

Report to LEXIS/NEXIS on LII Standards and Practices in Building Legal Course Materials with Folio Views

Overview of the Design Process (in Relation to Full Infobase Construction)

Background -- Types of Legal Material and Categories of Use with Which the LII Has Experience

For over three years, Cornell's Legal Information Institute has been building hypertext legal materials. Believing that among the several types of documents important to legal education complicated codes (statutes, sets of procedural or evidentiary rules, and the like) gained the most from presentation in rich electronic format, the LII has focused most of its energies on them. The Institute's current catalog of titles (see Appendix 1) strongly reflects that emphasis. The LII has, however, prepared an original reference work ("Basic Legal Citation"). It has also produced small collections of important decisions in four areas for the current academic year.

Like the Institute's experience this report on how the LII creates hypertext course materials as Folio Infobases will focus on the representation of existing works (statutes, regulations, treaties, judicial opinions). Many of the same considerations apply as well to the creation of original works (reference works, practice support systems and the like); however, the design of original hypertext works brings into play such an array of additional factors (many of them work specific) that no simple extrapolation of the LII's "standards" and "practices" to that process is possible.

Over-all Architecture -- The Critical Role of Prototyping (Resisting the Temptation to Import and Revise)

Since Folio Views will import word processor files with "full value," i.e., bringing in both structural and appearance elements, it invites the hypertext builder to import early and, thereafter, modify within the Views working space. Such a process runs at least two significant risks. The first is that structural decisions that had no practical impact so long as the work existed solely to be printed may be imported transparently (invisibly) into Views and prove to be suboptimal or even dysfunctional for the electronic version. A second risk is that the full potential of the new medium will not be explored because of the strong initial influence of the print version of the work. For these and other reasons, the LII's design process for any work begins with the creation of small prototype infobases, hand-crafted in Views, of a few representative portions.

This prototyping stage is the point at which we try to identify and resolve pervasive issues of organization and presentation. For example, if most or all of the sections of a code have official comments or historical notes (or both), this is the point to experiment with how the relationships between successive sections and these related texts should be represented. The word processing file, reflecting print presentation practice, will have them strung in sequence, but the hypertext environment allows section to follow section giving users the opportunity to follow the logical path that often unfolds that way, with links providing an alternative relevancy path for those who want to dig behind a particular section. However, even this construct entails alternatives, including: 1) a jump link structure with code or rule sections held in different portions of the same infobase or two separate infobases and 2) a query link structure allowing each section's comments to be adjacent but hidden in a query link triggered view of the infobase that would pull up all records in a "sections" group. The point here is simply that structural issues of this importance are best addressed before the full work has been assembled. Exploration through small prototypes is how the LII resolves fundamental issues like these.

The Importance of Moving Back and Forth Between Infobase and Flatfile

While prototyping and certain other elements of infobase construction are accomplished within the Views work space, many of the global operations of infobase construction are accomplished by the LII using the *fsr* utility and other editing tools to work on the infobase flat file. Once the over-all architecture and data structure of the work have been established, construction of the full work involves, typically, a process that moves back and forth between Views and a flat file.

If the initial text data file is a word-processor file (*e.g.*, MS Word or WordPerfect), the first move into Views takes advantage of the appropriate import filter. (This would be true as well of an on-line or CD-ROM download.) The aim in this initial step is to capture all structure and other value held in the word-processor file. The resulting infobase or a meaningful sample is then exported immediately to flat file for diagnosis. Comparison and analysis of this initial flat file with the flat file of the ultimate prototype yields a list of conversion steps -- most of which are most efficiently done with a good editor and macros, the *fsr* utility, or a combination of the two.

In designing the conversion process, the first step we take is to identify structural or typographic and font cues that can be keyed on to implement the ultimate data structure (records, levels, jump destinations, fields and so on). The second step we take is to strip out all unnecessary codes. Since the LII's formatting approach is to generate fonts and white space and other screen and print presentation elements almost completely through styles, this stripping step often includes the removal of double returns, tabs, and countless font specifications. Styles that operate below the paragraph (*e.g.*, character or field styles) may involve converting specific character or font codes -- which therefore should not be stripped. If, for

example, one or more types of italicized words or phrases are to receive that or some other specialized treatment the relevant codes are not stripped from the flat file but left for the next stage. Finally, we undertake straight translations. In some cases they can be performed without exercise of individual editorial inspection. If, for example, all italicized words are to be converted to a character style denominated “Italics” that can be handled by a global search and replace operation. If, on the other hand, some italicized words are foreign words or phrases and others are terms of art specially defined in the code and distinguishing between the two may hold value for some users so that two different styles are to be applied (even if the initial or default print rendering may be the same), a search and replace with editorial review is required.

To the extent possible, all operations that express or implement the underlying data structure -- such as the naming and placement of jump destinations, initial application of paragraph styles, the fielding of particular data -- are accomplished working with the flat file. In between, major steps we compile the flat file into an infobase for review and adjustment -- a way of catching anomalous paragraph styles or improper level assignments, for example.

