Table of Contents

List of Figures

1. 7.3 Ivory Concordia Installation Procedure

Ensure that you have installed Genera 7.3 Ivory and Genera 7.3 Ivory ECO #1 according to the installation instructions. (See the section "Genera 7.4 Ivory Software Installation Guide"and *Genera 7.3 Ivory ECO#1 Notes*.) Cold boot the MacIvory with a site-configured Genera 7.3 Ivory world containing the ECO patches. Then proceed as follows:

1. Boot a site-configured Genera 8.1 world.

2. Load the IP-TCP software:

    ```
    Load System IP-TCP
    ```

3. Restore the contents of the Symbolics Concordia tape, as follows:

    a. Place the Symbolics Concordia distribution tape in the tape drive.

    b. Issue this command to the Command Processor:

        ```
        Restore Distribution
        ```

    c. Enter a tape spec at the prompt:

        ```
        Enter a tape spec [default Local: Cart]:
        ```
        If you are using the cartridge tape drive on the local host, just press RE-
        TURN. Otherwise, enter the host name and the drive type.

    Load Symbolics Concordia, as follows (it takes approximately 20 minutes).

    ```
    Load System Concordia :Query No
    ```
    If you are loading Concordia on a 3600-family machine, you encounter an error that puts you in the debugger. You should simply resume. For more specific details, see the section "Loading Concordia on 3600-Family Machines" in *Symbolics Concordia*.

4. Decompress the Symbolics document database. The purpose of database compression and decompression is explained in "Documentation Database Compression and Decompression". You should refer to this section to decide if you should decompress the documentation database. This process takes up to 40 minutes. Use the function **sage:load-doc-database-for-writer**.

    ```
    (sage:load-doc-database-for-writer)
    ```

5. Reset the threshold at which the user receives warnings about address space being low. This is suggested because saving modified Symbolics Concordia buffers uses a lot of address space. The normal threshold for warnings is not adequate for protecting against running out of address space.

    ```
    (setq si:gc-warning-threshold 4000000)
    ```

6. Save the results of your installation. See the section "Customizing and Saving Worlds" in *Site Operations*.

7. Load the Lock Simple system to establish concurrency control for writing out Symbolics Concordia files. All hosts that are to store Symbolics Concordia files must agree on the server host to which they send their Lock-Simple requests. If the files to be locked are themselves resident on a particular host, it is a good idea to make that host a Lock-Simple server. If you use a server on another host, you have twice the risk that an unavailable host will make your files unlockable, preventing you from saving them. If you are storing Symbolics Concordia files on a MacIvory, the Concordia files must be stored in a LMFS, they cannot be stored in the MacFS. To install Lock Simple:

    a. Load the system called Lock Simple on each file server that is going to store Symbolics Concordia files.

        ```
        Load System Lock Simple
        ```

    b. Use the command Add Service to Hosts for each of these file servers to indicate that it is now a Lock Server:

        ```
        Add Service to Hosts LOCK-SIMPLE CHAOS-TOKEN-LIST LOCK-SIMPLE host-names
        ```

        See the section "Add Services to Hosts Command" in *Genera Handbook*.

    c. Do Enable Services LOCK-SIMPLE on each of the file servers that is now a Lock Server.

One host at your site should be designated as a surrogate Lock Server to process Lock-Simple requests for files on hosts that are not set up as Lock-Simple servers. To set up a surrogate Lock Server, use this procedure:

    a. Edit the namespace object for your site:

        ```
        Edit Namespace Object Site your-site
        ```

    b. Add the user property pair

        ```
        Lock-Simple-Server host
        ```

        where *host* is the server which will be the surrogate Lock server.

    c. Do Enable Services LOCK-SIMPLE on the surrogate Lock server.

2. New Features in Concordia on MacIvory

Illustrations drawn with MacDraw and MacPaint can be included in `.sab` files. You insert them using Create Picture and giving the picture type `Macintosh` and the pathname in MacIvory syntax, that is *Host: Volume: Folder: Folder: Filename.* See the section "Accessing the Macintosh File System" in *MacIvory User's Guide.*

Symbolics Concordia prompts you for the file format. If you specify `header` (get the format from the file header) then Symbolics Concordia takes care of figuring out what format (MacDraw or MacPaint) the file is in and inserts it just as it does pictures created with the Graphic Editor or Bitmap Editor.

MacDraw and MacPaint pictures look the same in Genera as they do in the Macintosh operating system, except, of course, for some minor differences associated with fonts. Symbolics Concordia tries to match fonts, but if you are using shadow fonts (which do not exist in Genera), for instance, a different font is substituted.

