Table of Contents

List of Figures

1. Introduction to Symbolics Concordia Book Design

In Symbolics Concordia, you can:

- *Control* the appearance of a book by using the Symbolics Concordia Markup Language. The markup language is a set of predefined specifications for formatting environments and commands.

- *Modify* the appearance of a book by modifying these specifications. This book describes how to modify these specifications.

This section assumes you are familiar with the Symbolics Concordia Markup Language. If you are not, see the section "Controlling Your Document's Appearance" in *Symbolics Concordia*.

1.1. What is Book Design?

Simply stated, *book design* is what determines the appearance of a book. Book design utilizes typography and page layout to produce two-dimensional compositions which are pleasing to the eye, and appropriate to the subject matter and audience. You could say that book design *is* the appearance of a book.

The book design capabilities of Symbolics Concordia are very strong but, at this time, the interface to them is a Lisp-like language. See the section "How the Book Design Sources Are Organized", page 7. Some of the things described in this document are going to seem like nuts-and-bolts internals. This is because they are.

1.2. Book Design Vocabulary

Discussions about book design can be confusing because too often terms are used interchangeably or colloquially. The following is a glossary of book design terms as used in Symbolics Concordia.

Book (Document) A registered record. That is, a record for which a **sage::register-book** form has been evaluated. To see how to register a record, see the function **sage::register-book**, page 52.

Book Design A set of aesthetic decisions which affect the appearance of a book or set of books.

Specification In a general sense, a description of how something in a book should look. A specification consists of a set of definitions which control the appearance of a *book design element*. In Symbolics Concordia, specifications live in the *book design sources* as clauses in **sage:define-book-design** forms.

Top-level Specification

Specifications in top-level style or text clauses in *top-level book design elements*. Top-level style specifications define the block in which type is formatted, that is the margins, and consists of attributes such as paper-width, leftmargin, rightmargin, topmargin, and bottommargin. Top-level text specifications control the appearance of running text not inside some environment and consists of attributes such as Font, LineWidth, Spacing, Spread, FaceCode, Indent, Justification, and Hyphenation.

Book Design Element

Any aspect of a book which can be modified to change the appearance of the book. Examples of book design elements are footnotes, chapter headings, pageheadings, enumerated lists, and the like.

In Symbolics Concordia, a book design element is a collection of one or more *specifications*. Book design elements provide a way to organize all the many specifications which make up the appearance of your documentation. See the section "Book Design Elements", page 11.

Top-level Book Design Element

A book design element whose name includes the name of a *document device type* is a top-level book design element. Such a book design element includes the top-level specifications for that particular document device type. For instance, **sage::letter-lgp2** contains the top-level specifications for the Letter LGP2 document device type.

Document Type

A collection of *book design elements* for one particular type of book but any *device type*. "Reference", "Letter", and "Article" are document types. See the section "Document Device Types", page 8.

Device Type

A device on which formatted documentation is output. Currently, Symbolics Concordia supports the device types Generic, Screen, LGP2, and Dex. The Generic device type does not actually represent an output device supported by Symbolics Concordia but instead a universal device type used by the other two. For more explanation of the other device types, see the section "Document Device Types", page 8.

Document Device Type

A collection of *book design elements* which together specify the appearance of formatted documentation. A document device type is made up of a *document type* and *device type*. "Reference LGP2", "Generic Screen", and "Lisp-dictionary LGP2" are document device types. See the section "Document Device Types", page 8.

Book Design Sources
A set of Lisp code files which contain all the book design specifications for Symbolics Concordia. See the section "How the Book Design Sources Are Organized", page 7.

Here are some additional terms used in Symbolics Book Design:

Collectors
Specifications for the table of contents, table of figures, table of tables, and index. Collectors collect entries as the book is formatted. For instance, the contents collector collects section headings and their page numbers as the book is being formatted. Then, when it is time to format the table of contents, the contents collector supplies the necessary information. See the function **sage:define-book-design**, page 45.

Command
A directive to the formatter. See the section "Markup Commands and How to Use Them" in *Symbolics Concordia*.

Counter
A specification which defines how a chapter, section, or other section heading gets formatted. See the function **sage:define-book-design**, page 45.

Environment
A collection of directives to the formatter that are bundled together (usually with some high level intent) that take effect over a region (defined by the environment markup delimiters) of a document. See the section "Environments and How to Use Them" in *Symbolics Concordia*.

1.3. Inheritance Dependency

Book design elements can inherit from other book design elements and environments can inherit from other environments. Children inherit from their parents and grandparents. Therefore, modifying something about a parent might affect its children and its grandchildren. You must consider these *inheritance dependencies* carefully when deciding where to make modifications to book design elements and environments. The "Book Design Browser" provides tools for researching these inheritance dependencies. See the section "Book Design Browser", page 15. See the section "Modifying a Document Device Type", page 19.

In the book design sources, a "use" clause introduces a parent book design element or environment. The book design element or environment which includes a "use" clause is the child. For example, consider the following book design element:

```
(define-book-design child-book-design ()
 (use parent-book-design)
 (define
  (child-envr (use parent-envr))))
```

Several "Book Design Browser" commands provide you with information about inheritance dependencies.

Describe Document Device Type
: Lists the given document device type's top-level book design element and that element's parents.

Describe Book Design Element
: Lists the given book design element's parents.

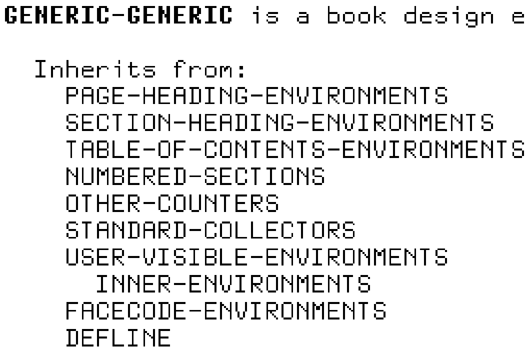
Show Dependencies on Book Design Element
: Lists the given book design element's children.

Describe Environment
: Displays a table of the given environment's parent environments and its parent book design elements and lists the given environment's children.

Show Dependencies on Environment
: Lists the given environment's children.

In the displays produced by these commands:

- "Inherits from" or "is specified by" introduces a list of *parents, grandparents*, and so on. In this example, **sage::generic-generic** (child) inherits from **sage::page-heading-environments** (parent), and so on. Note that indented elements are next-generation parents. So, in this example, **sage::user-visible-environments** (child) inherits from **sage::inner-environments** (parent).

  ```
  GENERIC-GENERIC is a book design element.

    Inherits from:
      PAGE-HEADING-ENVIRONMENTS
      SECTION-HEADING-ENVIRONMENTS
      TABLE-OF-CONTENTS-ENVIRONMENTS
      NUMBERED-SECTIONS
      OTHER-COUNTERS
      STANDARD-COLLECTORS
      USER-VISIBLE-ENVIRONMENTS
        INNER-ENVIRONMENTS
      FACECODE-ENVIRONMENTS
      DEFLINE
  ```

- "Is inherited by" introduces a list of *children*.

  ° For a book design element, "is inherited by" introduces a list of *children*, *grandchildren*, and so on. In this example, **sage::inner-environments** (parent) is inherited by by **sage::user-visible-environments** (child), and so on. Note that indented elements are next generation children. So, in this example, **sage::user-visible-environments** (parent) is inherited by by **sage::generic-generic** (child).

```
INNER-ENVIRONMENTS (a book design element) is inherited by:

    USER-VISIBLE-ENVIRONMENTS
        GENERIC-GENERIC
            GENERIC-SCREEN
                GENERIC-DEX
                    GENERIC-DEX-BACKGROUND
                MEMO-SCREEN
                LETTER-SCREEN
                GENERIC-MAC-SCREEN
            GENERIC-PAPER
                GENERIC-LGP2
                    3SYMANUAL-LGP2
                        LISP-DICTIONARY-LGP2
                        ARTICLE-LGP2
                        DOC-EX-LGP2
                        MAC-DOC-LGP2
                    APPROACH-LGP2
                    MEMO-LGP2
                    LETTER-LGP2
                    INSTALLATION-LGP2
                GENERIC-DMP1
            MEMO-GENERIC
                MEMO-LGP2
                MEMO-SCREEN
            LETTER-GENERIC
                LETTER-SCREEN
                LETTER-LGP2
            ARTICLE-GENERIC
```

° For an environment, "is inherited by" introduces a list of *children*.

```
Description (an environment) is inherited by:

Environment  In Book Design Element
Transparent  INSTALLATION-ENVR-MODS
Descript     APPROACH-ENVR-MODS
Description2 TOOLS-AND-TECHNIQUES-ENVIRONMENTS
Descript     USER-VISIBLE-ENVIRONMENTS
```

1.4. Modifying the Appearance of Your Documentation

You can modify the appearance of your documentation at any of three levels. You must first decide which level is appropriate.

To modify the appearance of:

- All books of a particular document device type, you modify the book design sources for that particular document device type. See the section "How the Book Design Sources Are Organized", page 7. See the section "Modifying a Document Device Type", page 19..

- A particular book, use the Symbolics Concordia command "Modify" in the top-level record of that book. See the section "Modifying a Single Book", page 31.

- A single instance of a markup environment, modify its attributes by clicking Right on one of its markup delimiters. See the section "Modifying a Single Instance of an Environment", page 37.

2. How the Book Design Sources Are Organized

When you want to modify a document device type, you must modify the book design sources for that document device type. The book design sources contain the design specifications which control the appearance of the various document device types. The sources are Lisp code files in the directory SYS:NSAGE;. The code used in these files is written in Lisp.

The "Book Design Browser" provides tools to help you investigate the contents of these files. It saves you from having to wade through the code in them and it assists you in identifying where to make modifications. See the section "Book Design Browser", page 15.

See the section "Modifying a Document Device Type", page 19 for examples of how to use the "Book Design Browser" to make modifications to the book design sources.

The sources include definitions for document types, book design elements, commands, environments, attributes, counters, fonts, and much more. Mostly what you will be concerned with are "Document Device Types" and "Book Design Elements".

2.1. Book Design Source Files

The following is a listing of the relevant files in the SYS:NSAGE; directory with a description of the contents of each.

BD-APPROACH.LISP The book design specifications for the Approach LGP2 document device type.

BD-ARTICLE.LISP The book design specifications for the Article Generic and Article LGP2 document device types. This file also defines several environments used just by these document device types.

BD-DICTIONARY.LISP The book design specifications for the Lisp-Dictionary LGP2 document device type. This file also defines several Lisp functions and macros used to format Lisp dictionaries.

BD-GENERIC.LISP The top-level book design elements for the various Generic document types. The book design specifications for these document types are defined in database.lisp.

BD-INSTALLATION.LISP The book design specifications for the Installation LGP2 document device type.

BD-LETTER.LISP The book design specifications for the Letter Generic, Letter Screen, and Letter LGP2 document device types. Various commands and environments used only by the Letter document types are defined in bd-letter-commands.lisp.

BD-LETTER-COMMANDS.LISP
Various commands and environments used only by the Letter document types.

BD-MEMO.LISP
The book design specifications for the Memo Generic, Memo Screen, and Memo LGP2 document device types. Various commands and environments used only by the Memo document types are defined in BD-LETTER-COMMANDS.LISP.

BD-MEMO-COMMANDS.LISP
Various commands and environments used only by the Memo document types.

BD-REFERENCE-CARDS.LISP
The book design specifications for the Reference LGP2 document device type.

DATABASE.LISP
The book design specifications for the 3symanual LGP3 document device type. In addition, all the various environments, attributes, counters, collectors, commands, fonts, and boxes are defined here.

2.2. Document Device Types

A book is formatted according to the book design specifications of two components: a *document type* and a *device type*.

- A document type supplies the specifications for the kind of a document you are formatting, for instance a letter or reference cards.

- A device type supplies the characteristics for the ouptut device you intend to use, the screen or a Symbolics LGP2, for instance.

- A document device type is the combination of these two components, for instance Letter Screen or Reference LGP2.

Document device types are defined by the following Lisp functions and macros:

Device type	**sage::define-device-type**.
Document type	**sage::define-document-type**.
Document device type	**sage:define-book-design** and **sage:note-book-design-specifics**.