In addition, we employ the Views workspace for certain kinds of link creation and for refining or correcting the links created by more automated search and replace methods. For example, our standard code format includes linking words and phrases defined in the code to their definitions. Our process for building these links begins with an editorial step -- the identification of all instances of term or phrase definition. Those definitions are each given both a jump destination name derived from the term and a “defined term” character style, within Views. That is followed by a second step that consists of a search within Views for all occurrences of each defined term. A jump link of the appropriate style from the first is hand-crafted from the first occurrence. That first jump-linked term can, thereafter, be pasted as a replacement for subsequent ones.

Critical Elements of Infobase Design

Format and Function

Effective infobase design calls for close attention to the interplay between two “F”s -- format and function. A work’s format, how it presents, how it communicates with the user is as used here supportive of but separate from the discrete “hyper” elements, the functions users can perform with it.

Several important questions to consider in developing the work’s format have to do with differences between electronic media and print. The screen offers a smaller window on a large text than the typical page, but offsetting that challenge or disadvantage is the capacity to use color, graphics, even sound. The distinctive additional functionality presented by the medium poses fresh format issues. How should the

option to follow a hypertext link from a section to related commentary be presented -- with what combination of text, color, font, graphic element? These and related format questions are significantly influenced by whether printing (or extracting for later printing) of portions of the work is an important contemplated use. Format can either encourage or discourage moving material out of the infobase. If key references are embedded in links that do not have explicit print representation they are lost upon export to print. And if hypertext links are represented in ways that carry over to print, even as the links themselves do not, printing is discouraged.

The basic hypertext functions to anticipate include, at a minimum:

- How is the user pursuing a particular issue or question or problem to be guided to the most valuable starting point? Options available to the infobase designer/builder include searches (including searches focused by one or more query templates), the dynamic table of contents derived by Folio Views from the data structure, alternative tables that map the infobase, graphic representation including flow charts or diagrams.
- How is the user provided with context for a particular piece of text, including the path he or she may have followed in reaching it? Options include the built in reference window view on the infobase structure, but also diverse links and search possibilities suggested to the user. Every section can carry its own map of adjacencies -- a tailor-made table of related other material ranging from definitions, through commentary, to other sections that are closely connected.

Data Structure

Records (Fighting the Tendency to Render Every Carriage Return a New Record Boundary)

Achieving the right granularity through setting record divisions appropriately is the first issue of data structure we address. The more important searching the infobase (as distinguished from following links) will be for users, the more important it is to get the material divided into suitable records. The greatest risk is of records that are too small, or more accurately bounded at illogical points in the text -- points where a shift in display may be called for, even as the flow of a single idea continues.

Text that is imported from word processor files and text entered directly in Folio Views by the unselfconscious typist will commonly be divided into suboptimal records, having record divisions where simple paragraph or line breaks belong.

The two factors that inform the judgment about the outer bound for record size relate to ultimate search behaviors. Records are too big if they lump multiple topics in text that extends over several screens of display. For if they do: 1) the user's searches are less effective and 2) record-based annotation, especially notes, grouping, and tagging lose precision.

Levels

Levels have major impact on both text search and table of contents functions. The LII standard for level setting is to push levels down to the lowest point in the document structure where logical units (or a significant number of them) carry headings. If a statute for example is divided into parts, parts into sections and lettered subsections carrying headings we will level down to and including subsections. If, however, a code has numbered or lettered subdivisions of rules or sections but they do not carry headings we stop at the rule or section level. The premise that underlies this practice is that with this type of legal material the principal gain achieved by leveling below the section is its effect on the dynamic table of contents. An expansion of a section 203 into a list of a, b, c, d, and e doesn't offer much to the user unless those subdivisions provide some clue as to content (*i.e.*, carry a heading).

The LII standard form has a generic list of levels with the topmost one representing the full code or act and level #4 corresponding to section or rule. Lying in between are enough intermediate levels for parts, chapters, subchapters, and the like -- given the modest scale of our present publications. Levels below section, rule, or the equivalent are available but are not used beyond the point noted above.

- U.S. Civil Rights Statutes - Title 42 USC
- CHAPTER 21 - CIVIL RIGHTS
- + SUBCHAPTER I - GENERALLY
- SUBCHAPTER I-A - INSTITUTIONALIZED PERSONS
- § 1997. Definitions
- § 1997a. Initiation of civil actions
- (a) Discretionary authority of Attorney General; preconditions
- (b) Discretionary award of attorney fees
- (c) Attorney General to personally sign complaint

Illustration 1. Levels Implemented in Civil Rights Statutes of the U.S. (LII)

Once applied, the LII's generic levels are renamed to correspond to the nomenclature of the particular body of material. That is, level #4 is renamed section or rule, level #2 chapter or subchapter and so on so that these labels carry meaning for the ultimate user.

Subsection Times New Roman 12 B I U T

CHAPTER 21 - CIVIL RIGHTS
 SUBCHAPTER I-A - INSTITUTIONALIZED PERSONS
 § 1997a. Initiation of civil actions
 (a) Discretionary authority of Attorney General; preconditions

either House of Congress is in session.