Note: Only MacDraw format files and MacPaint files can be inserted into Symbolics Concordia. MacDraw files in the `pict` format are not currently supported.

You can also read illustrations created with MacPaint into the Bitmap Editor for editing:

The Read MacPaint File Bitmap Editor Command

Loads a screen image file created with `COMMAND-#` (`COMMAND-sh-3`) or a file created with MacPaint into the Bitmap Editor. From there Save Image and Give Image To Graphic Editor are used to prepare it for insertion into a Graphic Editor picture with the Add Image Graphic Editor command . It is possible to place a MacPaint image into a Concordia record by just doing Create Picture and specifying picture type MacPaint. In this case, the source file for the image is stored on the Macintosh side only, there is no source file in the Genera file system.

A new command has been installed so that you can change the view of a single record using the keyboard instead of clicking `s-sh`-Middle on the record:

Change Record View

Change Record View
           Changes the view of the current record.

Click on	*To display*
All fields	The entire record. This is the default.
Contents text	Text and markup diagrams in the Contents field, plus the record header.
Outline	The names of the fields and any markup diagrams in the Contents field; no text.
Record header only	The header and trailer delimiting the record; displays the record name and type.

To abort the command, move off the menu. Reinvoke the command to return to the original view.

Enhancements have been made to two commands:

The Find Table of Contents command in the Symbolics Concordia editor has been renamed to Show Table of Contents and enhanced to accept a numeric argument (c-U) that sends the table of contents to a printer, analogous to c-U s-P formatting a topic for a printer. You should use Show Table of Contents in place of Find Table of Contents. The name Find Table of Contents will disappear in a future release.

When you specify the Graphic-Editor picture type to the Create Picture command and give it a file name, pressing HELP now lists the pictures in the file. The list is mouse sensitive and completion works on the picture names.

New Procedure for Picking Up Screen Images

Instead of pressing FUNCTION-s-Q or FUNCTION-s-sh-Q to pick up a screen image, in Genera 7.4 Ivory you press FUNCTION-0-Q. A menu comes up that allows you to specify the destination for the screen image and the portion of the screen to capture. You select the appropriate options and then click on Done. Subsequent uses of FUNCTION-Q use these options. To change the options, press FUNCTION-0-Q again.

3. Apple Bug That Affects Concordia

Due to a bug in the Apple Macintosh, the left `SUPER` and `SHIFT` key chord does not work for the number keys. This means that if you are using Symbolics Concordia on a MacIvory, commands like `s-sh-^` (Remove Markup) do not work. The right `SUPER` and `SHIFT` combination does work. This is true for both the Apple keyboard and the Symbolics keyboard. Use the righthand `SUPER` and `SHIFT` keys for `s-sh-^` and other `s-sh-` commands.

Additionally, the chord `s-SELECT` does not work. Click on the icons in the upper righthand corner of the Symbolics Concordia window to change Concordia activities.

4. New Concordia Documentation

4.1. Internal and External Markup

Additional markup commands and environments are available for use primarily in book design. (See the section "Book Design Functions Dictionary" in *Symbolics Concordia*.) Many of these *internal* commands and environments have hyphenated names and, since they are intended for use in book design, they cannot be inserted in a `.sab` file using Create Markup. However, sometimes it is necessary to use one of them. If you need to insert an internal command or environment in a `.sab` file, you can insert it using `c-U s-M`.

4.2. NewPage Markup Command

Begins a new page immediately. Its argument is an integer that controls the number of blank pages to leave (the default is 0).

4.3. Converting Documents From Concordia to Scribe

Symbolics Concordia can output Scribe `.mss` files. This is particularly useful if you need to submit a paper, for example, to someone who wants it in machine readable format but does not have Symbolics Concordia.

To create a Scribe `.mss` file, you go to a Lisp Listener and use the Command Processor command Convert Topic To Scribe.

Convert Topic to Scribe

Convert Topic To Scribe *documentation-topic pathname keywords*

Reads *documentation-topic* and writes a Scribe `.mss` file (*pathname*) that contains Scribe formatting directives to allow the topic to be formatted with Scribe.

documentation-topic The topic to convert.

pathname   The pathname of a file to put the Scribe source in.

keywords   :Output Destination, :Process Unconvertables

   :Output Destination
                      {Buffer, File, Kill Ring, None, Printer, Stream, Window}
                      Where to redirect the typeout done by this command. The de-
                      fault is the stream ***standard-output***.

