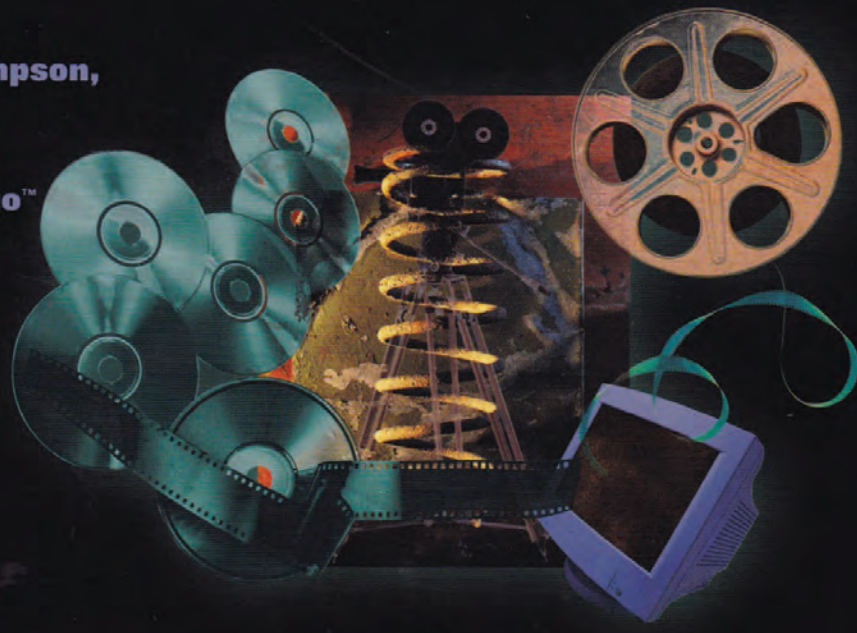
Links

This item in Amazon.com >
This item in WorldCat® >
Display Source Record >

```
Leader 00932cam a22003374a 4500
001    996474955003564
005    20020811200917.0
008    000915s2001    inua          001 0 eng
010    ##$a   00108995
015    ##$aGBA1-Y9741
019    ##$a46393439
020    ##$a0672320789
035    ##$a(OCoLC)ocm47038202
035    ##$a647495
035    ##$a(PInU)647495-iupdb
035    ##$z(OCoLC)46393439
040    ##$aDLC$cDLC$dDPL$dUKM$dTXA
042    ##$apcc
049    ##$aPZII
050    00$aTR897.7$b.K468 2001
082    00$a006.6/96$221
090    ##$aTR897.7$b.K468 2001
100    1#$aKerman, Phillip.
245    10$aActionScripting in Flash /$cPhillip Kerman.
260    ##$aIndianapolis, IN :$bSAMS,$c2001.
300    ##$axi, 618 p. :$bill. ;$c24 cm.
500    ##$aIncludes index.
630    00$aFlash (Computer file)
630    04$aMacromedia Flash (Computer file)
650    #0$aComputer animation.
650    #0$aActionScript (Computer program language)
650    #0$aWeb sites$xDesign.
994    ##$aE0$bPZI
AVA    ##$0996474955003564$822151820390003564$a01SSHELCO_IUP$bSTAPLETON$cStacks$dTR897.7 .K468 2001$eavailable$f1$g0$jPZII$k0$p1$qStapleton Library
CAT    ##$aArts_Architecture_Applied_Arts$bVisual_Arts$cPhotography
INST   ##$a01SSHELCO_IUP
INT    ##$aP
```

APPENDIX 1015A

Foreword by
John "JT" Thompson,
Macromedia
Chief Scientist,
Inventor of Lingo™



Special Edition

Using

Macromedia® Director® 8

CD-ROM included

Gary Rosenzweig

que


Special Edition
Using
Macromedia
Director 8
Rosenzweig
que

Special Edition

A Must-Have Reference for Macromedia® Director® 8 Users!

- Learn Director 8 from the ground up, beginning with chapters on how to use Director to create animation and presentations
- Explore the different media types used by Director—including bitmaps, text, fonts, sounds, digital video, vector graphics, and Flash cast members
- Use Lingo™, the programming language inside Director, to create interactive animation, complex presentations, applications, and games
- Take advantage of advanced Lingo to control graphics, sound, video, vector shapes, the Director environment, Web browsers, and data
- Extend the capabilities of Director with behaviors, MIAWs, Score recording, object-oriented programming, clip art, and Xtras
- Maximize performance, develop cross-platform applications, and use debugging techniques
- Discover the Shockwave® Multiuser Server and how you can use it to build multiuser applications
- Learn about all the latest features in Director 8, including Lingo image manipulation, publishing, scalable Shockwave, and new sound commands

Gary Rosenzweig is a multimedia software developer, entrepreneur, and writer. His business, CleverMedia, does consulting and Internet publishing and creates games, applications, and utilities using Director. CleverMedia's Web site contains the world's largest collection of Shockwave applets made by one company. Gary has a B.S. in computer science from Drexel University and an M.A. in journalism from the University of North Carolina at Chapel Hill.

"This book is cram-packed with demos and Lingo examples that will have you coding like a pro in no time."
—Pat McClellan
Managing Editor,
Director Online
www.director-online.com

"*Special Edition Using Macromedia Director 8* describes elements that can be used again and again, providing time savings and reducing testing time."
—Matthew Manuel
Senior Software Engineer,
Creative Capers
Entertainment



CD-ROM includes

- More than 160 development files used in the book
- Two completely unprotected applications with source code
- Four Xtras created by the author to help you work in Director
- Macromedia Director 8 Shockwave Studio Trial Edition
- Ari Feldman's SpriteLib 1.0 graphics library with more than 700 professional-quality images for use in your projects!
- Lots of other freeware and shareware that can be used in or with Director

$39.99 USA / $59.95 CAN / £28.99 Net UK (inc of VAT)