§ 1997a. Initiation of civil actions

(a) Discretionary authority of Attorney General; preconditions

Whenever the Attorney General has reasonable cause to believe that any State or political subdivision of a State, official, employee, or agent

Illustration 2. Subchapter, Section and Subsection Levels in Civil Rights Statutes of the U.S. (LII)
Named According to Act's Own Nomenclature

Jump Destinations (and Their Naming)

LII titles are designed to be hospitable targets for the links of others (teachers, students, other users). One direct consequence of that aim is that jump destinations are placed regularly through the document, without regard to whether these spots are in fact linked to from within the document or from any companion document published by the LII. Jump destinations are consistently located at the section level or below at whatever level of granularity users are likely to want to cite (*i.e.*, link) to the document. A second consequence of this aim is that jump destinations are given names that are derived from the organizational or addressing scheme of the document in its original or benchmark form. The goal is to allow someone creating a link to § 203(d)(4) of a particular statute to be able to do so straight from the jump destination list without having to deal with arbitrary or alien destination names.

(g)(1) Civil remedies

Whenever any agency --

(A) makes a determination under subsection (d)(3) of this

section

individ

to mak

(B) ref

subsec

(C) fail

with su

comple

determ

rights,

may be

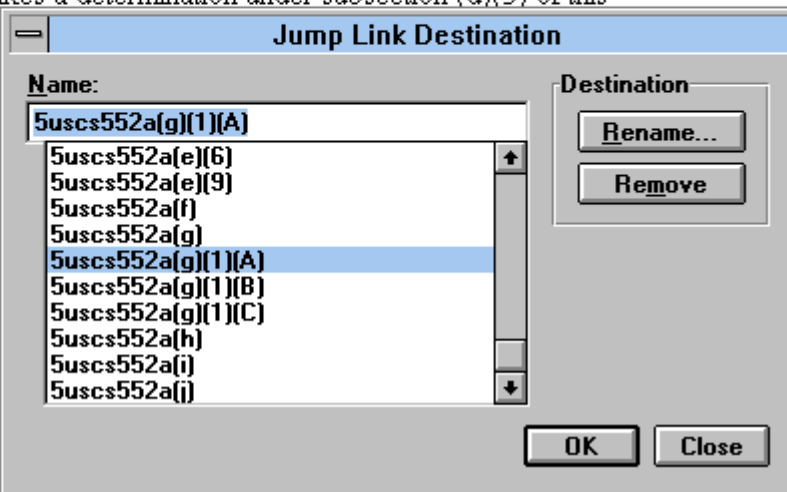


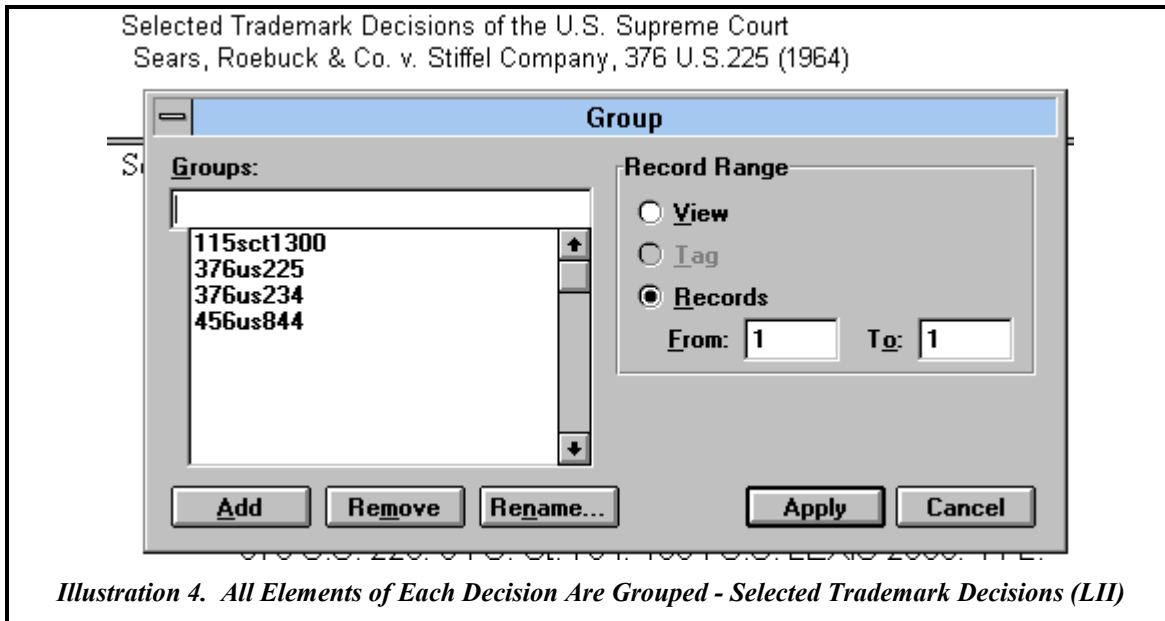
Illustration 3. Jump Destinations in U.S. Administrative Procedure Act (LII)

This practice places additional burden on the infobase creation process because the source files for most codes (coming from a print environment) do not carry a full citation string at each spot where this design calls for a jump destination. Typically, at the beginning of the text portion that is to be addressed as “203(d)(4)” the source text will simply have a “(4)”. (In print readers draw the rest from format and sequence.) The infobase creation process needs, therefore, to be sufficiently sophisticated to build full citation names for each citable point from the implicit information carried by successive indents and other format cues.