:Process Unconvertables

> {Yes, No} Whether to convert unconvertable Symbolics Concordia constructs into simple strings or signal an error. The default is No.

4.4. Creating Active Examples

Active examples are pieces of Lisp code that are run when the documentation is displayed and produce output. They are mouse sensitive and can be run or their code edited by readers from Document Examiner.

To create an active example, use the following steps:

1.   Create an ActiveExample markup.

2.   Type the code that belongs in the example.

3.   Use m-X Create Example Record Marker at the end of the code.

4.   Use m-X Test Example to run the example and save the results.

An example may produce text output, graphics output, and return values.

m-x Create Example Record Marker inserts a *record marker* in the example. The record marker saves the results of running the preceding Lisp form. The command prompts you for the type of record marker to create.

You can create the following types of record markers:

Typescript   Saves characters output to ***standard-output***.

Picture   Saves the graphics output to ***standard-output***. This does not capture text, unless it's drawn with **graphics:draw-string**.

Values   Saves the values returned by the forms in the example.

Bitmap   Saves the values as a bitmap.

m-X Test Example runs an active example and saves the results of running the example in the record markers embedded in the example.

Markup:

```
┌─────────────┐
│ActiveExample│
└─────────────┘
        (print "FOO BAR BAZ")       ; the output is:

        ⊕ Example record typescript

        (graphics:with-room-for-graphics (t 100)
            (graphics:draw-arrow 20 20 70 70 :thickness 2))

        ;;; The graphics output is:

        ⊕ Example record picture

        (+ 20 30)                    ; the value is:

        ⊕ Example record values
┌─────────────┐
│ActiveExample│
└─────────────┘
```

Display:

```
(print "FOO BAR BAZ")       ; the output is:

"FOO BAR BAZ"

(graphics:with-room-for-graphics (t 100)
    (graphics:draw-arrow 20 20 70 70 :thickness 2))

;;; The graphics output is:




(+ 20 30)                   ; the value is:

50
```

4.5. Creating Mouse Sensitivity in Your Output

Crossreference links, precis links, and Lisp objects are all mouse-sensitive in your output. In addition, you can make any text mouse-sensitive by using the Invisible appearance of a crossreference (see the section "Crossreference Link" in *Symbolics Concordia*).

You can also use the "Presentation Environment" to introduce arbitrary presentation types, allowing you to create mouse-sensitive items that perform any tasks you can program.

For example, if you were to display this section in a Lisp Listener using the Show Documentation command, the small picture of *Symbolics Concordia* below, when clicked on, would cause *Symbolics Concordia* to be selected.
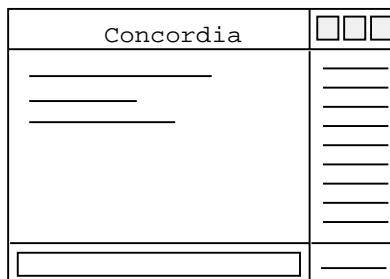

Figure 1. Click on me to select *Symbolics Concordia*.

This is because the picture of the Symbolics Concordia frame is surrounded by a Transparent environment that has the following additional attributes:

```
PresentationObject: (CP:BUILD-COMMAND 'SI:COM-SELECT-ACTIVITY "Symbolics Concordia")
PresentationType: 'CP:COMMAND
```

The PresentationObject associates the picture with the Command Processor command Select Activity, and the picture is now a *presentation* of type **cp:command**. (See the section "The Presentation Type System: an Overview" in *Programming the User Interface*.)

Note, however, that clicking on this picture does not select Symbolics Concordia if you do it in "Document Examiner", because the "Select Activity" command is in the Command Processor's command table but not in "Document Examiner"'s.

To install the "Select Activity" command in "Document Examiner", you must add it to Document Examiner's command table. We can do that here using an ActiveExample:

```
;; The way to determine what symbol names the function
;; that implements a particular command is to do (cp:read-command)
;; and type the command name.  You'll get the symbol
;; back as a value.

NIL
(cp:command-table-install-command "Doc-Ex" 'si:com-select-activity)
```

You can click on this example to run the code. First, you can try looking up this documentation section in "Document Examiner" to verify that the picture above is not mouse-sensitive. Then run this example and click on the picture again to verify that it has become mouse-sensitive.

The two environment attributes, **PresentationType** and **PresentationObject** can be added to any environment (or a Transparent environment if you do not want any particular environment). Each of these should be set to a Lisp form that, when evaluated, produces a presentation type or a piece of data, respectively. If you know the presentation type to be a constant, you should quote it with "'". If you do

not specify a PresentationObject, the internal Sage datastructure becomes the object.