So, for instance, the following code defines the Reference Cards document type for the LGP2 output device.

```
(define-device-type 'lgp2)
```

```
(define-document-type ’reference-cards)

(define-book-design reference-cards-lgp2 ()
  (use generic-lgp2)
  (use approach-lgp2)
  (use reference-cards-counters)
  (use reference-cards-headings)
  (use reference-cards-envr-mods)
  (modify
    (box (BoxFlushRight Yes))
;;  (contentsenv (rightmargin "+2inches"))
    (bodystyle (spacing 11pts) (spread 2pts))
    (K (FACECODE K) TABEXPORT (HYPHENBREAK off))
    )
  (first
    ;;Style
    (envr (text
            (leftmargin "2.0inches")
            (RightMargin "2.0inches")
            (BottomMargin "3.0inches")
            (TopMargin "1.0inches")))
    ;;Top-level begin
    (envr (Text
            (Font SmallBodyFont)
            ;;(leftmargin "2.25inches")
            (LineWidth 26picas)
            (Spacing 11pts)
            (Spread 2pts)
            (FaceCode R)
            (Indent 0)
            (Justification on)
            (Hyphenation On)
            (Spaces Compact)))
    (init
      initialize-reference-cards-lgp2)
    ))

(note-book-design-specifics ’reference-cards :lgp2 ’reference-cards-lgp2)
```

When you format a particular book, you specify the document type by using the :document-type keyword to **sage::register-book**. If you want the book to be an article, for instance, then you would write the **sage::register-book** form as follows:

```
(sage::register-book "A Record Name"
                     :document-type ’sage::article)
```

When you format a particular book, you specify the device type as follows:

To use device type	*Format record using*
LGP2	Page Previewer command Format Pages Command Processor command Show Documentation :destination keyword Editor command `c-U m-X` Show Documentation Editor command `c-U s-P`
Screen	Command Processor command Show Documentation `m-X` Show Documentation `s-P`
Dex	Document Examiner command Show Documentation

The following matrix shows the document device type used for particular formatting conditions:

	registered book	*record from registered book*	*non-registered record*
Format Pages	*document-type* lgp2	*document-type* lgp2	generic lgp2
Show Documentation	generic screen	generic screen	generic screen
m-X Show Documentation	generic screen	generic screen	generic screen
s-P	generic screen	generic screen	generic screen
c-U Show Documentation	generic paper	generic paper	generic lgp2
c-U s-P	generic paper	generic paper	generic lgp2

Note: *document-type* means whatever value you've given for **:document-type** in the **sage::register-book** form for that book.

The following matrix describes the current set of document device types provided in Symbolics Concordia Book Design:

device type *document type*	generic	screen	lgp2
3symanual			3symanual lgp2
approach			approach lgp2
article	article generic		article lgp2
generic	generic generic		generic lgp2
installation			installation lgp2
letter	letter generic	letter screen	letter lgp2
memo	memo generic	memo screen	memo lgp2
lisp-dictionary			lisp-dictionary lgp2
reference-cards			reference-cards lgp2

2.3. Book Design Elements

A book design element is a collection of one or more specifications. In the book design sources, book design elements are defined using the function **sage:define-book-design**. See the function **sage:define-book-design**, page 45.

The name of a book design element indicates what kind of specifications it contains. For instance, **sage::section-heading-environments** is a book design element which contains the specifications for environments used in formatting section headings.

A book design element whose name includes the name of a document device type is a top-level book design element. Such a book design element includes the top-level specifications for that particular document device type. For instance, **sage::letter-lgp2** contains the top-level specifications for the Letter LGP2 document device type.

The command "Describe Book Design Element" displays information about a given book design element. The command indicates what sort of specifications the book design element includes. The command also lists any other book design elements inherited by the given element. For top-level book design elements, the display also includes the top-level specifications. See the section "Describe Book Design Element", page 39.

You can modify the appearance of a document device type by modifying the book design elements which specify that document device type. For more information on modifying book design elements, see the section "Modifying a Document Device Type", page 19.

The following is a listing of the various book design elements for the generic document type. Some of these, as noted, are simply placeholders and not inherited by any other book design elements. The others are simply collections of book design elements which are inherited by other book design elements. They serve to modularize the book design sources.

sage::generic-dex A placeholder. Not intended for use at this time.

sage::generic-dex-background
A placeholder. Not intended for use at this time.

sage::generic-dmp1 A placeholder. Not intended for use at this time.

sage::generic-generic
A collection of book design elements inherited by several other book design elements.

sage::generic-lgp2 A collection of book design elements inherited by several other book design elements.

sage::generic-mac-screen
A placeholder. Not intended for use at this time.

sage::generic-paper
A collection of book design elements inherited by several other book design elements.

sage::generic-screen
A collection of book design elements inherited by several other book design elements.

The following is a description of the currently defined *top-level* book design elements. These are the book design elements which contain top-level specifications for the various document device types.

sage::3symanual-lgp2
The top-level specifications for the 3symanual LGP2 document device type.

sage::approach-lgp2
The top-level specifications for the Approach LGP2 document device type.

sage::article-generic
The top-level specifications for the Article Generic document device type.

sage::article-lgp2 The top-level specifications for the Article LGP2 document device type.

sage::installation-lgp2
The top-level specifications for the Installation LGP2 document device type.

sage::letter-generic A top-level specification for the Letter Screen and Letter LGP2 document device types.

sage::letter-lgp2 Top-level specifications for the Letter LGP2 document device type.

sage::letter-screen Top-level specifications for the Letter Screen document device type.

sage::lisp-dictionary-lgp2
Top-level specifications for the Lisp Dictionary LGP2 document device type.

sage::memo-generic
Top-level specifications for the Memo document types.

sage::memo-lgp2 Top-level specifications for the Memo LGP2 document device type.

sage::memo-screen Top-level specifications for the Memo Screen document device type.

The following is a listing of all the rest of the book design elements.

sage::approach-counters
The specifications for section counters for the Approach document type.

sage::approach-envr-mods
Modifications to environments for the Approach document type.

sage::approach-headings
Modifications to environments used in section headings for the Approach document type.

sage::article-counters
The specifications for section counters for the Article document type.

sage::article-headings
Modifications to environments used in section headings for the Article document type.

sage::defline The specifications for environments used in the first line of a Lisp object record.

sage::dmp1-inner-styles
A placeholder. Not intended for use at this time.

sage::facecode-environments
The specifications for the facecode environments, for instance B (for boldfaced type) and I (for italicized type).

sage::inner-environments
Specifications for environments which are not intended for use in your documentation. Instead, these environments are inherited by other environments.

sage::lgp2-inner-styles
Specifications for environments used in several document types for the LGP2 device type.

sage::lisp-dictionary-counters
Specifications for counters used in the Lisp Dictionary document type.

sage::lm-inner-styles
A placeholder. Not intended for use at this time.

sage::numbered-sections
Specifications for the numbered section counters.

sage::other-counters
Specifications for FigureCounter, TableCounter, and FootNoteCounter counters.

sage::page-heading-environments
Specifications for environments used in pageheadings and pagefootings.

sage::screen-inner-styles
Specifications for environments used in various document types for the Screen device type.

sage::section-heading-environments
Specifications for environments used in section headings.

sage::standard-collectors
Collectors for the table of contents, table of figures, table of tables, and index.

sage::table-of-contents-environments
Specifications for environments used in the table of contents, table of figures, and table of tables.

sage::tools-and-techniques-environments
Specifications for some environments not currently used anywhere else.

sage::user-visible-environments
Specifications for environments available for general use.

3. Book Design Browser

3.1. Introduction to the Book Design Browser

The Symbolics Concordia Book Design Browser is a book design maintenance and development tool. For definitions of the terms used in the Browser, see the section "Book Design Vocabulary", page 1.

You can use the Browser to:

- Investigate the default definitions of Concordia environments for particular document device types.
- Change those default definitions.
- Develop new document device types for your site.

Select the Browser by clicking on the Browser Icon in the upper right-hand corner of the Symbolics Concordia window.

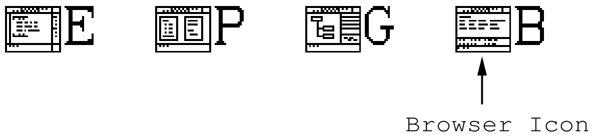


Figure 1. Book Design Browser Icon

Once you've selected the Book Design Browser, click on `Help` in the menu pane to display some introductory documentation about the Book Design Browser.

For a description of the Book Design Browser, see the section "Layout of the Book Design Browser Window", page 15.

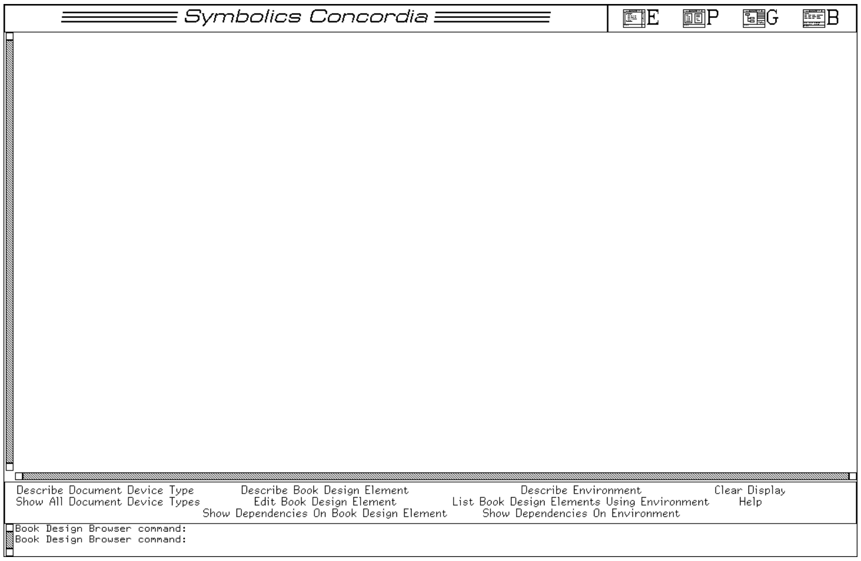
3.2. Layout of the Book Design Browser Window

Figure 2 shows the initial Book Design Browser.

The Book Design Browser consists of the "Symbolics Concordia" logo in the upper left of the pane; the icon pane in the upper right-hand corner of the pane; and the following additional panes:

Pane *Description*

Display Pane Information produced by the Book Design Browser commands is displayed in this pane. Much of the information displayed here is mouse sensitive, meaning that you can perform many operations on the information by using the mouse. See the section "Mouse Sensitivity in the Book Design Browser", page 18.


Symbolics Concordia
E
P
G
B
Describe Document Device Type
Show All Document Device Types
Describe Book Design Element
Edit Book Design Element
Show Dependencies On Book Design Element
Describe Environment
List Book Design Elements Using Environment
Show Dependencies On Environment
Clear Display
Help

```
Book Design Browser command:
Book Design Browser command:
```

Display Pane

Menu Pane

Interactor Pane

Figure 2. Initial Book Design Browser

Menu Pane The Book Design Browser commands are all listed here and each can be run by clicking Left on its name. When you click on a command name, the command appears in the Interactor Pane and prompts you for any information it might need. Note that clicking Middle on a command name displays the documentation about that command. Clicking on Help in the menu pane displays some introductory documentation about the Book Design Browser.

Interactor Pane You can type the Book Design Browser commands here. A command prompts you for any information it might need. When you click on a command name in the Menu Pane, the command appears here and prompts you for the information it needs to run.

To clear the display pane use the command "Clear Display". Information displayed in the display pane is never actually killed. You can scroll backward and forward using the scroll bar to the left.

3.3. Getting Help in the Book Design Browser

To see documentation for the commands, do one of the following:

- Click Middle on the command name in the menu.
- Use the Show Documentation command as follows:

`Show Documentation` *command-name*

To see documentation for environments and their attributes listed in displays, do one of the following:

- Click Middle on the environment or attribute name in a display. (Note that not all environments and attributes are documented. See the mouse documentation line while pointing at an environment or attribute name to make sure.)
- Use the Show Documentation command as follows:

  ```
  Show Documentation environment Environment
  ```

  or

  ```
  Show Documentation attribute Attribute
  ```

3.4. Issuing Commands in the Book Design Browser

To issue a command in the Book Design Browser, do one of the following:

- Type the command name at the "Book Design Browser command:" prompt in the Interactor Pane.

- Click Left or Right on the command name in the Menu Pane.

- Click Right on a mouse-sensitive item in a Display Pane display and choose a command from the pop-up menu.

If you're not sure what the allowable arguments for a particular command are, do one of the following:

- Check the documentation for that command by clicking Middle on the command name in the Menu Pane.

- To see a listing of possibilities when a command prompts you for an argument, press the `HELP` key. For instance:

```
Describe Document Device Type (document device type) <HELP>
=>
You are being asked to enter a document device type.

These are the possible document device types:
  3SYMANUAL LGP2   GENERIC DEX            GENERIC LGP2 ...
  APPROACH LGP2    GENERIC DEX-BACKGROUND  GENERIC MAC-SCREEN ...
  ARTICLE GENERIC  GENERIC DMP1           GENERIC SCREEN ...
  ARTICLE LGP2     GENERIC GENERIC        LETTER GENERIC ...
```

Except for "Clear Display" and "Edit Book Design Element", each of the Book Design Browser commands takes an argument which controls where the command's output is sent. The choices are:

Display Output is displayed in the Display Pane of the Browser. This is the default.

Printer Output is printed on your default printer. (Note that the output from "Describe Environment" is printed in landscape format since it can be very wide.)

Typeout Output is displayed in a temporary typeout window. This is the same as if you clicked Right on a mouse-sensitive item in the Display Pane, and then on one of the commands in the pop-up menu.

Here is an example which directs its output to your default printer:

```
Describe Environment (environment) Example (a document device
type) GENERIC GENERIC (where [default Display]) Printer
```

For a descriptions of Book Design Browser commands, see the section "Dictionary of Book Design Browser Commands", page 39.