Groups

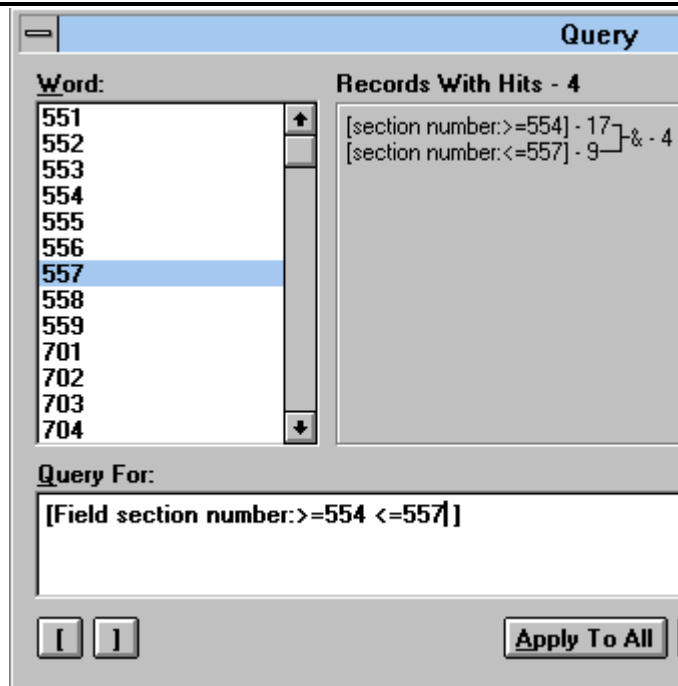
LII codes and statutes make minimal use of Groups. By contrast, LII’s standard format for cases includes grouping all elements of a case as a group. The difference between the two is rooted in search and printing considerations. A collection of cases is a collection of records in which the last record of one decision has a very strong relation to those immediately preceding it, but little or no logical relation to that immediately following. A user wanting to print or save some material is almost certain to think of each decision as a document and to want to work with it as a unit (or perhaps some smaller part). By contrast a researcher working with an intricate statute may want to collect multiple sections in sequence plus scattered other fragments in patterns very difficult to anticipate in advance. By putting all portions of a decision into a

single group and then linking the top level record of that case to the group the document becomes readily separable from its adjacencies. For the comparable self-linking function of a statutory section the LII style employs a jump link rather than a query link to a section group, precisely so that the section is not cut off from its neighbors.



Fields

LII practice is to use fields (and associated query links) for but a single function in code publications. It is not uncommon for a code section to contain an explicit cross reference to a range of other sections -- stated as a range, rather than an explicit list. When a statute section says "except as provided in § 203," the LII uses a jump link to implement the cross reference. When one says "except as provided in §§ 203 and 204," we use two jump links. When, however, a section says "except as provided in §§ 203 - 210," we implement the cross reference using a query link that depends on the section numbers being held in a numeric field that can be hit by a search for sections with numbers ≥ 203 and ≤ 210 . Handling the reference this way achieves, first, economy in creation and maintenance. The infobase builder can implement the cross reference without creating an explicit list of the sections in the range. (There may be no 204 but a 205a and a 205b. And if a new section is later added within the range, the link continues to work accurately.) The product of this field-based query link, a list of section headings contained within the range, also has distinct advantages for the user (over alternative ways of implementing a one-to-several cross reference). Each heading on the list can be clicked on to enter the statute at that point. (All records at the section level are self-linked.)



*Illustration 5. Search on Section Number Field of Administrative Procedure Act (LII)
Equivalent to a Multiple Section Reference to §§ 554 - 557*

§ 554. Adjudications

§ 555. Ancillary matters

§ 556. Hearings; presiding employees; powers and duties; burden of proof; evidence; record as basis of decision

§ 557. Initial decisions; conclusiveness; review by agency; submissions by parties; contents of decisions; record

Illustration 6. List of Sections Produced by a Field-Based Query Link

Links

Folio Views casts infobase links into three types (jump links, popup links, and query links). For many purposes, however, it is useful to break out of that typology and divide links according to whether they take the user: 1) from one point to another single point, 2) from one point to multiple predefined points (such as the cross-reference link to sections 107 through 110), or 3) along a pre-formulated search that can draw in new relevant material as the content of the queried infobase expands. The LII's practice is to employ link styles that reflect the type of material to which they lead rather than the FOLIO methodology (jump, popup, query) they use. For example, a link from one section to another and a link from one section to three others will be displayed in the same "reference" link style even though the first will be implemented through a jump link and the second through a query link.