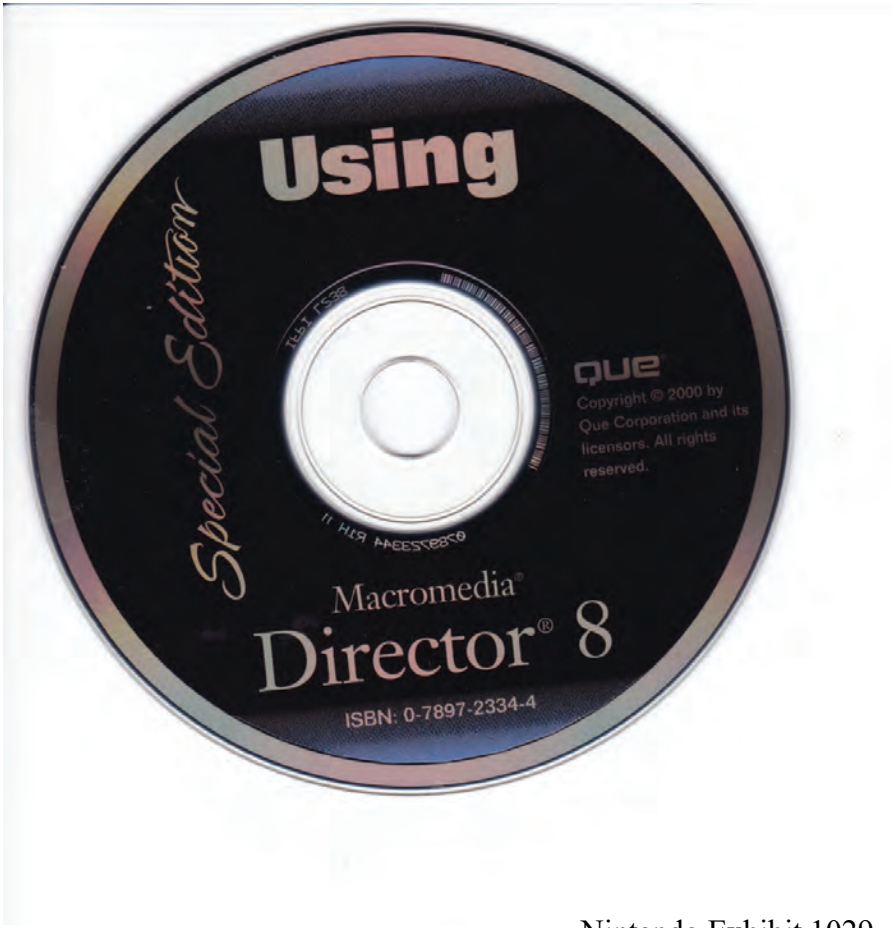
Category	Multimedia
Covers	Macromedia Director 8
User Level	Intermediate—Advanced





que®

www.quepublishing.com


Using
Special Edition
QUE
Copyright © 2000 by Que Corporation and its licensors. All rights reserved.
Macromedia®
Director® 8
ISBN: 0-7897-2334-4

Guide to What's New in Macromedia Director 8

- **The Property Inspector**—A universal palette window for viewing and changing sprite and member properties. See "The Property Inspector," p. 40.
- **Publishing**—You can use Director's new Publish function to preset how your Shockwave movies are created. See "Publish Settings," p. 173.
- **Zoomable Stage**—You now have the capability to view the Stage as greater than or less than 100%. See "The Stage," p. 28.
- **Scalable Shockwave**—Shockwave movies can now stretch to fit the browser window. See "The Shockwave Tab," p. 174.
- **Cast Window List View**—You now have the capability to view Cast members as a list and sort them in different ways. See "Cast Window Properties and Settings," p. 24.
- **Guides**—You can add movable guides to the Stage to help align sprites. See "Grids and Guides," p. 49.
- **Asset Management Fields**—Members now include a creation and modified time, as well as a created by and modified by timestamp. You can also add custom comments to each member. See "Commenting Your Code," p. 650.
- **Image Lingo**—New Lingo commands enable you to detect and edit pixels in bitmap members. See "Manipulating Bitmap Members," p. 396.
- **Sound Lingo**—A new set of sound Lingo enables you to queue sounds, loop sounds, and create a variety of other effects. See "New Sound Commands," p. 375.
- **Linked Scripts**—You can now have script cast members that use an external file. These scripts can be shared between movies and edited by external editors. See "Understanding Script Types," p. 220.
- **Behavior Typing**—You can now create behaviors that insist on a certain sprite type or situation in order to be used. See "Controlling Where Behaviors Are Used," p. 289.
- **Timeouts**—A new Lingo object enables you to trigger events at specific time intervals. See "Timeout Objects," p. 439.

For a more detailed list of what is new in Director 8, see Chapter 00, "What's New in Director 8," starting on page 7, just before Chapter 1.

Macromedia Director 8 Lingo Reference Card

Ink Numbers

0	Copy
1	Transparent
2	Reverse
3	Ghost
4	Not Copy
5	Not Transparent
6	Not Reverse
7	Not Ghost
8	Matte
9	Mask
32	Blend
33	Add Pin
34	Add
35	Subtract Pin
36	Background Transparent
37	Lightest
38	Subtract
39	Darkest
40	Lighten
41	Darken

Cursor Numbers

-1	(Reset Cursor)
0	Arrow
1	I-Beam
2	Crosshair
3	Crossbar
4	Watch/Hourglass
200	Blank
254	Help
280	Finger
260	Hand
290	Closed Hand
291	No Drop Hand
292	Copy Closed Hand
256	Pencil
257	Eraser
258	Select
259	Bucket
272	Lasso
281	Dropper
301	Air Brush
302	Zoom In
303	Zoom Out
284	Vertical Size
285	Horizontal Size
286	Diagonal Size

Behavior Property Description Formats

#integer
#float
#string
#symbol
#member
#bitmap
#filmloop
#field
#palette
#picture
#sound
#button
#shape
#movie
#digitalvideo
#script
#richtext
#ole
#transition
#xtra
#frame
#marker
#ink
#boolean

Member Types

#animgif
#bitmap
#button
#cursor
#digitalVideo
#empty
#field
#filmLoop
#flash
#font
#movie
#ole
#palette
#picture
#QuickTimeMedia
#script
#shape
#sound
#swa
#text
#transition
#vectorShape