3.5. Mouse Sensitivity in the Book Design Browser

Many items in Book Design Browser displays are mouse sensitive. Mouse sensitivity is indicated by a box outlining the item when you point the mouse at it. While pointing at a mouse-sensitive item, note the mouse documentation line for actions you can take on that item. "Mouse R: Menu" means that a menu of more commands pops up when you click Right on that item. For instance, you can see documentation for some attributes and environments by clicking Right on a mouse-sensitive attribute or environment name, and then click on "Show Documentation" in the pop-up menu.

4. Modifying a Document Device Type

4.1. Introduction

This section discusses how to modify the appearance of your documentation at the document device type level. Modifications made at this level affect all books formatted using that particular document device type. "Book Design Browser" is a Symbolics Concordia tool which facilitates this work. See the section "Book Design Browser", page 15.

You can modify two kinds of specifications at the document device type level:

- You can modify specifications for environments. When you modify an environment at the document device type level, the modification will appear in all instances of that environment in any book formatted using that document device type. See the section "An Example of a Simple Modification to an Environment in a Document Device Type", page 20. See the section "An Example of a Modification to an Environment with Dependencies in a Document Device Type", page 24.

- You can modify top-level specifications for a document device type. See the section "An Example of a Modification to a Top-level Specification in a Document Device Type", page 26.

When you make modifications at the document device type level, it is essential that you consider issues of inheritance dependencies. Remember that book design elements can inherit from other book design elements and environments can inherit from other environments. Such inheritence can create dependencies which you must keep in mind in making modifications at the document device type level. The "Book Design Browser" provides you with tools to see and identify these inheritance dependancies. See the section "Inheritance Dependency in Symbolics Concordia Book Design", page 3.

Put as simply as possible, to modify the appearance of your documentation at the document device type level you must:

1. Decide what you want to modify and how you want it to look.

2. Identify the current specification for the thing you want to modify.

3. Investigate any inheritance dependancies and thereby decide where to make your modification.

4. Make the modification.

To begin, select the Book Design Browser by clicking on Book Design Browser icon.

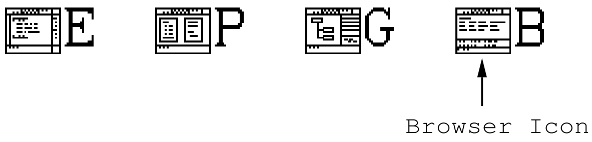


Figure 3. Book Design Browser Icon

4.2. Examples of Document Device Type Modifications

Some knowledge about book design elements, environments, counters, and attributes is assumed in these example. To find out about a particular book design element, use the command "Describe Book Design Element". To find out about a particular environment, counter, or attribute, see its documentation. See the section "Getting Help in the Book Design Browser", page 16.

4.2.1. An Example of a Simple Modification

This example makes a relatively simple modification to an environment for all document device types. There are no issues of inheritance dependency raised in this example. For an example of a modification with inheritance dependency issues, see the section "An Example of a Modification to an Environment with Dependencies in a Document Device Type", page 24.

1. **Decide what you want to modify and how you want it to look.**

   Let's say you want to modify the appearance of the "Itemize Environment" and you want that modification to appear in all document device types. You want the items in this environment to format without any blanklines between them. The attribute you need to modify is the "Spread Attribute". To affect all document device types, you need to modify Itemize in some book design element which is inherited by all other document device types. We can guess that this will be the Generic Generic document device type since it is inherited by all other document device types.

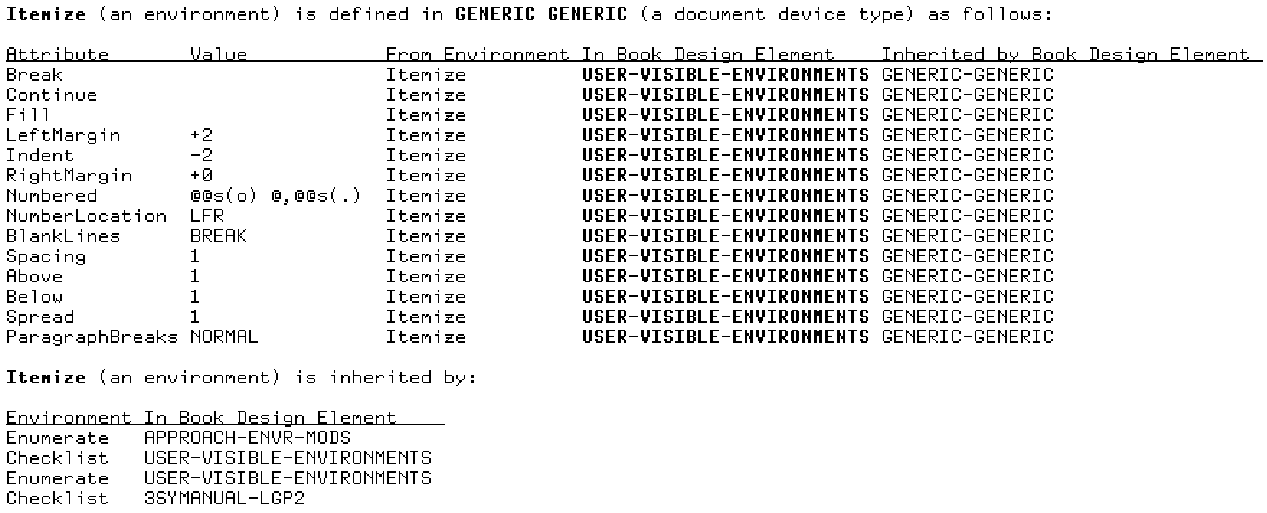
2. **Identify the current specification.**

   See what the current specification for Itemize in Generic Generic is by using the "Describe Environment" command as follows:

   ```
   Describe Environment (environment) Itemize (a document device type)
   GENERIC GENERIC
   ```

   This command produces the display in figure 4, page 21.

   In the first column, under "Attribute", look for "Spread". Notice its current value, in the second column under "Value", is "1". This means "leave one



```
Itemize (an environment) is defined in GENERIC GENERIC (a document device type) as follows:

Attribute       Value           From Environment In Book Design Element    Inherited by Book Design Element
Break                           Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Continue                        Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Fill                            Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
LeftMargin      +2              Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Indent          -2              Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
RightMargin     +0              Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Numbered        @@s(o) @,@@s(.) Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
NumberLocation  LFR             Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
BlankLines      BREAK           Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Spacing         1               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Above           1               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Below           1               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Spread          1               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
ParagraphBreaks NORMAL          Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC

Itemize (an environment) is inherited by:

Environment In Book Design Element
Enumerate   APPROACH-ENVR-MODS
Checklist   USER-VISIBLE-ENVIRONMENTS
Enumerate   USER-VISIBLE-ENVIRONMENTS
Checklist   3SYMANUAL-LGP2
```

Figure 4. Describe Environment Itemize GENERIC GENERIC

blank line between each item in an itemized list". You want to modify this to be "0" which means "don't leave any blanklines between items".

3. **Investigate any inheritance dependencies and decide where to make your modification.**

   a. In the third column, under "From Environment", you can see that this attribute and its value come from the definition of Itemize itself. This is helpful because if this attribute and its value were inherited from some other environment, then you would have to consider the inheritence dependency consequences of making the modification. Since this attribute and its value come from the definition of Itemize itself, it is safe to proceed.

   b. In the fourth column, under "In Book Design Element", notice that the definition of Itemize occurs in **sage::user-visible-environments**. Since the name of the book design element is in **boldface**, you know that this is where Itemize is defined. If the name of the book design element is in *italics*, then you know that the environment is modified in that book design element.

      In the last column, under "Inherited by Book Design Element", you can see that **sage::generic-generic** is the book design element which inherits **sage::user-visible-environments**. This is exactly what you want because you know that **sage::generic-generic** is the top-level book design element for the Generic Generic document device type. What this all tells you is that any modification to Itemize in **sage::user-visible-environments** is inherited by any book design element which inherits **sage::user-visible-environments**, for instance **sage::generic-generic**. You can see the dependencies on these book design elements by clicking Right on "user-visible-environments" and then on "Show Dependencies on Book Design Element".

These are important pieces of information because they can indicate inheritance dependencies which you might need to consider in making your modification. For more discussion of these considerations, see the section "An Example of a Modification to an Environment with Dependencies in a Document Device Type", page 24.

c. Next, you need to check for any inheritance dependencies on Itemize itself. If any other environments inherit Itemize, they are listed at the bottom of the display. In this case, Itemize is inherited by several other environments in several other book design elements. This list tells you that Enumerate and Checklist inherit Itemize in their definitions. Therefore, by modifying Itemize, you are going to affect Enumerate and Checklist. For discussion's sake, let's assume that this is OK and therefore it is safe to proceed.

4. **Make the modification.**

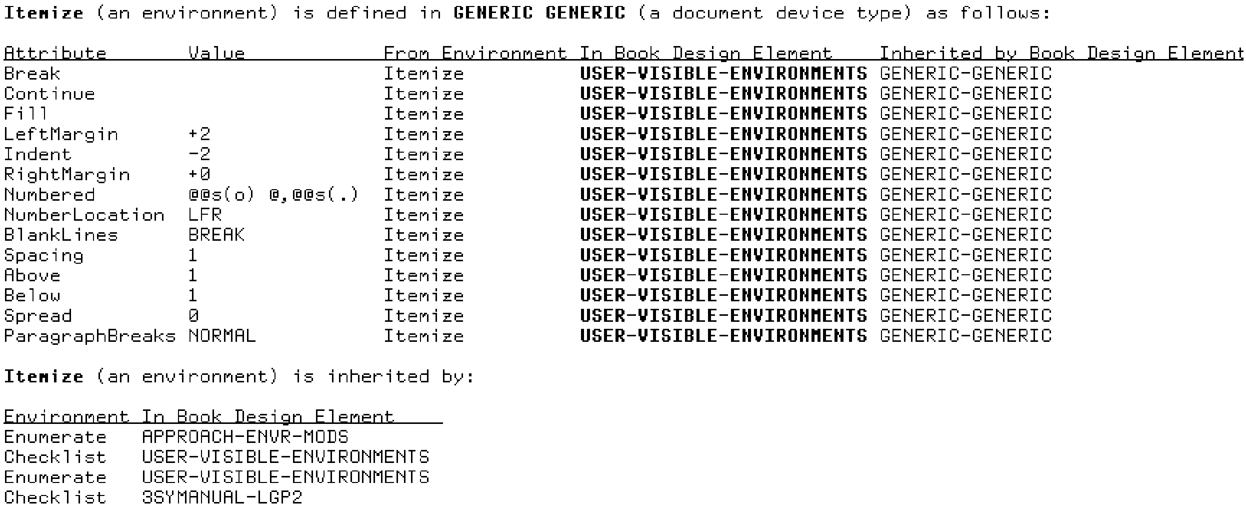
a. Now that you have investigated any and all inheritance dependencies, you can make your modification. Since the Spread attribute in Itemize is set in the **sage::user-visible-environments** book design element, and you have concluded that that is the appropriate place to make this modification, click Right on "user-visible-environments" and then on "Edit Book Design Element" in the pop-up menu. The Zmacs editor is selected, the source file containing the **sage::user-visible-environments** book design element is read into an editor buffer (if it was not already) and selected, and the cursor is positioned at the beginning of the definition of **sage::user-visible-environments**. Find the definition of Itemize within **sage::user-visible-environments**. (If the definition is not immediately visible, you can search for it using the Incremental Search command `c-S`.)

The relevant sections from the source for the book design element **sage::user-visible-environments** looks like this. (The "→→" characters are added here to show you where you want to make you modification.)

```
(define-book-design user-visible-environments ()
  (define
    (Itemize Break
             Continue
             Fill
             (Leftmargin "+2")
             (Indent "-2")
             (Rightmargin "+0")
             (Numbered "@@S(o) @,@@S(.) ")
             (Numberlocation Lfr)
             (Blanklines Break)
             (Spacing 1)
             (Above 1)
             (Below 1)
→→           (Spread 1)
             (Paragraphbreaks normal))))
```

b. Change the value of the Spread attribute from 1 to 0.

c. Compile the change by using the command c-sh-C. This adds the change you've made to your machine's memory. (Note that unless you patch this modification into a distributed system, you will loose the modification next time you boot your computer. See the following step for more information about this.) Now, when you format an Itemize environment in any document device type, no blank lines will be left between items. Now, the display produced by Describe Environment looks like figure 5, page 23.



```
Itemize (an environment) is defined in GENERIC GENERIC (a document device type) as follows:

Attribute       Value            From Environment In Book Design Element      Inherited by Book Design Element
Break                            Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Continue                         Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Fill                             Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
LeftMargin      +2               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Indent          -2               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
RightMargin     +0               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Numbered        @@s(o) @,@@s(.)  Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
NumberLocation  LFR              Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
BlankLines      BREAK            Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Spacing         1                Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Above           1                Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Below           1                Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Spread          0                Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
ParagraphBreaks NORMAL           Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC

Itemize (an environment) is inherited by:

Environment In Book Design Element
Enumerate   APPROACH-ENVR-MODS
Checklist   USER-VISIBLE-ENVIRONMENTS
Enumerate   USER-VISIBLE-ENVIRONMENTS
Checklist   3SYMANUAL-LGP2
```

Figure 5. Itemize for Generic Generic Document Device Type with the Spread Attribute Modified.

d. At this point, the modification you've made is known only to your computer. If you format a document on another computer, you will not see your modification. Likewise, if you boot your computer, your modification will be lost. To have the modification available even if you boot your computer, and to distribute your modification to other users at your site, you should patch the modification into a documentation system. For instructions on how to do this, see the section "Defining a Book Design System for a Site" in *Symbolics Concordia*, and see the section "Patching a Documentation System" in *Symbolics Concordia*.