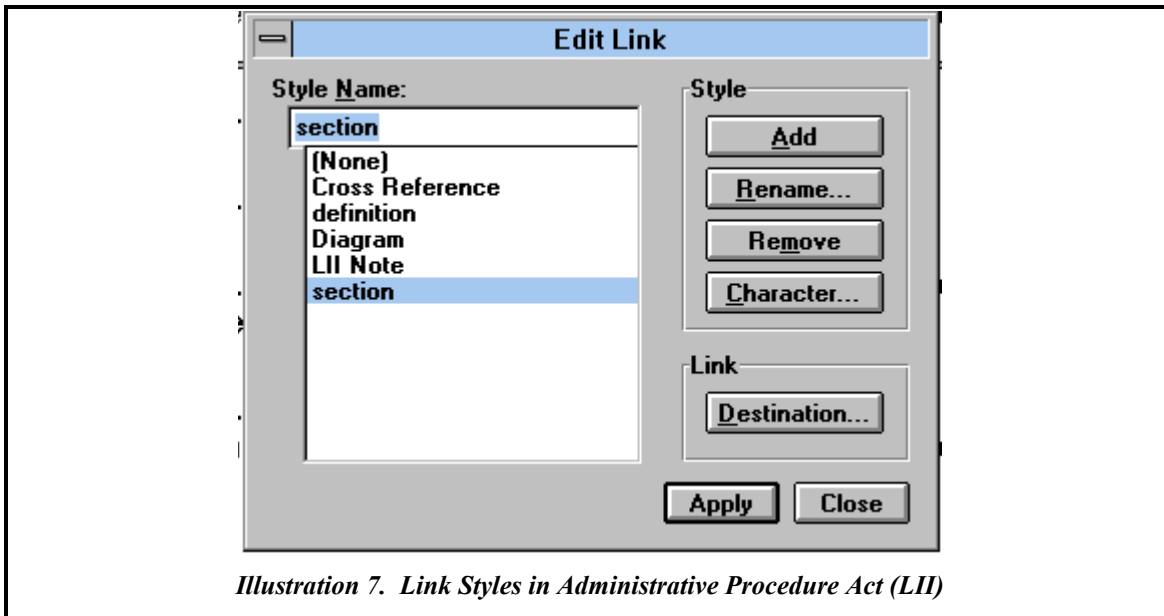


Illustration 7. Link Styles in Administrative Procedure Act (LII)

Other Specific Design Considerations

Role of Print in Use of Infobase

There is a natural tendency in creating hypertext to focus on screen presentation and hypertext function to the exclusion of print. However, many if not most users will want, on occasion, to print some portion of the infobase. In preparing hypertext codes, the LII has aimed at a format that will create attractive print versions of the full work. Since we imagine that at least some users will want to work some of the time from printed text we aim to encourage the use of the infobase itself as a cost-effective source of print (on-demand). With case collections the likelihood that users will want, on occasion, to print particular cases leads directly to the creation of a single case group with associated query-link so that the records that comprise the case can be easily isolated as a single printable view.

Some direct consequences of designing a “print friendly” work include: 1) use of link styles that do not carry through to print (no underlining or italics for link launch text), 2) double storing of code section annotations that appear in pop-up links as hidden text that will print with the code section rather than at the end of the document, and 3) creation of footers that indicate the source of the printed version. Nonetheless, some design choices made to enhance the work’s performance as hypertext will inescapably reduce its attractiveness as a print source or, at least, run against current print conventions. Most significant of these in the current LII treatment of codes is the separation of lengthy comments from the sections to which they relate. Such separation means that when an LII infobase is used to generate a print version of the UCC or Federal Rules of Civil Procedure, commentary and the code or rule text to which it relates are not adjacent.

[*Comment*]

§ 5-104. Formal Requirements; Signing.

(1) Except as otherwise required in subsection (1)(c) of Section 5-102 on scope, no particular form of phrasing is required for a credit. A credit must be in writing and signed by the issuer and a confirmation must be in writing and signed by the confirming bank. A modification of the terms of a credit or confirmation must be signed by the issuer or confirming bank.

(2) A telegram may be a sufficient signed writing if it identifies its sender by an authorized authentication. The authentication may be in code and the authorized naming of the issuer in an advice of credit is a sufficient signing.

[*Comment*]

§ 5-105. Consideration.

Illustration 8. Separation of Sections and Comments in UCC, Title 5 (LII)

Need to Conform to Some Benchmark (Print) Version

None of the LII publications in general release is based on a pre-existing set of paginated print materials with which they must be used in parallel. However, the LII has created a full hypertext version of an existing print coursebook. The principal design element called for in such a project is building all the important print landmarks into the hypertext version, including page numbers. When the teacher makes a day's assignment or calls attention to a particular passage using page numbers (e.g., "For Monday, pp. 250-272" or "What do you make of the second paragraph on page 123?"), the work should be set up to allow the infobase user to move with the same assurance and precision to the designated points as the printed book user. In LII practice this is accomplished by marking all page breaks with the page number (using either a character style alone or a field with an associated style that allows the numbers to be suppressed as hidden text or brought forward) and with a jump destination or group name that allows the user to go directly to the material on a cited page. Setting them up as jump destinations (with names derived from the page beginning at that point) allows the simplest user path to the point.

Platforms (DOS Reduces the Common Denominator)

Any infobase that is dependent on the use of graphic material including graphic icons will not run with full value under the DOS version of Views. And unlike HTML, Views does not provide for a text alternative to

Property Statutes infobase. The important Supreme Court decisions in the intellectual property field are held in three separate infobases divided along the same lines. Similarly, our publication of the U.C.C. divides the code into a separate infobase for each article.

We have followed this approach for several reasons. First, it allows us to offer an array of supporting materials for different courses more readily -- a collection of copyright materials (including the Copyright Act) for a copyright course as well as a collection supporting courses that cover patent, copyright, and trademark. Second, it has allowed us to distribute large works on diskette (the UCC infobases require three of them). Third, it allows users with limited hard disk space to load the material they need when they need it.