Transition Types

1	Wipe right	19	Reveal down	37	Venetian blinds
2	Wipe left	20	Reveal down, left	38	Checkerboard
3	Wipe down	21	Reveal left	39	Strips on bottom, build left
4	Wipe up	22	Reveal up, left	40	Strips on bottom, build right
5	Center out, horizontal	23	Dissolve, pixels fast*	41	Strips on left, build down
6	Edges in, horizontal	24	Dissolve, boxy rects	42	Strips on left, build up
7	Center out, vertical	25	Dissolve, boxy squares	43	Strips on right, build down
8	Edges in, vertical	26	Dissolve, patterns	44	Strips on right, build up
9	Center out, square	27	Random rows	45	Strips on top, build left
10	Edges in, square	28	Random columns	46	Strips on top, build right
11	Push left	29	Cover down	47	Zoom open
12	Push right	30	Cover down, left	48	Zoom close
13	Push down	31	Cover down, right	49	Vertical blinds
14	Push up	32	Cover left	50	Dissolve, bits fast*
15	Reveal up	33	Cover right	51	Dissolve, pixels*
16	Reveal up, right	34	Cover up	52	Dissolve, bits*
17	Reveal right	35	Cover up, left	* Does not work on monitors set to 32 bits	
18	Reveal down, right	36	Cover up, right		

MIAW Types for Windows

No.	Description	Movable	Close Box	Maximize	Minimize
-1	Default	Yes	Yes	No	No
0	Standard	Yes	Yes	No	No
1	Alert Box	No	No	No	No
2	Rectangle	No	No	No	No
3	Rectangle	No	No	No	No
4	Document	Yes	Yes	No	No
5	Document	Yes	Yes	No	No
8	Document	Yes	Yes	Yes	No
12	Document	Yes	Yes	Yes	No
16	Document	Yes	Yes	No	No
49	Palette	Yes	Yes	No	No

MIAW Types for Mac

No.	Description	Resize Box	Movable	Close Box	Stretch
-1	Default	Yes	Yes	Yes	No
0	Standard	Yes	Yes	Yes	No
1	Alert box	No	No	No	No
2	Rectangle	No	No	No	No
3	Rectangle/shadow	No	No	No	No
4	Document	Yes	Yes	No	No
5	Document	Yes	No	No	No
8	Document	Yes	Yes	Yes	Yes
12	Document	Yes	Yes	No	Yes
16	Curved border Box	Yes	Yes	No	No
49	Palette	Yes	Yes	No	No

Hexadecimal Color Values

F0F8FF	Alice blue	00CED1	Dark turquoise	D3D3D3	Light gray	FFC0CB	Pink
FAEBD7	Antique white	9400D3	Dark violet	FFB6C1	Light pink	DDA0DD	Plum
00FFFF	Aqua	FF1493	Deep pink	FFA07A	Light salmon	B0E0E6	Powder blue
7FFFD4	Aquamarine	00BFFF	Deep sky blue	20B2AA	Light sea green	800080	Purple
F0FFFF	Azure	696969	Dim gray	87CEFA	Light sky blue	FF0000	Red
F5F5DC	Beige	1E90FF	Dodger blue	B0C4DE	Light steel blue	BC8F8F	Rosy brown
FFE4C4	Bisque	B22222	Firebrick	FFFFE0	Light yellow	41690	Royal blue
000000	Black	FFFAF0	Floral white	00FF00	Lime	8B4513	Saddle brown
0000FF	Blue	228B22	Forest green	32CD32	Lime green	FA8072	Salmon
8A2BE2	Blue violet	FF00FF	Fuchsia	FAF0E6	Linen	F4A460	Sandy brown
A52A2A	Brown	DCDCDC	Gainsboro	FF00FF	Magenta	2E8B57	Sea green
5F9EA0	Cadet blue	F8F8FF	Ghost white	800000	Maroon	FFF5EE	Seashell
7FFF00	Chartreuse	FFD700	Gold	0000CD	Medium blue	A0522D	Sienna
D2691E	Chocolate	DAA520	Goldenrod	BA55D3	Medium orchid	C0C0C0	Silver
FF7F50	Coral	808080	Gray	9370DB	Medium purple	87CEEB	Sky blue
6495ED	Cornflower blue	008000	Green	191970	Midnight blue	6A5ACD	Slate blue
FFF8DC	Cornsilk	ADFF2F	Green yellow	FFE4E1	Misty rose	708090	Slate gray
DC143C	Crimson	F0FFF0	Honeydew	FFE4B5	Moccasin	FFFAFA	Snow
00FFFF	Cyan	FF69B4	Hot pink	FFDEAD	Navajo white	00FF7F	Spring green
00008B	Dark blue	CD5C5C	Indian red	000080	Navy	4682B4	Steel blue
008B8B	Dark cyan	4B0082	Indigo	FDF5E6	Old lace	D2B48C	Tan
B8860B	Dark goldenrod	FFFFF0	Ivory	808000	Olive	008080	Teal
A9A9A9	Dark gray	F0E68C	Khaki	6B8E23	Olive drab	D8BFD8	Thistle
006400	Dark green	E6E6FA	Lavender	FFA500	Orange	FF6347	Tomato
BDB76B	Dark khaki	FFF0F5	Lavender blush	FF4500	Orange red	40E0D0	Turquoise
8B008B	Dark magenta	7CFC00	Lawn green	DA70D6	Orchid	EE82EE	Violet
FF8C00	Dark orange	FFFACD	Lemon chiffon	EEE8AA	Pale goldenrod	F5DEB3	Wheat
9932CC	Dark orchid	ADD8E6	Light blue	98FB98	Pale green	FFFFFF	White
8B0000	Dark red	F08080	Light coral	FFEFD5	Papaya whip	F5F5F5	White smoke
E9967A	Dark salmon	E0FFFF	Light cyan	FFDAB9	Peach puff	FFFF00	Yellow
8FBC8F	Dark sea green	90EE90	Light green	CD853F	Peru	9ACD32	Yellow green

Key Codes

123	Left
124	Right
125	Down
126	Up

ASCII Character Codes

3	Enter
8	Delete
9	Tab
10	Line Feed
13	Return
32	[space]
33	!
34	“
35	#
36	$
37	%
38	&
39	‘
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48..57	0..9
58	:
59	;
60	<
61	=
62	>
63	?
64	’
65..90	A..Z
91	[
92	\
93	]
94	^
95	_
96	`
97..122	a..z
123	{
124	|
125	}
126	~
127	DEL