4.2.2. An Example of a Modification to an Environment with Dependencies

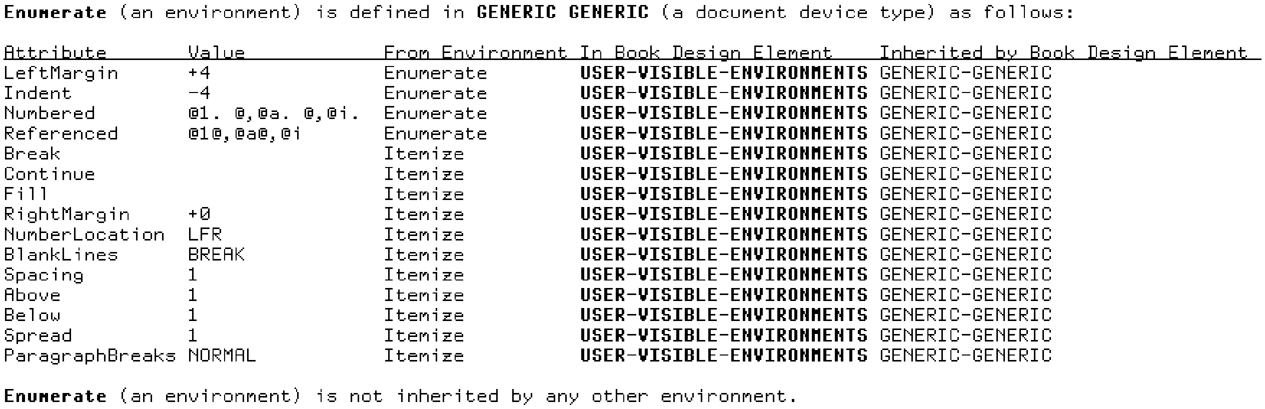
1. **Decide what you want to modify and how you want it to appear.**

   Let's say you want to modify Enumerate for all document device type so that no blank lines appear between items in the enumerated list. The attribute you need to modify is "Spread Attribute". Since you want this modification to affect all document device types, you have to make the modification someplace that is used by all other document device types. We know that the Generic Generic document device type is used by all other document device types, so that's a good place to start.

2. **See what the current definition of Enumerate is for the Generic Generic document device type.** Use the following command:

   ```
   Describe Environment (environment) Enumerate (a document device type)
   GENERIC GENERIC
   ```

   This command produces the display in figure 6, page 24.



```
Enumerate (an environment) is defined in GENERIC GENERIC (a document device type) as follows:

Attribute       Value            From Environment In Book Design Element    Inherited by Book Design Element
LeftMargin      +4               Enumerate        USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Indent          -4               Enumerate        USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Numbered        @1. @,@a. @,@i.  Enumerate        USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Referenced      @1@,@a@,@i       Enumerate        USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Break                            Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Continue                         Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Fill                             Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
RightMargin     +0               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
NumberLocation  LFR              Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
BlankLines      BREAK            Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Spacing         1                Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Above           1                Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Below           1                Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Spread          1                Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
ParagraphBreaks NORMAL           Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC

Enumerate (an environment) is not inherited by any other environment.
```

Figure 6. Describe Environment Enumerate GENERIC GENERIC

In the first column, under "Attribute", look for the attribute "Spread". Notice its current value, in column two under "Value", is "1" which means "leave one blank line between each item in an enumerated list". You want to change this to "0" which means "don't leave any blank lines between items".

3. **Investigate any inheritance dependancies and decide where to make your modification.**

   a. In the third column, under "From Environment", notice that this attribute and its value are inherited from Itemize. This means that the Spread attribute is set inside the definition of Itemize and not inside the definition of Enumerate. But you want to modify Enumerate not Itemize. What this tells you is that instead of *making* a modification to an attribute's value, you must *add* a modification to the definition of Enumerate.

   b. Look for where Enumerate is defined. In the fourth column, under "In Book Design Element", you can see that Enumerate is defined in **sage::user-visible-environments** because that book design element name is in **boldface** next to "Enumerate". This is where you can add your modification.

4. **Make your modification.**

   a. Click Right on **sage::user-visible-environments** and then on "Edit Book Design Element" in the pop-up menu. The Zmacs editor is selected, the source file containing the **sage::user-visible-environments** book design element is read into a buffer (if it was not already) and selected, and the cursor is positioned at the beginning of **sage::user-visible-environments**. Find the definition of Enumerate within **sage::user-visible-environments**. (If the definition is not immediately visible, you can search for it using the Incremental Search command `c-S`.)

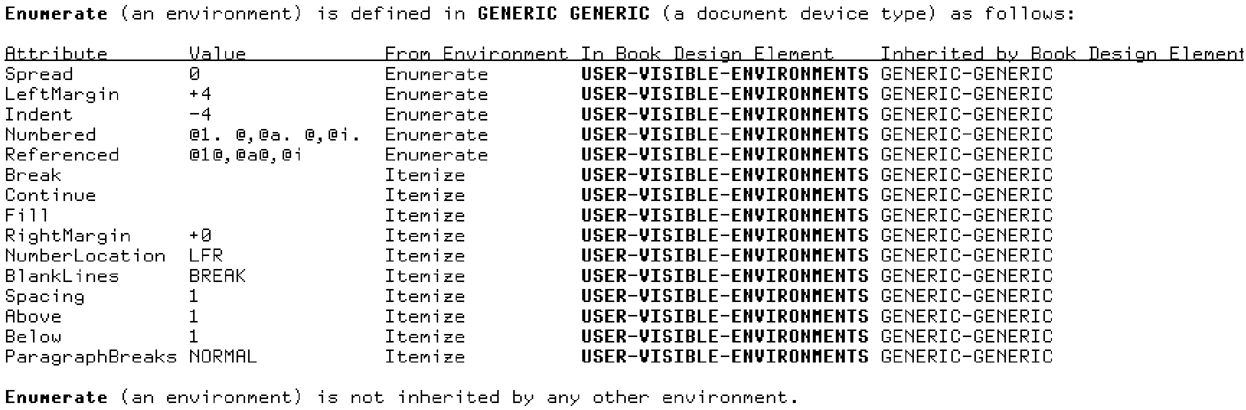
   The relevant sections from the source for the book design element user-visible-environments looks like this.

```
(define-book-design user-visible-environments ()
  (define
    (Enumerate (Use Itemize)
               (Leftmargin "+4")
               (Indent "-4")
               (Numbered "@1. @,@a. @,@i. ")
               (Referenced "@1@,@a@,@i"))))
```

   Inside the definition of Enumerate, add "Spread 0" as follows:

```
(define-book-design user-visible-environments ()
  (define
    (Enumerate (Use Itemize)
               (Spread 0)
               (Leftmargin "+4")
               (Indent "-4")
               (Numbered "@1. @,@a. @,@i. ")
               (Referenced "@1@,@a@,@i"))))
```

b. Compile the change by using the command `c-sh-C`. This adds the change you've made to you machine's memory. Now, when you format an Enumerate environment in any document device type, no blank lines will be left between items. Now, the display produced by Describe Environment looks like figure 7, page 26.



```
Enumerate (an environment) is defined in GENERIC GENERIC (a document device type) as follows:

Attribute       Value           From Environment In Book Design Element      Inherited by Book Design Element
Spread          0               Enumerate        USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
LeftMargin      +4              Enumerate        USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Indent          -4              Enumerate        USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Numbered        @1. @,@a. @,@i. Enumerate        USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Referenced      @1@,@a@,@i      Enumerate        USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Break                           Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Continue                        Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Fill                            Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
RightMargin     +0              Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
NumberLocation  LFR             Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
BlankLines      BREAK           Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Spacing         1               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Above           1               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
Below           1               Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC
ParagraphBreaks NORMAL          Itemize          USER-VISIBLE-ENVIRONMENTS GENERIC-GENERIC

Enumerate (an environment) is not inherited by any other environment.
```

Figure 7. Enumerate for Generic Generic Document Device Type with the Spread Attribute Modified.

5. At this point, the modification you've made is known only to your computer. If you format a document on another computer, you will not see your modification. Likewise, if you boot your computer, your modification will be lost. To have the modification available even if you boot your computer, and to distribute your modification to other users at your site, you should patch the modification into a documentation system. For instructions on how to do this, see the section "Defining a Book Design System for a Site" in *Symbolics Concordia*, and see the section "Patching a Documentation System" in *Symbolics Concordia*.

4.2.3. An Example of a Modification to a Top-level Specification in a Document Device Type

In this example you can see how to make a modification to a top-level specification in a document device type. Top-level specifications are specifications in top-level style or text clauses in top-level book design elements. See the section "Book Design Elements", page 11.

Since this example concerns itself with top-level specifications for a document device type, you must start your research at the document device type level instead of at the environment level as in the other examples. Here's how to proceed:

1. **Decide what you want to modify and how you want it to appear.**

   Let's say you want the top-level leftmargin to be 2 inches in the 3symanual LGP2 document device type. That's slightly larger than it currently is. This means that for any book formatted as a 3symanual LGP2 document device type, all running text (that is, all text not inside an environment, pageheading or pagefooting, or section heading, has a leftmargin of 2 inches.)

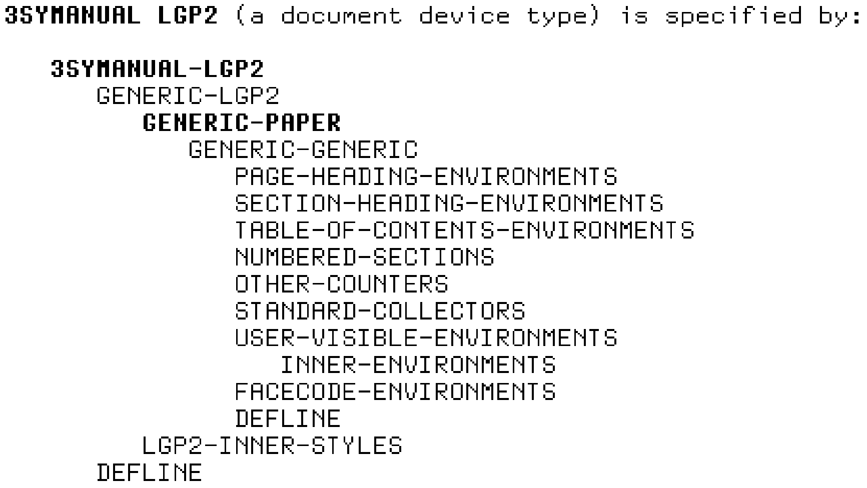
2. **Find out where the current top-level leftmargin is set in the 3symanual LGP2 document device type.**

   Since top-level specifications are set in top-level book design elements, you must first find out which book design element is the top-level one for the 3symanual LGP2 document device type.

   a. Use the following command to see what book design elements make up the specification for the 3symanual LGP2 document device type:

      ```
      Describe Document Device Type (document device type) 3SYMANUAL LGP2
      ```

      The command produces the display in Figure 8.



```
3SYMANUAL LGP2 (a document device type) is specified by:

   3SYMANUAL-LGP2
      GENERIC-LGP2
         GENERIC-PAPER
            GENERIC-GENERIC
               PAGE-HEADING-ENVIRONMENTS
               SECTION-HEADING-ENVIRONMENTS
               TABLE-OF-CONTENTS-ENVIRONMENTS
               NUMBERED-SECTIONS
               OTHER-COUNTERS
               STANDARD-COLLECTORS
               USER-VISIBLE-ENVIRONMENTS
                  INNER-ENVIRONMENTS
               FACECODE-ENVIRONMENTS
               DEFLINE
         LGP2-INNER-STYLES
      DEFLINE
```

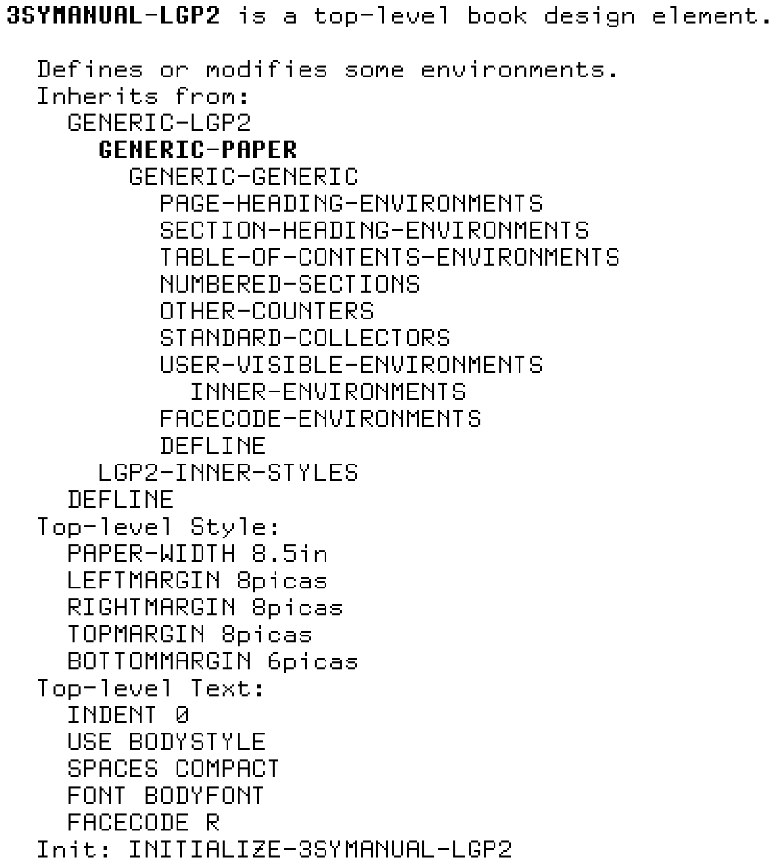
Figure 8. Describe Document Device Type 3symanual LGP2