See the section "Dictionary of Predefined Presentation Types" in *User Interface Dictionary* to use presentation types that are already defined in Genera. If you want a new kind of presentation, see the section "Defining Your Own Presentation Types" in *Programming the User Interface*.

You can create mouse sensitivity for parts of pictures, also. See the section "Making Pictures Using Lisp" in *Symbolics Concordia*.

4.6. Insert Graphic Editor Drawing Code

Inserts the Lisp code that draws a drawing into an editor buffer. It prompts you for a drawing loaded into the Graphic Editor.

You can then edit the code, or locate and modify specific entities. Here is a simple drawing:
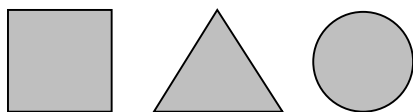
Using Insert Graphic Editor Drawing Code (m-X) inserts the following in your editor buffer:

```
;;; Drawing insert-drawing-code from SAP:>sys>graphic-editor>insert-drawing-code-example.pic

.1

(DEFUN DRAW-INSERT-DRAWING-CODE (&OPTIONAL (*STANDARD-OUTPUT* *STANDARD-OUTPUT*))
(GRAPHICS:WITH-SCAN-CONVERSION-MODE (T :SKETCH NIL)
    ;; Rectangle-6
    (PROGN (GRAPHICS:DRAW-RECTANGLE 158 438 212 385 :GRAY-LEVEL 0.25)
           (GRAPHICS:DRAW-RECTANGLE 158 438 212 385 :FILLED NIL))
    ;; Triangle-3
    (PROGN (GRAPHICS:DRAW-TRIANGLE 267 438 234 385 301 385 :GRAY-LEVEL 0.25)
           (GRAPHICS:DRAW-TRIANGLE 267 438 234 385 301 385 :FILLED NIL))
    ;; Circle-2
    (PROGN (GRAPHICS:DRAW-CIRCLE 343 411 26 :GRAY-LEVEL 0.25)
           (GRAPHICS:DRAW-CIRCLE 343 411 26 :FILLED NIL))))
```

Presentations can be added to entities, to create mouse sensitivity in parts of the picture. See the section "The Presentation Type System: an Overview" in *Programming the User Interface*.

The modified drawing should be saved in a Lisp file and can then be inserted into a Symbolics Concordia buffer using the Create Picture command and the picture type Lisp.

5. Updates to the Printed Concordia Documentation

5.1. Distributing Documentation Systems

When you distribute a documentation system with your application software, there are two documentation specific issues that you should know about.

- If you are distributing documentation to a site that is running Genera 7.2, you must distribute NSage patches.

- If you define any customized formatting environments or book designs, you must distribute them or your customers will not be able to read your documentation.

5.1.1. Distributing NSage Patches

In addition to your own software (application program and documentation system), if you are distributing documentation to a site running Genera 7.2, you must also distribute the binaries for the additional NSage patches that came on your Symbolics Concordia tape and the NSage patch on the Symbolics Concordia 1.0 ECO #1 tape distributed with this document. These patches are classified as part of development Genera and can, therefore, be distributed under the new Patch Exchange Policy announced to SLUG January 25, 1989 via mail (a copy of which is attached). Please note that distributing these NSage patches is not necessary for sites running only Genera 7.3 Ivory or later releases.

Restore the contents of the Symbolics Concordia 1.0 ECO #1 tape with Restore Distribution. Then load patches for NSage.

To distribute your software, you should use the Distribute Systems Frame and add NSage to the systems you distribute. See the section "Distribute Systems Frame".

Using the Distribute Systems menu frame, with initial default parameters, the NSage system spec should include:

```
NSage, version 27
   Distribute-sources         No
   Distribute-binaries        No
   Included-files-checkpoint   7.2
```

Be sure to mention in your installation instructions that after restoring the distribution tape, the user must load NSage patches.

5.1.2. Distributing Book Designs

If you define a new document type or book design that you want to have available online, or if you define your own special-purpose formatting commands or environments that you use in your documentation system, you must be sure to include these in your documentation system and distribute them with your documentation system.

Note: The Letter, Article and Memo book designs are defined in Symbolics Concordia, not NSage. They exist for the convenience of users of Symbolics Concordia who want to use Concordia for all their text handling work. If you use these document types for your online documentation, you must be sure to include the definitions for them and their commands as a module in your documentation system and be sure they are included in your distribution.