Special Edition

Using Macromedia Director 8

Gary Rosenzweig

QUE®

A Division of Macmillan Publishing USA
201 W. 103rd Street
Indianapolis, Indiana 46290

Contents at a Glance

Introduction

00 What's New in Director 8 7

I Director Basics

1 Animation with Director 13
2 Presentations with Director 55

II Members in Detail

3 Bitmap Members 79
4 Text and Field Members 103
5 Sound Members 117
6 Digital Video 127
7 Vector Members 135
8 Other Member Types 149

III Authoring in Director

9 The Director Environment 163
10 Properties of Sprites and Frames 183
11 Advanced Techniques 199

IV Using Basic Lingo

12 Learning Lingo 217
13 Essential Lingo Syntax 235

V Using Behaviors

14 Creating Behaviors 273
15 Graphic Interface Elements 305

VI Using Lingo to Control Media

16 Controlling Text 335
17 Controlling Sound 373
18 Controlling Bitmaps 383
19 Controlling Video 401
20 Controlling Vector Graphics 411

VII Using Advanced Lingo

21 Controlling the Director Environment 431
22 Shockwave and Internet Access 451
23 Object-Oriented Programming 467
24 Movies in a Window and Alternatives 479
25 Xtras 497
26 Developing for Developers 515

VIII Using Director to Create Professional Applications

27 Educational Applications 525
28 Business Applications 547
29 Graphics Applications 569
30 Sound Applications 589
31 Shockwave Applets 603
32 Games 615

IX Finishing a Project

33 Debugging 649
34 Performance Issues 671
35 Cross-Platform Issues 689
36 Delivering the Goods 699

X Advanced Topics

37 Creating Java Applets 711
38 Creating Multiuser Applications 729

XI The Lingo Reference Guide

A What's on the CD-ROM 745
B Director Glossary 749
C Online Resources 763
D Tables and Charts 769
E Lingo by Subject 783
F Behavior Library Quick Reference 801
G Keyboard Shortcuts 809
H Guide to Xtras 827
I Lingo Reference 869

SPECIAL EDITION USING MACROMEDIA DIRECTOR 8

Copyright © 2000 by Que

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

International Standard Book Number: 0-7897-2334-4

Library of Congress Catalog Card Number: 00-100575

Printed in the United States of America

First Printing: July 2000

02 01 00 4 3 2 1

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Que cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

This publication was produced using the Advent 3B2 Publishing System.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an "as is" basis. The author and the publisher shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD or programs accompanying it.

Associate Publisher
Greg Wiegand

Acquisitions Editor
Karen Whitehouse

Development Editors
Kezia Endsley
Nick Goetz

Managing Editor
Thomas F. Hayes

Project Editor
Tricia A. Sterling

Copy Editor
Julie McNamee

Indexer
Erika Millen

Proofreader
Maribeth Echard

Technical Editor
Allison Kelsey

Team Coordinator
Julie Otto

Media Developer
Brandon Penticuff

Interior Designer
Ruth Lewis

Cover Designers
Dan Armstrong
Ruth Lewis

Production
Brandon Allen
Susan Geiselman
Brad Lenser
Cheryl Lynch

CONTENTS

Introduction 1

00 What's New in Macromedia Director 8 7

I Director Basics 11

1 Animation with Director 13

An Introduction to Director 14
The Theater Metaphor 14
Playback Options 16
Movie Example 17

The Cast 22
Types of Cast Members 22
Cast Window Properties and Settings 24
Multiple Cast Libraries 26
External Cast Libraries 27

The Stage 28

The Score 30
Elements of the Score Window 30
Customizing the Score Window 33
Score Window Preferences 34
Sprite Preferences Dialog Box 35
Other Score and Sprite Settings 36

Other Controls 36
The Control Panel 37
The Tool Palette 38
The Property Inspector 40
Sprite Overlays 40
The Toolbar 41

Animating with the Cast, Stage, and Score 42
Step Recording 42
Real-Time Recording 43
Space to Time Recording 44
Cast to Time Recording 44
Tweening 45
Working with Sprites 48
Grids and Guides 49
Exporting Animations 51

Troubleshooting Animation 53

Did You Know? 54

2 Presentations with Director 55

Designing Screens 56
Planning a Consistent Design 56
Tools for Screen Design 58

Linear Presentations 59
Creating Your Presentation 59
Adjusting the Tempo 60

Nonlinear Presentations 62
Adding Back and Home Buttons 63
Adding a Menu Screen 64

Advanced Presentations 66
Play and Return 66
Button Highlights 66
Cursor Changes 67
Transitions 67
Adding Sounds 68
Adding Animation in Presentations 70

Showing Your Presentation 71
Using Director to Show Your Presentation 71
Creating a Standalone Projector 72
Creating a Shockwave Movie 75

Troubleshooting Director 75

Did You Know? 76

II Members in Detail 77

3 Bitmap Members 79

Types of Bitmaps 80
Using Palettes 81
Choosing the Bit Depth 82

Importing Bitmaps 83

Using the Paint Window 85
Lasso and Marquee Tools 85
Registration Point Tool 86
Eraser Tool 87
Hand Tool 87
Zoom Tool 87
Eyedropper Tool 88
Paint Bucket Tool 88
Text Tool 88
Pencil Tool 89
Air Brush Tool 89
Brush Tool 90
Arc Tool 90
Line Tool 90
Shape Tools 91
Color Chips 91
Gradients 91
Patterns 92
Line Weight Settings 92
Paint Window Inks 93
Paint Window Preferences 94
The Effects Toolbar 94

Bitmap Member Properties 95

Onion Skinning 97

PhotoShop Filters 98

Troubleshooting Bitmap Images 100

Did You Know? 101

4 Text and Field Members 103

Using Text Members 104
Text Editing 105
Text Importing 106
Text Member Options 107
The Text Inspector 110
The Font Dialog Box 110
The Paragraph Dialog Box 111

Using Field Members 112

Knowing When to Use Text and Fields 113

Using Font Members 114

Troubleshooting Text and Field Members 115

Did You Know? 116

5 Sound Members 117

Sound File Formats 118

Frequency and Bit Depth 119

Internal Sound Members 120

External Sounds 121

Shockwave Audio 121

Using Sound in Director 123
Using the Score to Play Sound 123
Using Library Behaviors to Produce Sound 123