The display lists the family tree of book design elements for the 3symanual LGP2 document device type. You can see from the display that two of the book design elements are "top-level", those in boldface. Top-level elements set the top-level style and text specifications so one of these is the book design element you want to look at. Since there are two listed, you must decide which of them is the appropriate place to make your modification. From its name, it is safe to assume that **sage::3symanual-lgp2** is the place to look. See the section "Book Design Elements", page 11.

b. Use the following command to see the specification for **sage::3symanual-lgp2**:

```
Describe Book Design Element (a book design element) 3SYMANUAL-LGP2
```

The command produces the display in Figure 9.



```
3SYMANUAL-LGP2 is a top-level book design element.

  Defines or modifies some environments.
  Inherits from:
    GENERIC-LGP2
      GENERIC-PAPER
        GENERIC-GENERIC
          PAGE-HEADING-ENVIRONMENTS
          SECTION-HEADING-ENVIRONMENTS
          TABLE-OF-CONTENTS-ENVIRONMENTS
          NUMBERED-SECTIONS
          OTHER-COUNTERS
          STANDARD-COLLECTORS
          USER-VISIBLE-ENVIRONMENTS
            INNER-ENVIRONMENTS
          FACECODE-ENVIRONMENTS
          DEFLINE
      LGP2-INNER-STYLES
    DEFLINE
  Top-level Style:
    PAPER-WIDTH 8.5in
    LEFTMARGIN 8picas
    RIGHTMARGIN 8picas
    TOPMARGIN 8picas
    BOTTOMMARGIN 6picas
  Top-level Text:
    INDENT 0
    USE BODYSTYLE
    SPACES COMPACT
    FONT BODYFONT
    FACECODE R
  Init: INITIALIZE-3SYMANUAL-LGP2
```

Figure 9. Describe Book Design Element 3symanual-lgp2

From this display you can see that the current setting for the top-level leftmargin is "8picas" (which, at 6 picas to the inch, is 1 5/16").

3. **Investigate any inheritance dependencies and decide where to make your modification.**

You've actually already done most of this step since you decided earlier to look at **sage::3symanual-lgp2** instead of **sage::generic-paper**. But you should still make sure that any modifications you make to **sage::3symanaul-lgp2** aren't inherited by other book design elements. So, use the following command to see if **sage::3symanual-lgp2** has any children:

```
Show Dependencies On Book Design Element (a book design element)
3SYMANUAL-LGP2
```

This command produces a list of four other book design elements which inherit **sage::3symanual-lpg2**:

```
LISP-DICTIONARY-LGP2
ARTICLE-LGP2
DOC-EX-LGP2
MAC-DOC-LGP2
```

Each of these is a top-level book design element for another document device type (you can tell from their names), so it is probably safe to assume that each of them has their own top-level specifications which would override any modification you make to **sage::3symanual-lgp2**. If you want to make sure, just click Right on each and then on "Describe Book Design Element" in the pop-up menu. You can see from the displays that each does indeed have their own top-level specifications so it is safe to proceed.

4. **Make your modification.**

   a. Click Right on **sage::3symanual-lgp2** and then on "Edit Book Design Element" in the pop-up menu. The Zmacs editor is selected, the source file containing **sage::3symanual-lgp2** is read into a buffer (if it was not already) and selected, and the cursor is positioned at the beginning of **sage::3symanual-lgp2**.

      The essential source code for **sage::3symanual-lgp2** follows with "→→→" marking the leftmargin attribute.

```
(define-book-design 3symanual-lgp2 ()
  (use generic-lgp2)
  (use defline)
  (define
    ...)
  (modify
    ...)
  (first
    (envr (text
            (Paper-Width "8.5in")
→→→         (LeftMargin "8picas")
            (RightMargin "8picas")
            (TopMargin "8picas")
            (BottomMargin "6picas")))
    (envr (text (Indent "0")
                (Use BodyStyle)
                (Spaces Compact)
                (Font BodyFont)
                (FaceCode R)))
    (init initialize-3symanual-lgp2)))
```

b. Change the "8picas" to "2inches". This adds 11/16" to the leftmargin since 8 picas equals 1 5/16". (You could also change this value to "12picas" since that equals "2inches".)

c. Compile the change by using the command `c-sh-C`. This adds the change you've made to your machine's memory. Now when you format a record registered as a 3symanual LGP2 document device type you will get a 2" top-level leftmargin.

5. At this point, the modification you've made is known only to your computer. If you format a document on another computer, you will not see your modification. Likewise, if you boot your computer, your modification will be lost. To have the modification available even if you boot your computer, and to distribute your modification to other users at your site, you should patch the modification into a documentation system. For instructions on how to do this, see the section "Defining a Book Design System for a Site" in *Symbolics Concordia*, and see the section "Patching a Documentation System" in *Symbolics Concordia*.

5. Modifying a Single Book

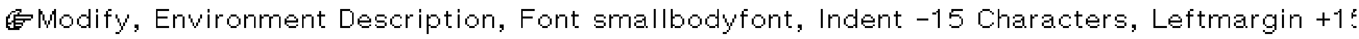
You can modify the appearance of all occurrences of an environment in a single book by using the command markup Modify.

1. Insert the command markup in a sab buffer by typing `s-m Modify`. The command markup is inserted in the buffer opened so that all the attributes available for modification are listed.

2. Fill in the name of the environment you want to modify.

3. Set the attributes which you want to modify.

4. Click Right on one of the Modify markup icons and on "Change View" in the pop-up menu to close the display. Figure 10 shows a Modify markup which modifies the Description environment.



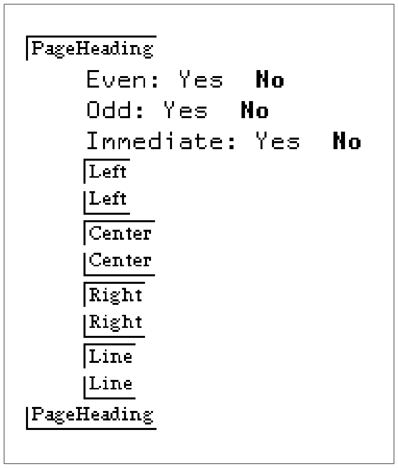
```
☞Modify, Environment Description, Font smallbodyfont, Indent -15 Characters, Leftmargin +1
```

Figure 10. Modify Markup modifying Description environment.

By convention, you should place Modify command markups only in the script (top-level) record of your book. This way, you can more easily keep track of what's affecting the formatting.

5.1. Specifying Page Headings and Footings

You can specify running heads for your document by placing a Pageheading command in your top level record. Create Markup Pageheading requests a pageheading command. A template pops up to allow you to insert the information to be printed.



PageHeading
Even: Yes **No**
Odd: Yes **No**
Immediate: Yes **No**
Left
Left
Center
Center
Right
Right
Line
Line
PageHeading

The fields in the template are:

`Even` and `Odd`
Whether this heading information is for an even page (lefthand or *verso*) heading or an odd page (righthand or *recto*) heading. You need two pageheading commands, one with **yes** for Even and the other with **yes** for Odd.

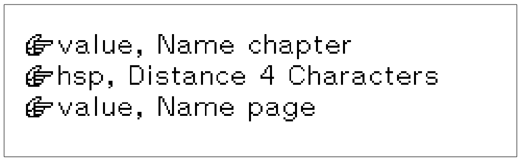
`Immediate`
Whether the headings are to take effect immediately, on the current page, or not until the second page. If you want the headings on the first page to differ from those on the second and subsequent pages, you need two pageheading commands, one with Immediate **yes**, followed by another with Immediate **no**.

`Left`, `Center`, and `Right`
The three text areas in the heading. You insert the text you want to appear in each place. To get the page number, insert the markup Value and specify the counter Page.

`Line`
An additional line of text, appearing under the left heading and possibly extending across the top of the page. If you place the command Replicate-Pattern followed by an underscore, you get a bar across the page. (Note: you insert Replicate-Pattern with `c-U s-M`, since it is an internal command.)

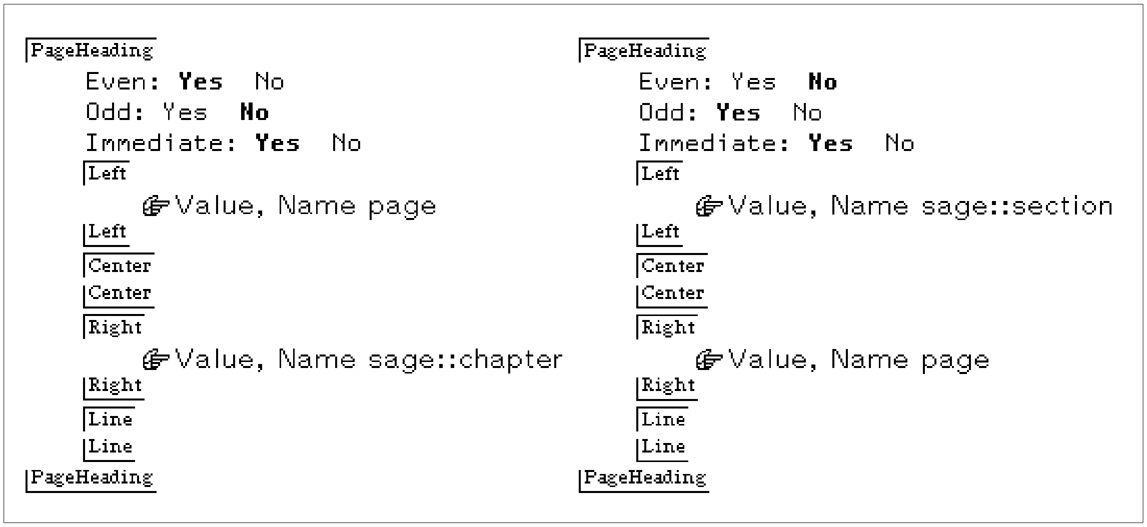
To insert literal spaces into a heading, for instance to specify that you want the current chapter title followed by 4 spaces and then the page, use the markup Hsp (horizontal space) which takes a horizontal distance as an argument.

For example:



```
☞value, Name chapter
☞hsp, Distance 4 Characters
☞value, Name page
```

Here is what the command templates might look like to print headings for this section:



```
PageHeading
    Even: [✓ Yes] No
    Odd: Yes [✓ No]
    Immediate: [✓ Yes] No
    Left
        ☞Value, Name page
    Left
    Center
    Center
    Right
        ☞Value, Name sage::chapter
    Right
    Line
    Line
PageHeading
```

```
PageHeading
    Even: Yes [✓ No]
    Odd: [✓ Yes] No
    Immediate: [✓ Yes] No
    Left
        ☞Value, Name sage::section
    Left
    Center
    Center
    Right
        ☞Value, Name page
    Right
    Line
    Line
PageHeading
```

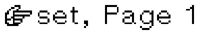
Page footings are specified similarly.

Page headings and footings have no effect online.

5.2. Handling Tables of Contents and Frontmatter

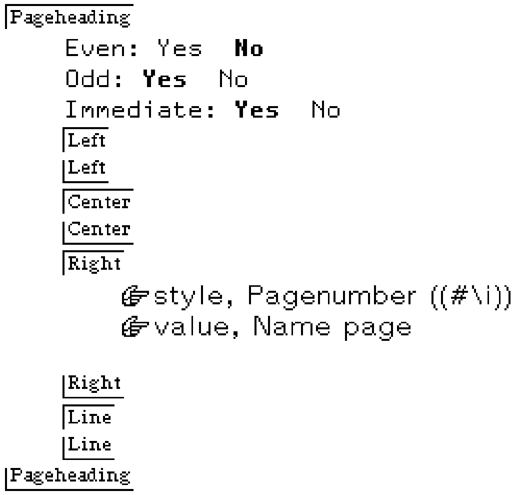
Table of contents pages automatically have their page numbers printed in lower case roman numerals. The page counter is automatically reset after the table of contents, list of figures and list of tables so that the text starts on page 1 and page numbers are printed in arabic numerals.