Applying These Principles to Particular Document Types

Codes (with Commentary)

Illustrative examples -- Civil Rights Statutes of the U.S.; Administrative Procedure Act of the U.S.; UCC, Title 5

Collections of Cases

Illustrative example -- Important Trademark Decisions of the U.S. Supreme Court

Some Larger Considerations

Focusing on the Distinct Needs of the User (Students Have Different Needs of and Use Patterns with Statutes and Cases Than Practitioners)

Good authors working in print devote a lot of time to thinking about, and in a revision cycle listening to, their readers. Given the greater capacity and capability possible with a hypertext work, attention to the user holds even greater importance. Since the work may present itself as both wider (more comprehensive) and deeper (more thorough) than print works, users will be less tolerant than book readers of failures to meet their reasonable expectations of coverage. Since the work can provide major levels of functionality not possible with print, failures to realize the full potential carry high stakes. All of this is the more challenging because of the flexibility of the medium. Even a legal infobase of modest size will present users with so many different possible use patterns and access paths that envisioning precisely how it will be used can be daunting.

Open Architecture (An Invitation to Links)

Creating works that we want law students and faculty to work with, indeed, to build upon, we have sought to publish them “ready to receive links.” Ideally, the student or faculty member wanting to link to a logical unit of a code or statute will find an appropriately named jump destination or group (or both) ready and waiting for the link. “Appropriately named” means simply that the user wanting to link to a portion of the work known by a standard designation (e.g., section 210(c)) should be able to create the link by selecting it from the list of pre-existing jump destinations without having to go to text chunk itself to determine whether it has a jump destination and, if so, what arbitrary name it was assigned. The techniques described elsewhere that enable links to several units that will display as a list (via a query link) should be available to users -- which means that the structure should be documented and also that it should be applied consistently through the work and not limited just to portions that are the subject of existing links of that sort. (See the prior discussion of Jump Destinations (and Their Naming).)

Institutional Factors

Security

All LII infobases are set up with guests being limited to the creation of shadow files, to printing, and to “save as” functions. Our aim has been to have a set of security options that work well for the law school putting our infobases in a school-wide network but also fit the needs of an individual student with the infobase on her hard drive. For the network situation we’ve set up shadow file rights to include everything except changing the infobase information, headers or footers. Being comfortable with the individual licensee placing annotation directly in the infobase, we have set up a password protected user name (“licensee”) with the same broad rights as a shadow file receives. Individual licensees are informed of the original licensee password and advised to change it.

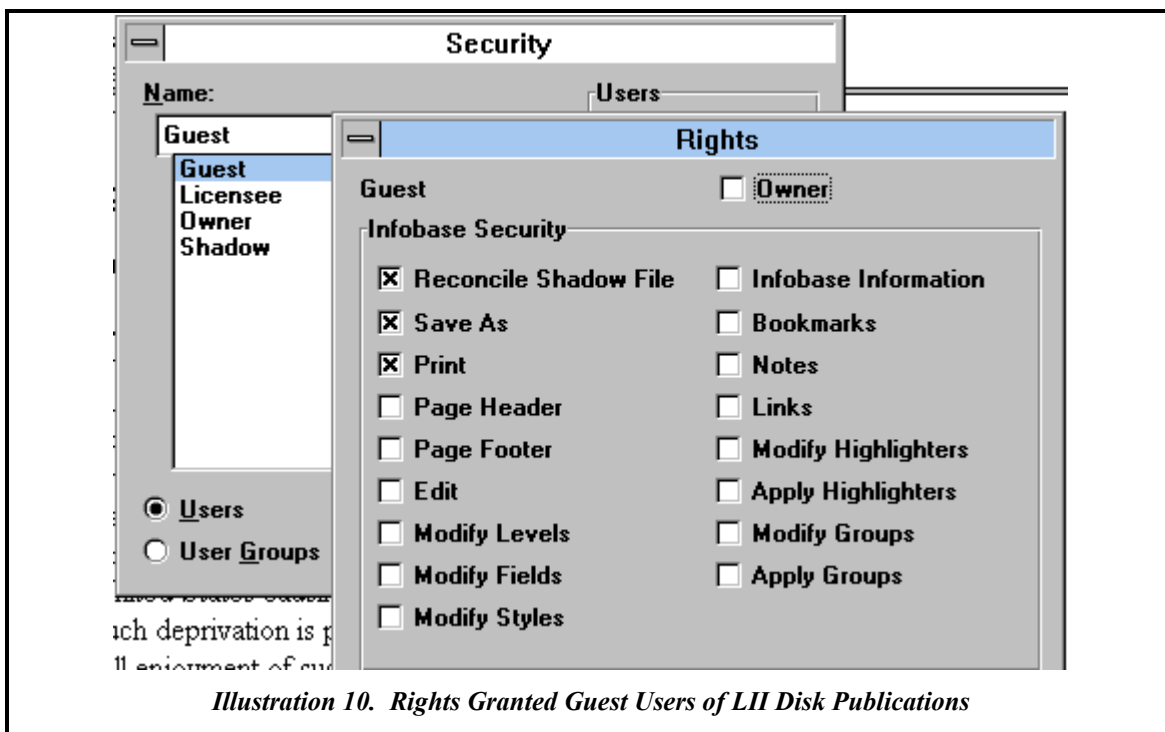


Illustration 10. Rights Granted Guest Users of LII Disk Publications

Setup

The most effective way to offer a broad collection of infobase material at a public workstation is through a small infobase menu that contains query links to each of infobase titles that make up the collection. The LII prepares such a menu each year and offers it to institutional licensees. For the menu to work from the start for users who may not be experienced Views users, it is important that the network setup be arranged so that the gname.ini or lname.ini file (as appropriate) is properly created in advance so that menu choices don't drop a puzzled user into an unfamiliar file structure in search of an unknown file name.