Waiting for Sounds and Cue Points 124

Troubleshooting Sounds 125

Did You Know? 125

6 Digital Video 127

Using Digital Video Formats 128
File Types 128
Compression 129
Data Rate 130

Importing Digital Video 130

Understanding Digital Video Settings 130

Working with Digital Video 132

Making Custom Video Controls 133

Troubleshooting Digital Video 133

Did You Know? 134

7 Vector Members 135

Using Shape Members 136

Using Vector Members 139
Vector Shape Editing Window 139
Vector Shape Properties 141
Vector Shape Techniques 143

Using Flash Members 144

Troubleshooting Vector Members 146

Did You Know? 147

8 Other Member Types 149

Creating Pushbuttons 150

Creating Check Boxes 151

Creating Radio Buttons 152

Using Palette Members 153

Adding Cursors 156
Built-In Cursors 156
Custom Bitmap Cursors 156
Animated Cursors 157

Using Animated GIFs 158

Using QuickTime VR 158

Adding New Cast Member Types with Xtras 159

Troubleshooting Other Member Types 159

Did You Know? 160

III Authoring in Director 161

9 The Director Environment 163

Ways to Work in Director 164
Planning Your Attack 164
Creating Documentation 165
Division of Labor 166
Hardware Network 166
Teamwork 166

Setting Preferences 167
General Preferences 167
Network Preferences 168
Editors Preferences 169
Script Window Preferences 170
Movie Properties 170
Movie Playback Properties 172
Memory Inspector 172

Publish Settings 173
The Formats Tab 173
The General Tab 174
The Shockwave Tab 175
The Compression Tab 176
The Shockwave Save Tab 177
Publishing 177

Using the Director Library Palette 178

Using Xtras 178

Getting Help 179

Troubleshooting the Director Environment 179

Did You Know? 180

10 Properties of Sprites and Frames 183

Controlling Frame Tempo 184

Using Frame Palettes 185

Adding Frame Transitions 187

Setting Sprite Inks 188

Using the Sprite Blend 191

Setting Sprite Colors 192

Adjusting the Sprite Shape 192

Other Sprite Properties 195
Editable Property 195
Moveable Property 195
Trails Property 195
Lock Property 196

Troubleshooting Properties of Sprites and Frames 197

Did You Know? 197

11 Advanced Techniques 199

Using the Behaviors Library 200
- Animation Behaviors 200
- Controls Behaviors 201
- Internet Behaviors 201
- Media Behaviors 202
- Navigation Behaviors 202
- Paintbox Behaviors 202
- Text Behaviors 202

Building Your Own Behaviors 202
- Creating More Complex Behaviors 204
- Adding Events to Behaviors 205
- Adding Actions to Behaviors 206

Using Film Loops and Linked Movies 208

Color Cycling 209

Shockwave Streaming 211

Troubleshooting Advanced Techniques 213

Did You Know? 213

IV Using Basic Lingo 215

12 Learning Lingo 217

What Is Lingo? 218

First Steps with the Message Window 218

Understanding Script Types 220

Using Messages and Handlers 223
- Types of Messages 223
- Message Hierarchy 224
- Creating Your Own Handlers 225

Using Variables 226

Writing Lingo Code 228
- The Lingo Programmer 228
- Programming As Problem Solving 229
- Solving Smaller Problems 229
- Setting Up Your Script Members 230
- Writing Your Code 231

Troubleshooting Lingo 232

Did You Know? 232

13 Essential Lingo Syntax 235

Using Number Variables 236
- Performing Operations 236
- Integers and Floats 237
- Precedence 238
- Operations and Variables 239
- Functions 240

Using String Variables 241
- Chunk Expressions 241
- Manipulating Strings 243

Comparing Variables 244

Using Handlers 246
- Event Handlers 246
- Custom Handlers 247
- Functions 248

Using *If...Then* Statements 250
- Simple *If* Statements 250
- *Case* Statements 251
- Nested *If* Statements 252
- Logical Operators 252

Using *Repeat* Loops 253
- *Repeat With* 253
- *Repeat While* 254
- Other *Repeat* Commands 255
- Repeating Forever 256

Using Lingo Navigation Commands 257
- *Go* 257
- Using *Go* 259
- *Play* 259
- Leaving the Movie 260

Controlling Sprite Properties 260
- Sprite Location 261

Sprite Member 262
Sprite Rectangle 262
Sprite Ink 263
Sprite Color 264

Controlling Member Properties 265

Using List Variables 265
Linear Lists 265
Property Lists 267

Using Frame Scripts 268

Troubleshooting Lingo Syntax 268

Did You Know? 269

V Using Behaviors 271

14 Creating Behaviors 273

Controlling a Single Sprite 274
Sprite Messages 274
Properties 275
Using *me* 276

Creating Simple Behaviors 277
Navigation Behaviors 277
Rollover Behaviors 279
Animation Behaviors 282

Creating a Simple Button Behavior 283

Using Complete Behaviors 285
Behavior Descriptions 285
Behavior Property Description List 286
Automatic Property Setting 288
Controlling Where Behaviors Are Used 289

Tooltips 290

Building a Complete Button Behavior 290
Creating the Parameters 291
Writing the Event Handlers 293

Creating Animation Behaviors 297
Bouncing off Walls 297
Adding Gravity 299

Communicating Between Behaviors 302
Sending a Message to a Sprite 302
Sending a Message to All Sprites 303
Sending a Message to Specific Behaviors 303

Troubleshooting Creating Behaviors 304

Did You Know? 304

15 Graphic Interface Elements 305

Creating Display Rollovers 306

Using Check Boxes 308

Using Radio Buttons 310

Dragging Sprites 313
Simple Drag Script 313
A Better Drag Behavior 314
Click, Drag, and Lock 315
Drag and Throw 318

Creating Sliders 320

Creating Progress Bars 326

Creating Graphical Pop-Up Menus 328

Troubleshooting Graphic Interface Elements 332

Did You Know? 332

VI Using Lingo to Control Media 333

16 Controlling Text 335

Using Strings and Chunks 336
Building Strings 336
Using Chunk Expressions 337
Comparing Strings 338
String Functions 339
String Constants 342
Text References 343