You can add additional frontmatter, for example a Preface or Acknowledgments, simply by setting the page counter by hand in your top level record just before the include links for the Preface and/or Acknowledgments. Take into account how many pages the table of contents and lists of figures and tables take up and use the markup Set, into which you insert the markup Value, specifying the counter **sage::page** followed by the appropriate value.



```
☞set, Page 1
```

Then insert the appropriate pageheading and pagefooting commands to specify page headings and footings for this additional frontmatter. Just before the markup for Value Page, insert the markup Style. Select the parameter Pagenumber by clicking on it and give it the template "@i", which specifies lower case roman numerals.



```
Pageheading
    Even: Yes  No
    Odd: Yes  No
    Immediate: Yes  No
    Left
    Left
    Center
    Center
    Right
        ☞style, Pagenumber ((#\i))
        ☞value, Name page

    Right
    Line
    Line
Pageheading
```

Then, following the links to the Preface and/or Acknowlegments, reset the page counter to 1 and add a style parameter that specifies a Pagenumber template of "@1". (For a list of the available templates, see the section "Numbering Templates" in *Symbolics Concordia*.)

5.3. Creating Title Pages

Create your title page by creating a record of type fragment and link it at the beginning of the top-level record for your book using a Contents link.

Unless you want your title page to have the same page headings as the rest of your document, you probably want to give it a Pageheading command. See the section "Specifying Page Headings and Footings", page 31.

You probably do not want it numbered, so you should adjust the page counter appropriately after the link to your title page. See the section "Handling Tables of Contents and Frontmatter", page 33.

You use formatting environments to place the text where you want it on your title page (see the section "Positioning Text in Concordia" in *Symbolics Concordia*). If you have a boiler plate or copyright page, you should be sure to place a Newpage command in the appropriate place.

If your title pages are very elaborate, you might want to define a title page environment in your book design to handle them. See the Article document type in SYS:CONCORDIA;BD-ARTICLE.LISP for an example of a title page environment.

6. Modifying a Single Instance of an Environment

Sometimes you might find that you want to modify the appearance of a single instance of an environment. Please note that this is not recommended. Here's an example of why: Let's say a record is included in a dictionary and also in some reference cards. If you modify an environment in the record to format just the way you want in the dictionary, then it might look entirely wrong in the reference cards. This is an issue of modularity. See the section "Workstyle Issues in Writing with Symbolics Concordia" in *Symbolics Concordia*.

If you must modify a single instance of an environment you do this by modifying the environment markup. For example, you might want to close up the space around some instance of an environment, in which case you modify the Above and Below attributes of the environment markup. Or, you might decide that the items in a particular list should be bulleted rather than numbered. In this case you replace one environment with another by modifying the name attribute of the environment markup.

Follow this procedure to edit the attributes of a single instance of an evironment.

1. Use one of the following to begin the procedure:

   - Position the mouse cursor on the markup (either delimiter) and click c-m-Right.

   - Put the editor cursor within an environment and click on Change Environment under **Markup** in the menu.

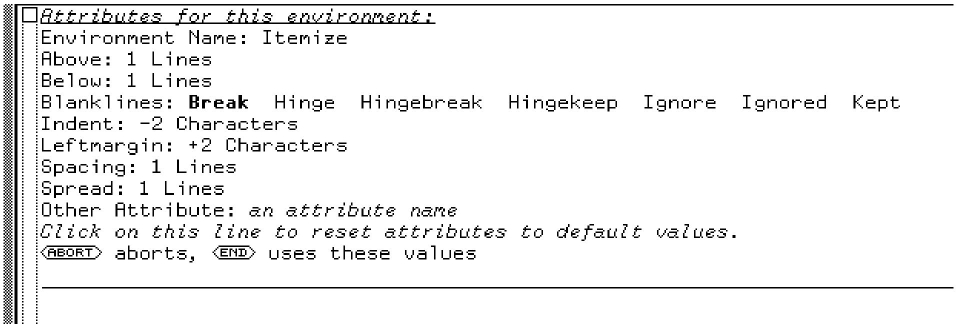
   - Put the editor cursor within an environment and use the editor command "Change Environment".

   A menu pops up showing the most common and useful attributes for that environment and their current values. (If the environment has never been modified, these are the attributes' default values.)

2. For most attributes, you simply click on the value and type in a new one. Where choices are given, as in the Blanklines attribute, the current value is shown in boldface. See the section "List of Symbolics Concordia Attributes" in *Symbolics Concordia*.

3. Click on another choice to select it. For a description of the common attributes and their valid values, see the section "The Most Common Environment Attributes" in *Symbolics Concordia*. The menu item Other Attributes lets you select any of the other attributes provided by Symbolics Concordia. Click on "an attribute name", then press HELP to see a complete list.

Modifications made to a single instance of an environment are local to that instance environment. In other words, changing the attributes of one Itemize envi-



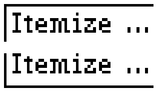
```
□Attributes for this environment:
Environment Name: Itemize
Above: 1 Lines
Below: 1 Lines
Blanklines: Break  Hinge  Hingebreak  Hingekeep  Ignore  Ignored  Kept
Indent: -2 Characters
Leftmargin: +2 Characters
Spacing: 1 Lines
Spread: 1 Lines
Other Attribute: an attribute name
Click on this line to reset attributes to default values.
<ABORT> aborts, <END> uses these values
```

Figure 11. The menu shows the current attributes of an Itemize environment.

ronment does not alter any other Itemize environment. If you need to make the same modification for more that one instance of an environment, mark the environment as a region, kill it, and yank it back at another location. (See **Kill History** in the menu.)

To modify all instances of an environment in a single book, see the section "Modifying a Single Book", page 31. To modify all instances of an environment for all books of a particular document device type, see the section "Modifying a Document Device Type", page 19.

Modified environments are denoted in the record by an ellipsis. See Figure 12.



```
Itemize ...
Itemize ...
```

Figure 12. Modified Environment Markup showing telltale ellipsis.

To return to the default values of all attributes, click on the next-to-last line of the menu. The ellipsis on the environment markup delimiters will disappear. Values of individual attributes can only be reset manually.

When you change from one environment to another (by changing the Name attribute), the default definition for the new environment is inserted in the menu.

"Query Change Environments" enables you to change instances of one environment to instances of some other environment within the current buffer. This is convenient if you suddenly realize that all your bulleted lists should be numbered lists.

7. Dictionary of Book Design Browser Commands

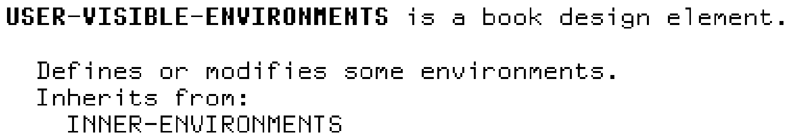
7.1. Clear Display

Clears the display pane of the Book Design Browser. The display is not actually lost. You can use the scroll bar to the left of the Display Pane to scroll back to previous output. Output for each command is separated by a line of dashes.

7.2. Describe Book Design Element

`Describe Book Design Element` *book-design*

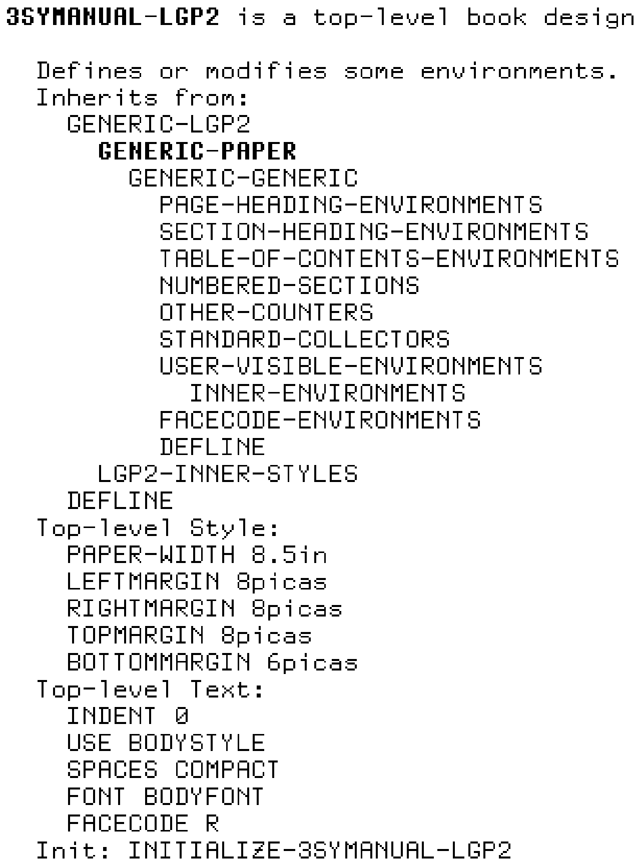
Lists other book design elements from which *book-design* inherits. In other words, this command lists *book-design's* parent and grandparent book design elements. See the section "Inheritance Dependency in Symbolics Concordia Book Design", page 3. The command also includes a brief description of what the element does.



```
USER-VISIBLE-ENVIRONMENTS is a book design element.

  Defines or modifies some environments.
  Inherits from:
    INNER-ENVIRONMENTS
```

Figure 13. Describe Book Design Element user-visible-environments

If *book-design* is a top-level book design element, then the command lists the top-level style and top-level text clauses, which includes top-level specifications, and the init clause function name.



```
3SYMANUAL-LGP2 is a top-level book design element.

  Defines or modifies some environments.
  Inherits from:
    GENERIC-LGP2
      GENERIC-PAPER
        GENERIC-GENERIC
          PAGE-HEADING-ENVIRONMENTS
          SECTION-HEADING-ENVIRONMENTS
          TABLE-OF-CONTENTS-ENVIRONMENTS
          NUMBERED-SECTIONS
          OTHER-COUNTERS
          STANDARD-COLLECTORS
          USER-VISIBLE-ENVIRONMENTS
            INNER-ENVIRONMENTS
          FACECODE-ENVIRONMENTS
          DEFLINE
      LGP2-INNER-STYLES
    DEFLINE
  Top-level Style:
    PAPER-WIDTH 8.5in
    LEFTMARGIN 8picas
    RIGHTMARGIN 8picas
    TOPMARGIN 8picas
    BOTTOMMARGIN 6picas
  Top-level Text:
    INDENT 0
    USE BODYSTYLE
    SPACES COMPACT
    FONT BODYFONT
    FACECODE R
  Init: INITIALIZE-3SYMANUAL-LGP2
```

Figure 14. Describe Book Design Element 3symanual-lgp2

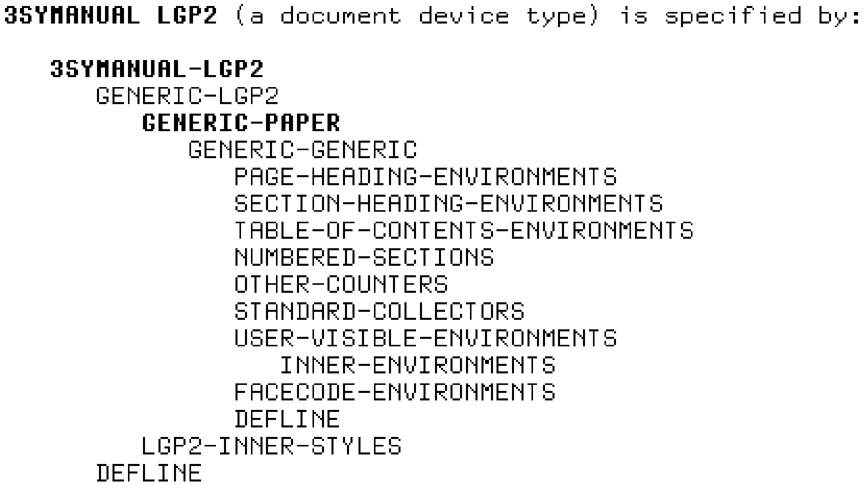
Use this command to investigate the top-level specifications for a document device type (once you've found the top-level book design element for the given document device type). See the section "Describe Document Device Type", page 40. See the section "An Example of a Modification to a Top-level Specification in a Document Device Type", page 26.

You can also use this command to investigate the book design elements which contribute to the definition of an environment. See the section "Describe Environment", page 41. See the section "Examples of Document Device Type Modifications", page 20.

7.3. Describe Document Device Type

`Describe Document Device Type` *document-device-type*

Lists the book design elements which make up *document-device-type*.



```
3SYMANUAL LGP2 (a document device type) is specified by:

   3SYMANUAL-LGP2
      GENERIC-LGP2
         GENERIC-PAPER
            GENERIC-GENERIC
               PAGE-HEADING-ENVIRONMENTS
               SECTION-HEADING-ENVIRONMENTS
               TABLE-OF-CONTENTS-ENVIRONMENTS
               NUMBERED-SECTIONS
               OTHER-COUNTERS
               STANDARD-COLLECTORS
               USER-VISIBLE-ENVIRONMENTS
                  INNER-ENVIRONMENTS
               FACECODE-ENVIRONMENTS
               DEFLINE
         LGP2-INNER-STYLES
      DEFLINE
```

Figure 15. Describe Document Device Type 3symanual LGP2

The first book design element listed is the top-level element for *document-device-type*. The other elements are those which the top-level element inherits from, that is, its parents. The indentation indicates which elements inherit from which other elements. Therefore, in this example, **sage::3symanual-lgp2** (child) inherits from **sage::generic-lgp2** (parent) which in turn inherits from **sage::generic-paper** (grandparent).