5.1.3. Quality Assurance for Documentation Systems

We recommend that before distributing a documentation system, you do installation and use testing.

Before sending out a documentation system, you should configure a machine at your site with the software that you expect your customers to be using (without Symbolics Concordia) and load your documentation system and the NSage patches. Then use Document Examiner to look at your documentation.

Check that your documents appear in the candidates pane of Document Examiner. If they do not, check your **sage::register-book** forms against the top-level records to make sure the titles are correct.

Use Show Documentation on records that make use of any features such as a document type or screen book design you have designed yourself. Browse through your documentation, displaying a variety of records to make sure that everything you expect to have included is there.

5.1.4. Replacing Symbolics Documentation with Your Own Documentation System

When you package an application, you might want to prevent your users from seeing any documentation system except for the one that you provide. You can remove the Symbolics Documentation System from a distribution world you are building for your customers by using **sage::clear-record-registries**. It removes all the index information currently loaded in your world. Then you can load your own documentation system, which is the only information available in Document Examiner.

sage::clear-record-registries   *Function*
> Zeros out all the unique-id registries, the keyword token indices, and the ***topic-aarray***. Returns the message "Your NSage doc system has been cleared". This removes all index information and record information currently loaded in your world.

5.1.5. Distributing Documentation for Macintosh Document Examiner

Overview   You can deliver any documentation created with Symbolics Concordia in Document Examiner on the Macintosh. You need only turn it into a Document Set.

You first determine the topics that you want to include. After you determine the topics, you create a Macintosh Document Examiner Document Set.

You can designate an existing record as a Document Set, for example, a chapter, a section, or an entire book. Additionally, you can designate a Document Set by creating a new record and including all records pertaining to your topic. After you specify a record to serve as the top level record of the Document Set, you use the Write Producer Files command from a Lisp listener to make the Document Set available on the Macintosh desktop.

DocSet Size   Unlike traditional Macintosh documents, you can create a Macintosh Document Set as slarge as you want. For example, you could create a Document Set including all 10,000 pages of the Encyclopedia Britannica. The Write Producer Files command automatically places all 10,000 pages of this book into one document on the Macintosh desktop as in Figure 2.
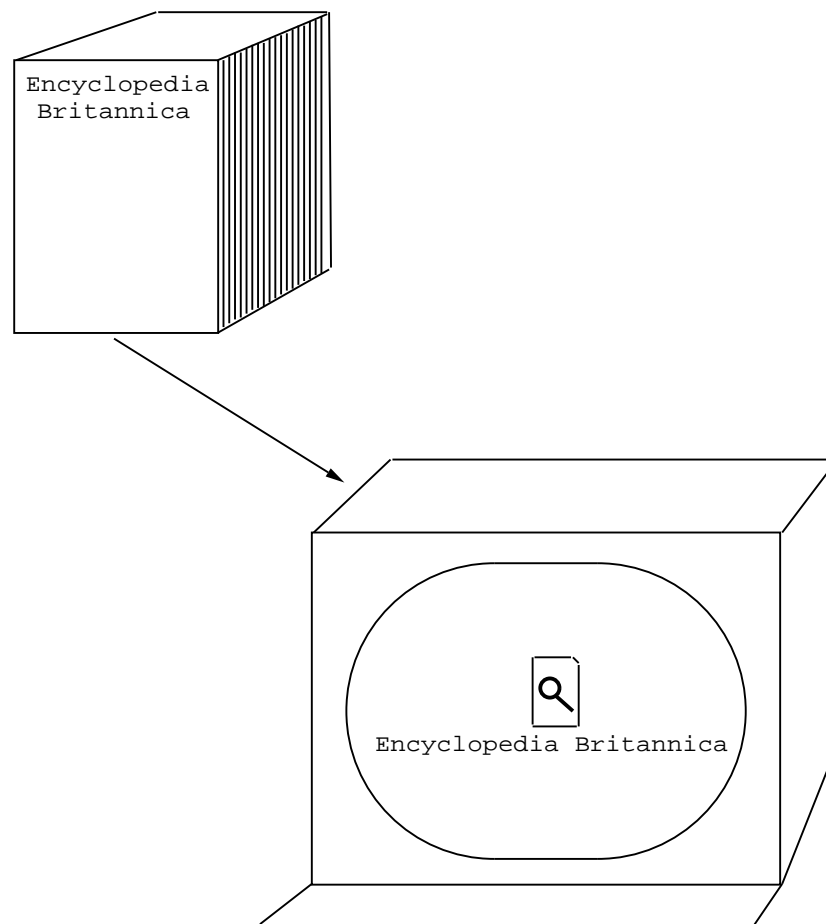
Figure 2. Document Sets.