Using Text Members and Fields 343
Fields 343
Text Members 347

Nintendo Exhibit 1029
Page 00180

Creating Text Lists 349

Creating Text Pop-Up Menus 351

Using Keyboard Input 355
Keyboard Lingo 355
Recognizing the Return 356
Restricting Input 357
Capturing Keystrokes 358

Using Rich Text Format 360

Using HTML and Tables 360
A Simple HTML Application 360
Applying Tables 362

Using HTML and Hypertext 364
Setting and Using Hyperlinks 364
Modifying Hyperlinks with Lingo 366

Using Text Files and the FileIO Xtra 367

Troubleshooting Text and Strings 370

Did You Know? 371

17 Controlling Sound 373

Using Lingo's Sound Commands 374
Old Sound Lingo 374
New Sound Commands 375
Other Sound Commands 376

Using Cue Points 378

Playing External Sounds 379

Using Shockwave Audio 380

Troubleshooting Sound 382

Did You Know? 382

18 Controlling Bitmaps 383

Distorting Sprites 384
The *rect* Property 384
The *rotation* Property 384
The *flipH* and *flipV* Properties 386
The *skew* Property 386
The *quad* Property 387
Other Distortion Properties 389

Adding 3D Effects 390
Shrinking Sprites 390
Shrinking Sprites with Movement 391
Using the *quad* Property to Create Illusions 392
Mapping Sprites 395

Manipulating Bitmap Members 396

Troubleshooting Bitmaps 399

Did You Know? 399

19 Controlling Video 401

Using Video Commands 402
Member Properties 402
Sprite Properties 403
Masks 403

Building Video Controls 404

Using Other Video Techniques 406

Troubleshooting Video 410

Did You Know? 410

20 Controlling Vector Graphics 411

Using Flash Member Lingo 412

Using Vector Shape Lingo 414

Building Vectors with Lingo 417

Using Vectors for Collision Detection 426

Troubleshooting Vector Lingo 428

Did You Know? 428

VII Using Advanced Lingo 429

21 Controlling the Director Environment 431

Using Menus 432
Creating Menus 432
Controlling Menus 434

Using Cursors 435
Using Built-In Cursors 435
Using Custom Bitmap Cursors 437
Using the Cursor Xtra 437

Using Timeouts 438
Traditional Timeouts 438
Timeout Objects 439

Learning About the Computer 440

Telling Time 441

Memory Management 444
Member Loading 444
Memory Information 446

Launching Other Applications 447

Quitting and Shutting Down 447

Troubleshooting Environment Lingo 448

Did You Know? 449

22 Shockwave and Internet Access 451

Controlling the Web Browser with Lingo 452

Getting Text over the Internet 454

Sending Text 456
Using *getNetText* to Send Text 457
Using *postNetText*> to Send Text 458

Working with Browsers 459
<EMBED> and <OBJECT> Tag Parameters 459
JavaScript to Shockwave 460
Shockwave to JavaScript 462

Using Shockwave Preference Files (Cookies) 462

Controlling Streaming Media 463
Checking Streaming Status 463
Loading in New Media 465

Troubleshooting Shockwave Lingo 465

Did You Know? 466

23 Object-Oriented Programming 467

What Is an Object? 468

Reasons to Use Objects 469
Objects Organize Your Code 469
Objects Persist over Time 469
Objects Are Easy to Test 470
Objects Make Coding Easier and More Efficient 470
Objects Can Be Reused 470

Creating an Object in Lingo 471

Creating Object Properties 473

Using OOP 474

Using Ancestors 475

Troubleshooting OOP 477

Did You Know? 477

24 Movies in a Window and Alternatives 479

Using MIAWs 480
Creating a MIAW 480
MIAW Properties 481
Window Commands 483
MIAW System Properties 483
MIAW Event Handlers 484

Creating Dialog Boxes 485
Confirmation Dialog Boxes 485
Alert Dialog Boxes 486
Text Input Dialog Boxes 486

Creating Oddly Shaped MIAWs 487

Creating Self-Contained MIAWs 488

Other Uses for MIAWs 488

Using Linked Movies 489

Using MUI Xtra Dialog Boxes 490
- Creating a File Open Dialog Box 490
- Creating a File Save Dialog Box 490
- Creating a Get URL Dialog Box 491
- Creating Alert Boxes 491
- Creating Custom MUI Dialog Boxes 492

Troubleshooting MIAWs and Alternatives 495

Did You Know? 496

25 Xtras 497

What Are Xtras? 498

Xtras That Come with Director 8 498

Third-Party Xtras 500
- AlphaMania and PhotoCaster 500
- Audio Xtra 501
- Beatnik Xtra 502
- Buddy API Xtra 503
- CD Pro Xtra 506
- DirectOS Xtra 506
- DirectTransitions/DirectTransitions 3D Xtras 507
- DirectXport Xtra 508
- DM Tools 508
- LiveCD 508
- MasterApp Xtra 508
- PDF Xtra 509
- PiMz Image Xtra 509
- Popup Xtra 509
- Print-O-Matic Lite 510
- Relaunch Utility 510
- TextCruncher Xtra 511

Using Xtra Lingo 511

Using Xtras in Shockwave 512

Other Xtras 513

Troubleshooting Xtras 514

Did You Know? 514

26 Developing for Developers 515

Score Recording 516
- Writing to the Score 516
- Score Recording Tools 517
- Setting Behaviors and Parameters Through Lingo 519

Creating MIAW Xtras 519

Using Behavior Libraries 520

Troubleshooting Developing for Developers 520

Did You Know? 521

VIII Using Director to Create Professional Applications 523

27 Educational Applications 525

Creating a Matching Game 526

Making a Drawing Activity 531

Creating Overlays 536

Creating a Geography Quiz 538

Creating Standardized Tests 542

Troubleshooting Educational Applications 546

Did You Know? 546

28 Business Applications 547

Creating Database Applications 548
- The Main Menu Screen 548
- The Database Entry Screen 549
- Navigating Through the Database 551
- Listing Records in the Database 553