Boldfaced elements set top-level specifications.

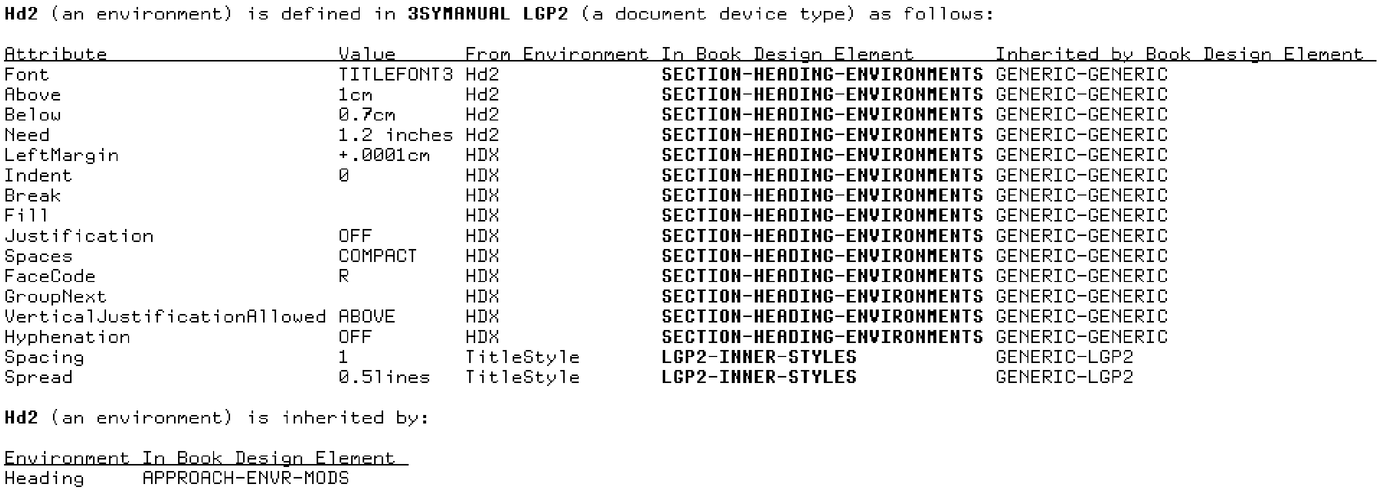
Use this command when you want to make a modification to the top-level specifications for a document device type. See the section "An Example of a Modification to a Top-level Specification in a Document Device Type", page 26.

7.4. Describe Environment

`Describe Environment` *environment document-device-type*

Displays a table of the attributes and theirs values which make up the default definition for *environment* in *document-device-type*. See the section "Dictionary of Symbolics Concordia Markup Environments and Commands" in *Symbolics Concordia*.

This table also includes environments and book design elements which are *environment's* parent environments and book design elements. See the section "Inheritance Dependency in Symbolics Concordia Book Design", page 3.



```
Hd2 (an environment) is defined in 3SYMANUAL LGP2 (a document device type) as follows:

Attribute                    Value      From Environment In Book Design Element      Inherited by Book Design Element
Font                         TITLEFONT3 Hd2              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Above                        1cm        Hd2              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Below                        0.7cm      Hd2              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Need                         1.2 inches Hd2              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
LeftMargin                   +.0001cm   HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Indent                       0          HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Break                                   HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Fill                                    HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Justification                OFF        HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Spaces                       COMPACT    HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
FaceCode                     R          HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
GroupNext                               HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
VerticalJustificationAllowed ABOVE      HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Hyphenation                  OFF        HDX              SECTION-HEADING-ENVIRONMENTS GENERIC-GENERIC
Spacing                      1          TitleStyle       LGP2-INNER-STYLES            GENERIC-LGP2
Spread                       0.5lines   TitleStyle       LGP2-INNER-STYLES            GENERIC-LGP2

Hd2 (an environment) is inherited by:

Environment In Book Design Element
Heading     APPROACH-ENVR-MODS
```

Figure 16. Describe Environment Example 3symanual LGP2

Column #/name *Description*

1/Attribute The names of the attributes which make up the definition of *environment* in *document-device-type*.

2/Value The value of the attribute in column 1. If this slot is empty, then the value of that attribute is "On" or "Yes". If the value is "*pageheading*" or "*pagefooting*", then the actual value is too lengthy to be of use in the display. In that case, you can see the full attribute value in the source file.

3/From Environment
The environment which donates the attribute in column 1 to the definition of *environment* for *document-device-type*.

4/In Book Design Element
The element which contains the environment in column 3. Boldface type indicates that the environment in column 3 is defined in this element; italics indicates that the environment in column 3 is modified in this element.

5/Inherited by Book Design Element
The element which inherits the element in column 4.

In addition, after the table, the command lists any environments which are *environment's* children (that is those environments which inherit from and are therefore dependent upon *environment*'s definition) and the book design elements in which they are defined.

The display shows the complete inheritance of *environment* for *document-device-type*. Many entries in this display are mouse sensitive. See the mouse documentation line for instructions.

Use this command to investigate the definition of an environment for a document device type. See the section "An Example of a Simple Modification to an Environment in a Document Device Type", page 20. See the section "An Example of a Modification to an Environment with Dependencies in a Document Device Type", page 24.

7.5. Edit Book Design Element

`Edit Book Design Element` *book-design*

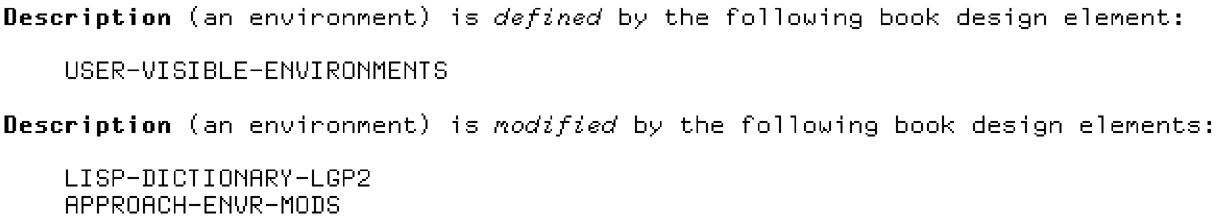
Selects the Zmacs editor, reads the source file for *book-design* into an editor buffer (if it is not already read in), selects the buffer, and positions the cursor at the beginning of the **sage:define-book-design** form for *book-design*.

See the section "Modifying a Document Device Type", page 19.

7.6. List Book Design Elements Using Environment

`List Book Design Elements Using Environment` *environment*

Lists the book design elements which define and/or modify *environment*.



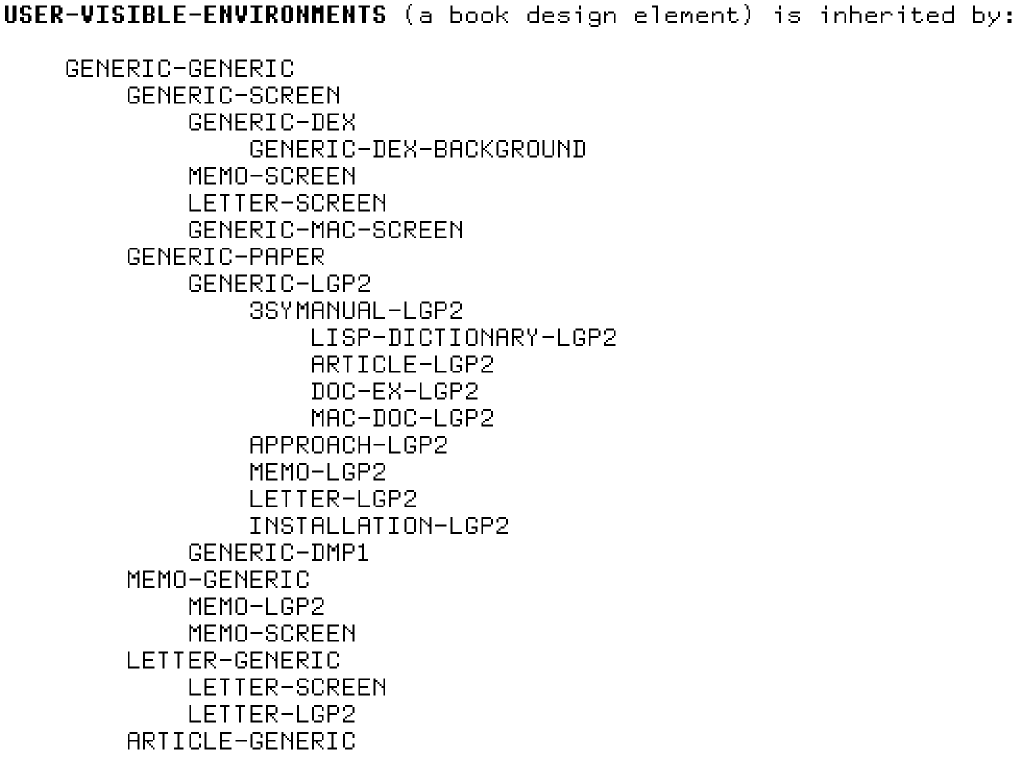
```
Description (an environment) is defined by the following book design element:

    USER-VISIBLE-ENVIRONMENTS

Description (an environment) is modified by the following book design elements:

    LISP-DICTIONARY-LGP2
    APPROACH-ENVR-MODS
```

Figure 17. List Book Design Elements Using Environment Description

7.7. Show Dependencies on Book Design Element

`Show Dependencies On Book Design Element` *book-design*

Lists book design elements which inherit *book-design*. In other words, this command lists *book-design's* children and grandchildren book design elements. See the section "Inheritance Dependency in Symbolics Concordia Book Design", page 3.



```
USER-VISIBLE-ENVIRONMENTS (a book design element) is inherited by:

    GENERIC-GENERIC
        GENERIC-SCREEN
            GENERIC-DEX
                GENERIC-DEX-BACKGROUND
            MEMO-SCREEN
            LETTER-SCREEN
            GENERIC-MAC-SCREEN
        GENERIC-PAPER
            GENERIC-LGP2
                3SYMANUAL-LGP2
                    LISP-DICTIONARY-LGP2
                    ARTICLE-LGP2
                    DOC-EX-LGP2
                    MAC-DOC-LGP2
                APPROACH-LGP2
                MEMO-LGP2
                LETTER-LGP2
                INSTALLATION-LGP2
            GENERIC-DMP1
        MEMO-GENERIC
            MEMO-LGP2
            MEMO-SCREEN
        LETTER-GENERIC
            LETTER-SCREEN
            LETTER-LGP2
        ARTICLE-GENERIC
```

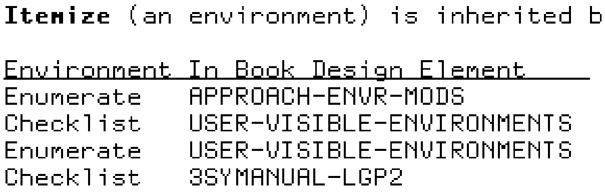
Figure 18. Show Dependencies On Book Design Element user-visible-environments

This example shows that **sage::user-visible-environments** (parent) is inherited by **sage::generic-generic** (child) which in turn is inherited by **sage::generic-screen**, **sage::generic-paper**, and so on (grandchildren). **sage::generic-paper** (parent), in turn, is inherited by **sage::generic-lgp2** (child) which is inherited by **sage::3symanual-lgp2** (grandchild).

7.8. Show Dependencies on Environment

`Show Dependencies On Environment` *environment*

Lists environments which inherit *environment*. In other words, this command lists *environment's* children. See the section "Inheritance Dependency in Symbolics Concordia Book Design", page 3.



```
Itemize (an environment) is inherited by:

Environment In Book Design Element
Enumerate   APPROACH-ENVR-MODS
Checklist   USER-VISIBLE-ENVIRONMENTS
Enumerate   USER-VISIBLE-ENVIRONMENTS
Checklist   3SYMANUAL-LGP2
```

Figure 19. Show Dependencies On Environment Itemize

8. Book Design Functions and Variables

sage::*allow-index-commands-from-record-titles* *Variable*

Controls whether or not all record names become index entries. The default is **t**, all record names provide index entries. If you have indexed your document using Index Primary and Index Secondary commands (see the section "Creating an Index in a Symbolics Concordia Document" in *Symbolics Concordia*), you might want to turn off the automatic indexing. In that case set this variable to **nil** in your book design.

sage:*default-device-type* *Variable*

Specifies the device type for which to format a topic. There are two devices currently supported by Symbolics Concordia, SCREEN and LGP2. SCREEN is the default for online formatting. LGP2 is the default for hardcopy.

sage:*default-document-type* *Variable*

Controls what book design is used for formatting when no **sage::register-book** form has been evaluated for the topic. The default is GENERIC for online formatting and 3SYMANUAL for hardcopy formatting. Your book design system can set **sage:*default-document-type*** for your site to your book design.

sage:define-book-design *name options* &body *forms* *Function*

A **sage:define-book-design** form can contain six kinds of forms: **use**, **define**, **modify**, **first**, **counters**, and **collectors**.

use Takes a book design and inherits the attributes of that design. Generally you want to **use** Generic-Generic and one or more other predefined book designs. If you want to supersede some of the predefined internal book designs (see the section "Book Design Elements", page 11), you might find it easier to specify explicitly which internal book designs you want, omitting Generic-Generic. The Lisp command **sage:graph-book-design-users** is useful to check which book designs use which other book designs so that you can be sure you are omitting those you do not want included. See the function **sage:graph-book-design-users**, page 50. **sage:graph-book-design-uses** can also be used to see exactly which book designs are being included in any given book design.

define Redefines some attribute(s) of the parent book designs completely or defines some environments which are exclusive to *name*. You specify all the parameters for the attribute(s).

modify

Modifies some attribute(s) of the parent book designs. You specify only those parameters you want to change; all others are inherited from whatever the parent design is.

first

Sets up the top-level environment (**envr**), which defines the paper size and margins, and the environment (**init**) which calls the initialize function for the book design. The initialize form can set up page headings and footings.

collectors

Lets you define behavior associated with the frontmatter and endmatter of a book.