Requirements   You can create a Macintosh Document Examiner Document Set
                      and read documentation online using a MacIvory with these
                      features:

                      • Eight Megabytes of Nubus memory.

                      • 320 Megabytes of disk space.

                      • The Concordia hypertext documentation system.

Procedure   Follow these steps for creating a Document Set:

                      1.   Make sure you are using a MacIvory running Concordia.

                      2.   Restore the Producer system to your SYS host by insert-
                           ing the Producer disk and specifying RESTORE DISTRIBUTION
                           :USE DISK FLOPPY. At the pathname prompt, specify
                           HOST:PRODUCER:DISTRIBUTION.VOLUME-1.

                      3.   From a Lisp listener, load the Producer system using the
                           Load System command.

                      4.   Use the Write Producer Files command from a Lisp lis-
                           tener for creating a Macintosh Document set. Specify the
                           Document Set name, the top level record of the Document
                           Set, and the Macintosh directory pathname in which the
                           Document Set will reside. For example, you can create a
                           Document Set named Encyclopedia Britannica and place in

```
the host:disk:DEX-Help directory.
      Command:  Write Producer Files (DocSet Name)
         "Encyclopedia Britannica"
         (a documentation topic) Encyclopedia Britannica
         (a directory pathname) Host:Disk:DEX-Help:
```

                      You have to perform the first three steps only when creating
                      your first Document Set. When creating subsequent Document
                      Sets, perform step 4 only.

Note   If many users view the same Document Sets, you can use the
                      Appleshare server for storing Document Sets.

Write Producer Files

Write Producer Files *DocSet-Name Documentation-Topic Directory-Pathname key-
                      words*

DocSet-Name   The name of your Macintosh Document Set. The name that you specify identifies the Document Set that will contain your documentation on the Macintosh. For example, you can create a Document Set named Encyclopedia Britannica. When you complete the Write Producer Files command, a Document Set icon named Encyclopedia Britannica appears on the Macintosh desktop. **Note**: Document Examiner only accepts Document Set names under 32 characters long.

Documentation-Topic
                       A topic including all records that you want to read online on the Macintosh. For example, specifying Encyclopedia Britannica automatically creates a Document Set containing all records included by the topic Encyclopedia Britannica.

Directory-Pathname The Macintosh folder in which your Document Set will reside. For example, you can place the Encyclopedia Britannica Document Set in a folder named DEX-Help by specifying *Host:Disk:DEX-Help:* where *Disk* corresponds to the name of your hard disk, and *DEX-Help* corresponds to the name of the folder in which you place your document set.

keywords   :Format, :Preload-Documentation, :Documentation-Version

                       :Format   {Macintosh, Lisp-Machine} The format of your document set. The default is Macintosh.

                       :Preload-Documentation
                                               {Yes, No} Preloading documentation enables you to load all records necessary for creating a Document Set at one time. For example, if your Document Set contains two hundred records, using this option enables your machine to read all records for the Document Set during one network connection. The default is Yes.

                       :Documentation-Version
                                               {*number*} The version number of the Document Set you are formatting.

                       Specifying HOST:DISK:DEX-Help: as the directory pathname for the Write Producer Files command enables you to place your documentation into a folder named DEX-Help as in figure 3.

Considerations   You can store documentation either on your local Macintosh or on a remote Macintosh server. Store your documentation on the machine enabling you to read the documentation most efficiently.
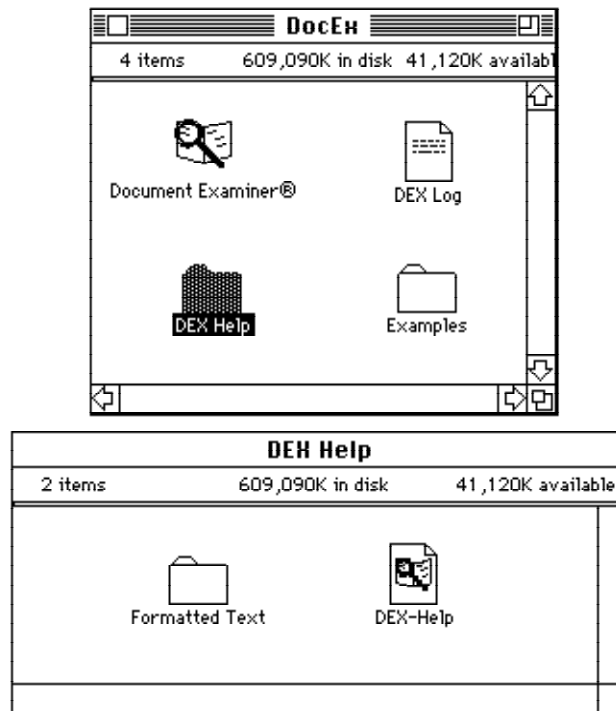
Figure 3.   The Macintosh desktop.