Creating Graphs and Pie Charts 555
Bar Graphs 556
Pie Charts 557

Creating Questionnaires 560

Creating Computer-Based Training Programs 564

Troubleshooting Business Applications 568

Did You Know? 568

29 Graphics Applications 569

Creating Slideshows 570

Panning Large Images 573

Making Scrollbars for Large Images 576

Zooming In on Large Images 582

Utilizing Ink and Color Manipulation 585

Troubleshooting Graphics Applications 587

Did You Know? 588

30 Sound Applications 589

Creating a Piano Keyboard 590

Creating a Player Piano 592

Simulating 3D Sound 594

Creating Volume Controls 595
Volume Slider 596
Volume Button 598

Troubleshooting Sound Applications 600

Did You Know? 600

31 Shockwave Applets 603

Creating Navigation Pages 604

Creating Navigation Bars 606

Creating Shockwave Ads 608

Processing and Displaying Information 609

Troubleshooting Shockwave Applets 613

Did You Know? 613

32 Games 615

Creating a Matching Game 616

Creating a Sliding Puzzle Game 619

Creating a Falling Objects Game 622

Emulating a Shooting Gallery 625

Creating Sprite Invaders 629
Creating the Invading Sprites 629
Creating the Ship 632
Creating the Frame Script 635

Creating Trivia Games 635

Creating a Blackjack Game 638

Troubleshooting Games 645

Did You Know? 646

IX Finishing a Project 647

33 Debugging 649

Writing Good Code 650
Commenting Your Code 650
Using Descriptive Names 652
Writing Error-Proof Code 655
Error-Handling Lingo 662

Using Lingo Debugging Tools 663
Using the Debugger 664

Using the Watcher 665
Using the Script Window 666
Using the Message Window 666
Using Trace 667
Using Other Debugging Methods 668

Testing Your Code 668
Test Early and Often 669
Testing In-House 669
Beta Testing 669

Did You Know? 670

34 Performance Issues 671

Designing for a Target Machine 672
Compromise 672
Sample Requirements 673

Recognizing Issues Affecting Performance 674
Hardware Considerations 675
Software Considerations 677
Network Considerations 679

Improving Performance 680
Member Loading 680
Shockwave File Size and Streaming 681
Compensating with Alternatives 681
Trimming Media 682
Cutting Out Dead Wood 683

Optimizing Projector Performance 684

Optimizing Lingo Performance 685

Did You Know? 687

35 Cross-Platform Issues 689

Developing for the Mac and Windows 690
Fonts 690
Color Palettes 694
Display Brightness 695
Digital Video 695
File Pathnames 695
Non–Cross-Platform Lingo 696

Developing for Netscape Navigator and Microsoft Internet Explorer 697

The Cross-Platform Checklist 698

Did You Know? 698

36 Delivering the Goods 699

Making Projectors 700
Stub Projectors 700
Compressed Projectors 701
Shockwave Projectors 701

Making Shockwave Movies 702
Netscape Navigator <EMBED> Tag 702
Microsoft Internet Explorer <OBJECT> Tag 704
Using Both the <EMBED> and <OBJECT> Tags 705

Building CD-ROMs 706

Making Installers 707

Building Screen Savers 707

Troubleshooting Delivering the Goods 708

Did You Know? 708

X Advanced Topics 709

37 Creating Java Applets 711

Understanding Save As Java Basics 712
Translating an Existing Movie into an Applet 713
Building a Movie with Java in Mind 713

Understanding What Works and What Doesn't with the Xtra 714
Supported Member Types 714
Using Sprites in Java 716
Java-Supported Fonts 717
Cast Library Issues 717
Network Lingo Issues 717
Other Lingo Issues 718
Transitions Supported in Java 719

A Look Inside the Save As Java Xtra Process 719

Using the Save As Java Xtra 720

Learning Java Building Techniques 721
Hiding Lingo from Java 722
Embedding Java Code 722

A Save As Java Example 722
Converting to a Java Applet 724
Debugging the Applet 725

Finishing a Java Applet 726
Bundling the Applet 726
Compatibility Issues 727
Problems and More Information 727

Troubleshooting Java Applets 727

38 Creating Multiuser Applications 729

Setting Up the Multiuser Server 730
Running the Multiuser Server 730
Configuring the Server 731

Using the Multiuser Behaviors 732

Learning Multiuser Lingo 734

Creating Your Own Multiuser Application 736

Using Other Functions of the Multiuser Xtra 739
Using the Multiuser Database Features 739
Peer-to-Peer Connectivity 740

Troubleshooting Multiuser Applications 741

Did You Know? 741

XI The Lingo Reference Guide 743

A What's on the CD-ROM 745

B Director Glossary 749

C Online Resources 763

D Tables and Charts 769

E Lingo by Subject 783

F Behavior Library Quick Reference 801

G Keyboard Shortcuts 809

Macintosh Shortcuts 810

Windows Shortcuts 818

H Guide to Xtras 827

I Lingo Reference 869

Index 973

About the Author

Gary Rosenzweig is the Chief Engineer, founder, and owner of CleverMedia, a game and multimedia development company in Denver, Colorado. This is his fifth book on Director.

Before the multimedia industry started, he was a computer science student at Drexel University in Philadelphia. There, he took a greater interest in journalism than computers. After working for years in various positions at the student newspaper, he eventually rose to the position of Editor-in-Chief.

Gary followed his journalistic interests to the University of North Carolina at Chapel Hill, where he earned a master's degree. As an undergraduate, he found that he was someone who knew a lot about publishing at a school where everyone was a computer expert. In graduate school, he found that he was someone who knew a lot about computers at a school where everyone was involved in publishing.

While in school, he got his first taste of a scripting language with Hypercard 1.0. Although class projects were done in Pascal, C, and other "serious" programming languages, Gary found that he was able to build programs for freelance jobs with Hypercard in a fraction of the time.

After school, rather than enter one field or another, Gary combined both and began freelancing for publications to build multimedia products. Shortly thereafter, he moved to Denver to take a job at Ingenius, an educational news service partially owned by Reuters. It was there that he began using Director all day, every day.

As a serious developer, Gary went to bookstores to find books on Director. Seeing none, he decided to write one. He then published the book, for free, on the Internet. Soon after that, he was paid to rewrite and update the book as *The Comprehensive Guide to Lingo*.