Production (Building the Full-Scale Work)

Prototype

(The importance and nature of this step is discussed previously.)

Import

The challenge in importing text whether from a word processor file format, relatively flat ascii, or some form of tagged ascii (as for example HTML or data from GPO sources) is to capture all the structural information it contains -- either as straightforward logical unit styles or more commonly encoded in a set of format conventions (*e.g.*, each section begins with a double return, followed by two spaces and the section

symbol). If the source is other than ascii or is a form of ascii for which Views has a special filter (e.g., a LEXIS/NEXIS download), the first importing step will usually be to bring the document into Views through the appropriate filter and then save it as a flat file for analysis of the conversion requirements. Some of the analysis may be possible (and indeed some of the conversion work may be done) in the relevant word processing environment. A key question for example is whether the headings introducing key units of the work (articles and sections, say) were simply given font assignments on an ad hoc basis or whether a set of styles was consistently applied.

Strip and Convert

The LII's process aims to remove all or nearly all formatting achieved in basic units of text by arbitrary tabs, hard returns and the like, relying instead on a set of level and paragraph styles for nearly all such format. This requires a conversion process that keys on all format elements that signal different logical units or styles and substitutes the corresponding flat file structural markup for the combination of font and format cues used in the original. After all structure that can be derived from the original text format has been translated into level and paragraph style designations, all character and font codes that will be preserved in the infobase are converted into a suitably named character or paragraph style. Thus rather than designate a pair of words as bold, the LII practice is to create one or more character styles (that may very well be rendered in bold). If for example one use of bold will be to highlight words that are being defined in the particular passage the document will have a character style denominated "defined term."

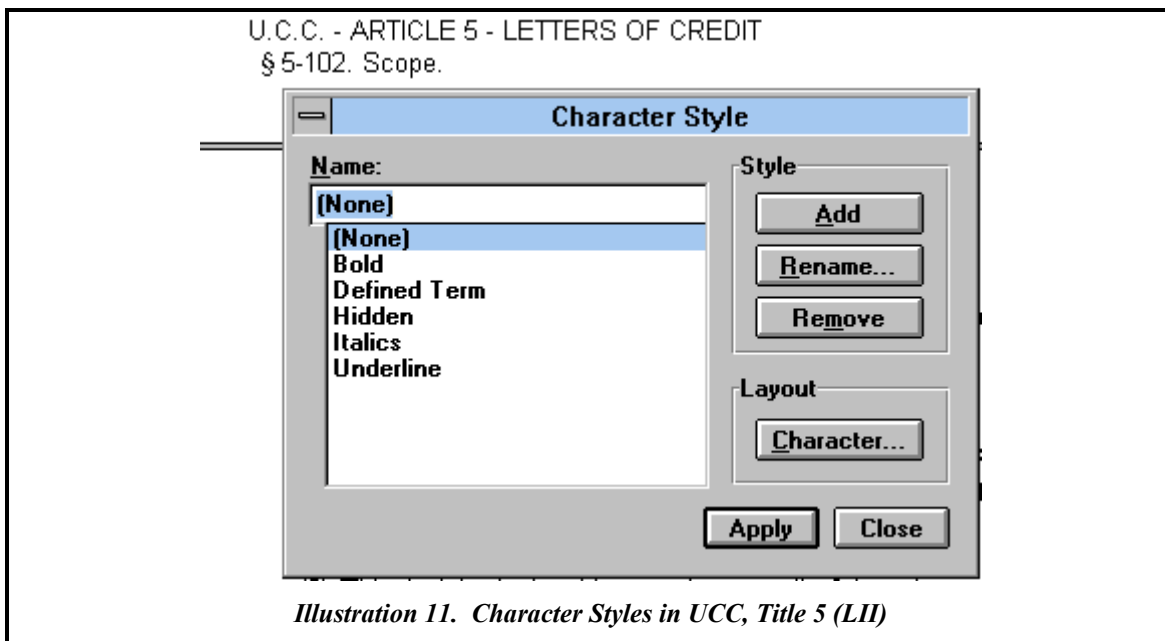


Illustration 11. Character Styles in UCC, Title 5 (LII)

A second important element of this stage of conversion is separating out returns that should be made the beginning of new records, those that should be made the beginning of new paragraphs within a continuing record, and those that should simply begin a new line.