5.2. Documentation Database Compression and Decompression

5.2.1. What is Documentation Database Compression?

Online documentation resides on the disk in binary files. The indexing information for searching and lookup is always loaded into the world. In a world including Symbolics Concordia, the index information for the documentation records (unique ids and so on) is large, because each record may have several versions and other information associated with it. This information is organized by a structure called a *record group*. Each record can have a version of itself that is a *published record* (patched or compiled into the database) and an *edited record* (one that has been changed but not installed or patched yet). Symbolics Concordia keeps careful track of each version of the record in the record group. This adds quite a number of blocks to the world size. The majority of the information cached in a record group is of interest only when editing documentation. For users who aren't editing documentation, it is definitely advantageous to have all vestiges of anything other than the published record removed.

During the compilation of the Symbolics Documentation Database (see the section "Compiling" in *Symbolics Concordia*), the index information loaded into the world

is *compressed*. This means that all the .sab files included in the new version are loaded and the compiler removes the "group" from each record in the record registry. When this compression is finished, the information that was removed is stored on disk in two index files, SYS:DOC;INSTALLED-N;COMPRESSED-INDEX-A.BIN, and SYS:DOC;INSTALLED-N;COMPRESSED-INDEX-AW.BIN. where *n* is the version number of the new documentation database. Finally, a copy of each .sab file in the documentation system is written out to this installed-*n* directory. This provides a small directory of only those versions of the documentation files included in the current documentation database for lookup in Document Examiner.

CAVEAT: At this time there can be only one compressed documentation database in a world. The Symbolics documentation is it.

5.2.2. Why Decompress the Documentation Database?

You decompress the documentation database to be able to make crossreferences to Symbolics documentation and so that any documentation systems you create can be patched (incrementally updated; see the section "Patching a Documentation System" in *Symbolics Concordia*).

When someone makes crossreferences to Symbolics documentation records, the complete record group information for those records must be made available. Similarly, when a documentation system is patched its complete record group information must be added to the index information in the world. This means that the structure to hold record group information must be in place in the world. *Decompression* is the process that restores the structure, which was removed when the documentation database was compressed.

5.2.3. Can Decompressing the Documentation Database Be Skipped?

Most users of Symbolics Concordia should decompress the documentation database. However, if you are never going to make reference to Symbolics documentation in your own documentation, or to create a documentation system of your own to which you add patches (incrementally updates, see the section "Patching a Documentation System" in *Symbolics Concordia*), you can skip the decompression step. This has the advantage of making your incremental world a little smaller. If you want the option to have your documentation refer to Symbolics documentation, or are planning to create your own patchable documentation system, you should decompress the database.

5.2.4. Decompression Procedure

After Symbolics Concordia is loaded (see the section "Symbolics Concordia Installation Procedure", page Symbolics-Concordia-Installation-Procedure-SECTION), run the function **sage:load-doc-database-for-writer**. This process takes up to 40 minutes. It uses the information stored on disk after the compression to construct a structure to hold the record groups. The record groups themselves are loaded in as a record is edited or reinstalled (temporarily or permanently by patching). The addition of this record group structure makes the world larger by about 10,000 blocks.

sage:load-doc-database-for-writer &key *(:system-name* **"DOC"***)*   *Function*
    (:major-version **sage::*doc-system-version****)*
    Reverses the compression process done to the documentation database dur-
    ing recompilation. To do this, it loads the files `compressed-index-a.bin` and
    `compressed-index-aw.bin` from the installed documentation database directo-
    ry and reconstructs the record group for each record in the documentation
    system. This allows tracking the edited and installed versions of each
    record for writers.

5.3. Creating Simple Tables

The SimpleTable environment allows you to create simple tables. It is an unfilled
environment, so each line in the table corresponds to a line in your buffer.
Columns are separated by tabs. The number of columns (as determined by the tabs
in each line) and how much text is in each column, are used to calculate column
width based on the longest lines in the table. Tabstops are set for you by the Sim-
pleTable environment. You can change character styles (bold, italic) but you cannot
use `s-L` markers or links inside a SimpleTable environment.