After the book was published, Gary left Ingenius and started his own company, CleverMedia. This was made possible because of Shockwave, which gives users the capability to play Director movies on the Web. Demand for Shockwave applets rose and CleverMedia was hired by many companies, both large and small, to build Shockwave applets. Today, CleverMedia owns four of the largest Shockwave and Flash game sites on the Web, and creates games for many of the others.

Gary lives in Denver, Colorado, with wife Debby, a cat named Lucy, and a dog named Natasha. Other than computers and the Internet, he also enjoys film, camping, classic science fiction books, and writing.

Personal Home Page: `http://clevermedia.com/rosenz/`

Email: `http://clevermedia.com/email.html`

Dedication

To my Grandmom, Rebecca Jacob, who bought me my first computer.

ACKNOWLEDGMENTS

Very special thanks go to William Follett, Jay Shaffer, and Brian Robbins, my co-workers at CleverMedia. They make developing games even more fun than playing them. Also, thanks to my other Colorado friends and Director users: John Nyquist, Rob Martin, Andy Schmit, Jenny Thomas, and Heath Burr for ideas and encouragement.

Thanks also to the team at Que: Karen Whitehouse, Beth Millett, Kezia Endsley, Nick Goetz, Tricia Sterling, Julie McNamee; and thank you to my technical editor, Allison Kelsey.

Thanks to the many people who participate in the Hopper-Ex and DXR mailing lists, DOUG and the Director 8 Beta List: Darrel Plant, Colin, Holgate, Ron Manke, Ely Greenfield, Hudson Ansley, Alex Zavatone, Bruce Epstein, Phillip Kerman, Emmanuel Beuque, Mark Reijnders, Terry Schussler, Rebecca Lovelace, Scott Flowers, Dave Cole, James Newton, Andrew White, Roy Pardi, Alan Levine, Ron Bearry, Eric Coker, Dorian Dowse, Julian Baker, Marvyn Hortman, Oogie McGuire, Raul Silva, Paul Hertz, Rob Dillon, Gretchen MacDowel, Matt Craig, Miles Lightwood, Mark Hagers, Glenn Picher, Peter Lundholm, Simon Biggs, Ian Chia, Zac Belado, Patrick McClellan, and all the rest.

Thanks to all of the Macromedia Director team, including John Thompson, Buzz Kettles, Karen Silvey, James Khazar, Chris Nuuja, David Mendels, Jake Sapirstein, Glenn Ruehle, Mike Edmunds, Werner Sharp, Gordon Smith, Dan Sadowski, Tracy Stampfli, Tom Higgins, Sarah Allen, Peter Grandmaison, Perihan Cumali, Lalit Balchandani, Ken Day, Karl Miller, Joe Schmitz, Jim Corbett, Jenny Suter, John Dowdell, Jay Armstrong, David Calaprice, Christophe Leske, Chris Nuuja, Alex Lelievre, Kraig Mentor, Greg Yachuk, Eliot Greenfield, Doug Wyrick, and the rest.

Thanks to my family for their continuing lifetime of support: Jacqueline, Jerry, and Larry Rosenzweig; Rebecca Jacob; Barbara and Richard Shifrin.

The most thanks go to my wife, Debby, for her love and support.

Music is a very important part of my writing environment. Here are a few of the artists whom I listened to while writing. I recommend them all: Leah Andreone, The Beatles, Ludwig van Beethoven, The Cogs, Cowboy Junkies, Dixie Chicks, Sean Colvin, The Donnas, Juliana Hatfield, Hole, Diana Krall, Letters to Cleo, Magnapop, Alanis Morrisette, The Muffs, Liz Phair, Poe, Sixpence, None the Richer, Sleater-Kinney, and the soundtracks to the movies *Bandits* and *Run Lola Run*.

Nintendo Exhibit 1029
Page 00188

Tell Us What You Think!

As the reader of this book, *you* are our most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

As an Associate Publisher for Que, I welcome your comments. You can fax, email, or write me directly to let me know what you did or didn't like about this book—as well as what we can do to make our books stronger.

Please note that I cannot help you with technical problems related to the topic of this book, and that due to the high volume of mail I receive, I might not be able to reply to every message.

When you write, please be sure to include this book's title and author as well as your name and phone or fax number. I will carefully review your comments and share them with the author and editors who worked on the book.

Fax: 317-581-4666

Email: office_que@mcp.com

Mail: Associate Publisher
Que
201 West 103rd Street
Indianapolis, IN 46290 USA

FOREWORD

A large community of Director users has developed over the past few years. People from diverse backgrounds have learned, experimented, and grown with Director to be able to make a living doing what they enjoy. A few in the Director community have gone beyond this and shared their expertise or pushed the boundary of what was thought possible with Director. Gary Rosenzweig has done both to a unique degree.

I first became aware of Gary's educational efforts back in 1994 when I saw his book on Director that he had self published online. At that time, the Director book market had not yet proven itself and it was hard to convince a publisher to stand behind a Director title. Gary did not let this deter him and he provided a valuable service to aspiring Director users by making his first book available online for no charge.

My next early encounter with Gary was through his Web site, `www.clevermedia.com`, where he provided many Director Shockwave movies that could be enjoyed by all. As early as 1995, Gary had innovative and imaginative titles such as "Pretty Good Golf," which packs a lot of activity into a very small size; "CleverChess," the very first real multiplayer Internet game in Shockwave; "Space Pirate," a multiplay game so popular it was elevated to its own private site.

Eventually, Gary and I met in person at a Macromedia Developer User Conference, where developers from around the world come to learn more about Macromedia tools and to exchange ideas. I was pleased to find out that he and I share a strong interest in education and gaming. We continued chatting on these topics as we met over the years at other conferences. In addition to informal discussions with other developers, Gary has shared his extensive expertise at these conferences as a presenter.

While many of you may not get a chance to meet Gary at a conference, you do have this excellent book. It's a big book because it covers just about everything you might choose to do in Director—from basic animation to beginning and advanced Lingo programming. The Director community is fortunate to have one of its most innovative developers willing to share his knowledge so that you can benefit from his many years of experience.

John (JT) Thompson

Macromedia Chief Scientist, Inventor of Lingo