Its syntax is

```
(define-book-design ()
  ...
  (collectors
    collector-spec-1
    collector-spec-2
    ...))
```

Each *collector-spec-n* looks like

(*collector-name* &key *name init*)

collector-name is a name from the set {Contents, Tables, Figures, Index}

name should either be a string or a symbol. If a string, it is presumed to be the "name" for that document section, and will be emitted as Prefatory text for that section. For example, the default name for Contents is "Table of Contents". If a symbol, it should name a function of no arguments that is called within the context of **sage:making-sage-directives** to produce the name for the section (see **sage:making-sage-directives**). *name* is not required. If not specified, no name appears.

init, if present, should be a symbol naming a function of no arguments that is called within **sage::making-sage-directives** to initialize formatter variables, etc. before this part of the document is formatted. It can change the pageheadings, change the rendering of the page counter, etc. Often it will do **NewPage UntilEven**. The default *init* for Contents is **sage::init-standard-table-of-contents**. The default *init* for Index is **sage::init-standard-index**.

Please note: The **collectors** clause lets you define behavior, not modify it. Therefore, if you want the table of contents to come out as "ToC", you need to include:

```
(contents :name "ToC"
          :init init-standard-table-of-contents)
```

counters

Lets you define behavior associated with section headings like chapter, subsection, and the like. For example, in the book design element **sage::numbered-sections**, a counter for Chapter is defined as follows:

```
(chapter
   :numbered "@1."
   :referenced "@1"
   :title ((title counter-contents)
            (idirs
                (envr (HD1 above 2 below 2 facecode b)
                      counter-contents (spaces 2) title)))
   :contents ((title counter-contents)
                (render-regular-toc
                  '(Tc1) title counter-contents)))
```

In this example, the keywords specify the following:

:numbered

The numbering template used in the chapter opening. For examples of numbering templates, see the section "Defining Counters for an Article" in *Symbolics Concordia*.

:referenced

The numbering template used in references to the chapter, for instance in the table of contents.

:title

The formatting specifications for the chapter opening. This is where the environment Hd1 is used.

:contents

The formatting specifications for the table of contents for chapter entries. This is where the environment Tc1 is used.

sage:define-box-type *box-type-name document-type device-type* &rest *options* &key *:left :top :right :bottom :horizontal :vertical :all* *Function*

Defines boxes to be used for boxing text in book designs.

sage:define-crossreference-appearance *appearance-name documentation booleans* &key *:formatter :editor* *Macro*

Defines the appearance of a crossreference link type.

appearance-name

The name for this appearance. It should be a symbol. This name (string-capitalized) will appear in appearance name menu you get when editing a link.

documentation

A string providing brief documentation for this link. It will appear when the mouse is moved over this link type in the menu.

booleans

A set of options to this link type. Each option is of the form *option* or (*option default* &key *:documentation :pretty-name*)

option is a symbol naming the option.

default is the default value. Since these are booleans, the default must be **t** or **nil**.

:documentation is a string providing brief documentation of the purpose of this option. It will appear when the mouse is moved over this option in the menu.

:pretty-name is a string which appears in place of this option name in the menu.

:formatter

A form whose **car** is an argument list and **cdr** is a functional body. The body can reference the variables in the argument list, as well as any of the booleans. The two arguments passed to this function are a record-group and a "contents-list".

So it is ((*record-group contents-list*) &body *body*)

The *record-group* is the particular record-group this link points to. The *contents-list* is a sage data structure which corresponds to the formatted name of that *record-group*.

Although **:formatter** is a keyword argument, it is required.

The formatter should do output to ***standard-output***. It is called within the scope of a **sage:sage-formatting**, so it can use things like **sage:sage-paragraph** or **sage:in-environment**. If it wants to output the formatted name of the target of the link, it can use **(sage:sage-contents-list t** *contents-list***)**.

The output from formatter is what appears in place of the link when the sage formatter runs. So the formatter for the See appearance outputs things like "See the ..."

:editor

A form whose **car** is an arglist and **cdr** is a functional body. The body can reference the variables in the arglist, as well as any of the booleans. The two arguments passed to this function are a record-group and a stream.

So it is ((*record-group stream*) &body *body*)

The editor should do output to stream. This is how the link displays itself in the editor.

If there is no editor, the formatter is run and its output is what shows up in the editor.

sage::define-device-type *name* *Function*
Defines a device type.

sage::define-document-type *name* *Function*
Defines a document type.

sage:define-line-type *line-type-name document-type device-type* &rest *options* &key *:weight :groove :halftone :character :rawfont* *Function*
Defines a line for use in highlighting text in book designs.

sage:define-sage-attribute *name argument-type environment-flavor form* *Function*
Defines a new environment attribute.

sage:define-sage-command *name arglist* &key *:document-type :device-type :definition :text? :starts-with-printing-char? :ends-with-newline? :ends-with-space?* *Function*
Defines a command to be used in a `.sab` file. Usually you do this in the course of creating a specialized document type. See the section "Creating a Letter Document Type" in *Symbolics Concordia*.

sage::define-sage-counter *name document-type device-type definition* *Function*
Defines a new counter type.

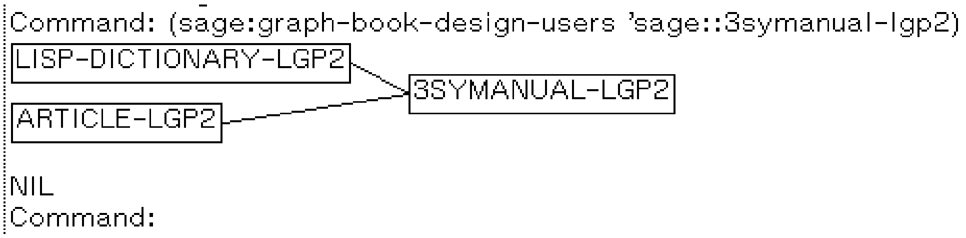
sage::define-sage-font *name document-type device-type definition* *Function*
Specifies a font to be used with the formatter for hardcopy output.

sage::describe-environment *envr* *Function*
Takes an formatting environment and displays its parameters and attributes. For example:

```
(sage::describe-environment 'sage::description)
The definition of SAGE::DESCRIPTION in
#<Directive ENVIRONMENT DESCRIPTION> looks like this:
     (BREAK :NONE)
     (SAGE::CONTINUE :NONE)
     (SAGE::ABOVE (1 SAGE::LINES))
     (SAGE::BELOW (1 SAGE::LINES))
     (FILL :NONE)
     (SAGE:LEFTMARGIN (+ 16 SAGE::CHARACTERS))
     (SAGE::SPACES SAGE::COMPACT)
     (SAGE::INDENT (- 16 SAGE::CHARACTERS))
     (SAGE::SPACING (1 SAGE::LINES))
     (SAGE::UNNUMBERED :NONE)
     (SAGE::PARAGRAPHBREAKS SAGE::NORMAL)
```

sage:graph-book-design-users *name* *Function*

Draws a graph of all the document designs that use the document design *name*.



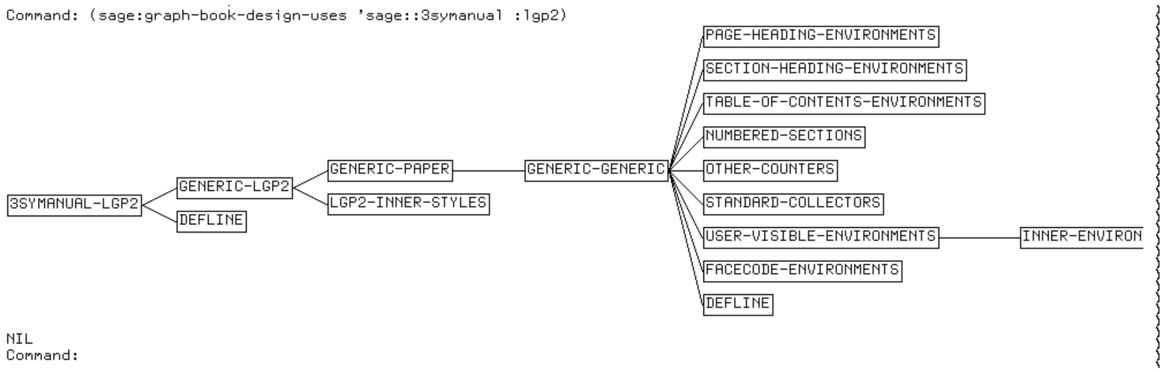
```
Command: (sage:graph-book-design-users 'sage::3symanual-lgp2)
```

```
NIL
Command:
```

Figure 20. **sage:graph-book-design-users**

sage:graph-book-design-uses *document-type device-type* *Function*

Draws a graph of the document designs from which the document design *document-type device-type* inherits.



```
Command: (sage:graph-book-design-uses 'sage::3symanual :lgp2)
```

```
NIL
Command:
```

Figure 21. **sage:graph-book-design-uses**

sage:idirs &body *body* *Macro*

Allows you to insert formatting directives for the formatting of titles in counter definitions, and for the formatting of counter contents for tables of contents, indexes, and pageheadings.

spaces — Accepts a number and puts in that many literal spaces. It uses the command **sage::literal-space**.

L — Accepts a text string and puts it in the LS character style, Lisp like typeface.

R — Accepts a text string and puts it in the roman character style.

envr — Accepts a sage environment and any necessary set up for placement of generated text. It uses the form **sage:in-environment**.

command — Accepts a sage command and evaluates it.

value — Accepts a counter or string with a value and retrieves the value.

lisp — Accepts a lisp form and executes it as a **progn**.

Here is a chapter counter, using **sage:idirs** to control how its title appears.

```
(chapter
  :within majorpart
  :numbered "@1."
  :referenced "@1"
  :title ((title counter-contents)
          (idirs
               (envr (HD1 above 2 below 2 facecode b)
                     counter-contents (spaces 2) title)))
  :contents ((title counter-contents)
             (render-regular-toc '(Tc1) title counter-contents)))
```

sage:*lgp2-default-character-style* *Variable*

Specifies the character styles to be used with a particular document type.

sage:making-sage-directives *(*&optional *stream)* &body *body* *Macro*

Binds *stream* to a special stream, which collects all formatting and output done in *...body...*, making Sage directive structure out of them, which it returns. It must run in a **sage:with-sage-device** dynamic environment.

sage:note-book-design-specifics *document-type device-type entry-name* *Function*

Associates a document type and a device type to produce a book design.

For example, 3symanual-lgp2 is produced by the form

```
(note-book-design-specifics '3symanual 'lgp2 '3symanual-lgp2)
```

sage::register-book *topic-string* &key *(:mnemonic "") :document-type :highest-structural-level :cover :remarks :sym-copy :doc# :effectivedate :releaseversion :marketing :mitcopy :authorgroup :design :cover-printer :text-printer :printer :confidential :doctrademarks :deferred-home :sage-variables* *Function*

Registers a Symbolics Concordia topic as a book and specifies its book design.

:document-type {**sage::3symanual**, **sage::approach**, **sage::reference-cards**, **sage::lisp-dictionary**} Specifies the document type.

sage::3symanual - The standard format of the Symbolics Document Set.

sage::approach - The new *Approachability* book design.

sage::reference-cards - The smaller format for the Symbolics Reference Cards.

sage::lisp-dictionary - The format for dictionaries, for example books 2B and 7B in the Symbolics Document Set.

:highest-structural-level
Tells the formatter the overall structure of the book. The default is to have Majorparts, subdivided into chapters (like *Symbolics Concordia* , for example). To specify that book is only one structural unit, use **:highest-structural-level 'sage:chapter**.

:sage-variables Allows you to set default document formatting directives. This is a list of variable/value pairs that sets Case selector variables used in the book. Using this option to **sage::register-book** is equivalent to using the Set Sage Variables command for each variable/value pair in the list.

For example,

```
(sage::register-book "My Book"
  :document-type 'sage::user-guide
  :sage-variables '((Country "USA")(Version "4.0")))
```

`"My Book"` uses the Case selectors, `Country` and `Version`. By default `Country` is set to `USA`, and `Version` is set to `4.0`. Note that you can override these document formatting directives when you format the book (use the **:query** keyword of the "Format Pages Page Previewer Command ").

Index