Automating Jump Destination, Group and Field Creation

With a well structured source document, levels; jump destination names for sections, subsections and so on; the fielding of section numbers; and grouping can largely be automated with a regular expression script (FSR or PERL), an editor with macros, or some combination of the two. Typically, the most difficult step of the process to design is the creation of full jump destination names for units that do not carry their full identity in print. (See the preceding discussion of Jump Destinations (and Their Naming).)

Automating Explicit Cross-Reference Linking

In some works, explicit cross references employ a readily parsed format and terminology (the word section or sections followed by a character string of specific characteristics, for example). Where that is the case and where the cross references are all internal to the work or those that are can be parsed separately from those that aren't, a regular expression script (FSR or PERL) or editor with macros is used to create "rough draft" cross reference links. This is done working directly with the flat file. Before compiling the flat file as an infobase the "rough draft" links are all inspected (using an editor search) to catch anomalies. For example, if linking is being carried out to the subsection level and the text in a particular case reads "subsections c and e of section 423," the automated process will have created a single link to section 423 while a double link (to the two subsections) may be desirable.

Editorial Linking from Within Folio Views

While automated "rough draft" linking is to be preferred. Cross reference links made from within Views are sometimes the most effective route. This will be true, for example, of works with a low ratio of internal cross references to references to material not to be included in the infobase. In building cross reference links from within Views the LII's practice is to rely on two distinctive strengths of the Views environment. First, a search is run for all occurrences of the word or words associated with such references (*e.g.*, section or sections). As all hits are reviewed links are created as appropriate but once one link to a section has been created it can then be used in a copy and paste operation for all other links to the same section. Working from the top down in the infobase all potential link launch points identified by the search are reviewed and linked with two variants possible. The first simply uses copy and paste when recurrent links to the same section occur within reasonable proximity to one another (often the case with statutes). The second

involves creating a link scratch pad at the end of the infobase where links are pasted in sequence (and accessed through a second window) so that they can be used for the copy and paste operation.

Building links between defined terms and their definitions involves the same technique options. A set of defined terms must first be constructed and all definitions assigned suitable jump destination names. Then a “rough draft” set of links can be built using a regular expression script or editor and macros working with the flat file (with the need to attend to such variations as initial capitalization, plural and possessive instances of the word or phrase). Or the links can be created working with the Views environment using a search for each of the words and the copy and paste technique described above. Because all “rough draft” links will need to be reviewed more closely with definition links than with explicit section cross references our experience has been that most infobase editors building such links prefer to work completely within the Views environment.

There are two reasons why greater editorial scrutiny is required with this type of link. First, especially when codes are accompanied by commentary and the commentary is also being linked to the definitions, there are, almost inevitably, instances of a defined word being used in a different sense than the definition speaks to (with that being quite clear from context). (The statute defines “act” as a particular set of statutory provisions and the word is being used in the passage in question in the sense of something done.) Viewing these links as calling for informed editorial judgment, the LII’s process becomes labor intensive at this point. A second and quite different consideration requires close editorial review of definition links -- namely, an esthetic one. Explicit section cross references are not used recurrently throughout a paragraph, while a single defined term may well be. For example, if the word “bank” or “copy” or “lender” is at the center of a section’s treatment of an issue that word may recur many times within it. No matter how links of this sort are signaled, if every occurrence of the defined word in such a paragraph is linked the result produces a high order of visual distraction. The LII’s practice is to link only the first occurrence of a defined term in a paragraph unless the paragraph extends over more than a screen. In the latter case, our aim is to link roughly one occurrence per screen.

§ 5-109. Issuer's Obligation to Its Customer.

(1) An issuer's obligation to its customer includes good faith and observance of any general banking usage but unless otherwise agreed does not include liability or responsibility

(a) for performance of the underlying contract for sale or other transaction between the customer and the beneficiary; or

(b) for any act or omission of any person other than itself or its own branch or for loss or destruction of a draft, demand or document in transit or in the possession of others; or

(c) based on knowledge or lack of knowledge of any usage of any particular trade.

Illustration 12. Defined Term Links in UCC, Title 5 (LII)

Infobase Maintenance

An infobase aimed at legal education or lawyers must be maintained. Indeed, one of the attractions of electronic media is ease of updates. That ease calls for careful management. The LII has, to date, updated its materials on an annual cycle, releasing new editions for the beginning of each academic year. The basic elements of the update process are straightforward. It begins with retention of an identified archive copy of the prior edition. All amendments to the document(s) occurring since that prior edition are collected in digital form by a student editor. Working from a copy of the archival infobase file, a student editor identifies all sections that have been changed. The changes are entered directly in the infobase and high-lighted. Every new or altered character is high-lighted. The nature of the change is entered by the student in a note to each amended record. The changes are then reviewed by an LII faculty editor. In some cases the high-lighter may be retained to allow users to pull up the changes (with the style changed so that it does not display unless changed to do so by a user). More commonly the high-lighting is removed as the changes are verified.

Porting to Other Platforms

The LII has ported many of its infobase publications to HTML. That process is documented in a separate process paper, available from the LII in print and directly via the Internet. In a sense the process is generic for since the Folio VIEWS flat file contains full representation of all structure and all links in the infobase in an SGML-like form, it is possible to port an infobase to any other hypertext environment that allows importation, compilation, or use of an SGML-like source file. Currently, nearly all hypertext platforms include that capability.