Note: Simple Tables do not display correctly in the Page Previewer. They take up
the appropriate amount of space on the page, but because the Page Previewer does
not scale fonts, the text appears to overlay the rules. They come out correctly in
hardcopy.

The SimpleTableSpecs command specifies horizontal and vertical rules for a simple
table.

The possible horizontal rules start above the first line of the table (position 0).
The possible vertical rules start to the left of the first column (position 0). You
must specify each rule you want explicitly (remember, this is a SimpleTable), so if
you have a three column table that is three lines long, and you want rules around
all sides and between each line and column, you would specify:

```
☞SimpleTableSpecs Hrules 0,1,2,3; Vrules 0,1,2,3
```

You enter the SimpleTableSpecs with `s-M`.

You probably should not make rules around an entire SimpleTable because of the
way that the individual "cells" of the table work. Each table entry lives in a cell.
The cells have a border around them and the rules are drawn centered in the bor-
der. This is fine for the interior walls of cells, but if you want your corners on the
outside to meet it won't work. Here's an upper left corner cell with rules at 0,0:
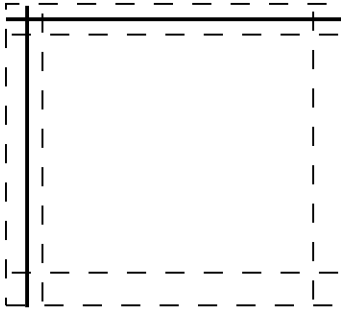
Markup

```
Simpletable
      ☞SimpleTableSpecs Hrules 1; Vrules 1,2
      thing described⊢ its discussibility⊢ what it means
      :this⊢ at hand⊢ what this means
      :that⊢ remote⊢ What that means
Simpletable
```

Display

thing described	its discussibility	what it means
:this	at hand	what this means
:that	remote	What that means

Empty cells are ignored, so if you want a gap in your table (say, if you wanted "at hand" above to be a blank), you would place a space between the two tabs in the input.

You can omit the SimpleTableSpecs and have no rules. However, if you supply SimpleTableSpecs you must supply at least one vertical and one horizontal rule. You do not have to specify a continuous series of rules. For example, if you have an item in your table that runs onto two lines, you can skip that position in the horizontal rules so that both lines of the item end up in the same "box". This SimpleTable has horizontal rules at positions 1, 2, and 5:

This	Here
That	There
The Other Thing That Takes More Than One Line	Way Over Yonder
Another One	Nearby

5.4. Clarifications and Hints for Concordia and the Graphic Editor

- If you have a renamed record, and a version of that record with its old name is read into your machine (perhaps because you need to refer to an old version of a file for some reason), the record appears to revert to its old name. This can be very confusing, but it is only a local confusion in your environment. You can straighten it out by using Rename Record again. (It is not necessary to patch this re-renaming, since it is only a local phenomenon in your environment and does not affect any other users.)

- Sometimes figures have added space above or below them when they appear in text. Most likely there is a small, degenerate object that is invisible somewhere above or below the main part of the drawing. In the Graphic Editor the Fit View command will place the entire picture in the viewing pane. If there is un-wanted white space above or below (or on one side) of the picture, try selecting the unwanted region. This causes the *handles* of any invisible objects to become visible, and then Delete can be used to remove them, making the picture fit the space.

- When you create a markup command or environment for a book design, do not use a hyphen in the command or environment name. Hyphenated names are for internal commands or environments, and you cannot insert such an environment or command in a .sab file using Create Markup. You must use c-U s-M to insert them.

- Explicit line breaks are not recommended in filled text, but occasionally you need them. The format directive Force-Line-Break cannot be entered using s-M (Create Markup). You can insert one, however, by doing the following: Put @* wherever you want a Force-Line-Break, mark it and do m-X Parse And Replace Region. That inserts the untypeable command.

- There is a known bug that affects pathnames of pictures. Even if you are care-ful to specify a logical pathname when you insert a picture, Symbolics Concordia translates the logical pathname to a physical pathname. This causes no harm for formatting or display, but you should be aware of it when maintaining your doc-umentation system.

- The Modify environment is listed in the menu of environments (when you press `HELP` after Create Markup). However, it is an internal environment and is not for use in `.sab` files. It should be used only in book design.

Index