

* Software Modules

** Storyboard Interpreter

The storyboard interpreter reads in and parses the fields in a storyboard file, and drives the HyperText machinery based on the commands and text therein.

*** Fields of a Storyboard

**** Title Field

A storyboard starts with the line ".title", followed by the storyboard's title on the next line.

**** Synonym Field

Starts with the line ".synonyms" (or ".synonym").
Followed by the synonyms of the storyboard, each on its own line.
The synonym field is optional.
It ends with the beginning of the .definition field.

**** Definition Field

Starts with the line ".definition".
It contains text and formatting commands that describes the definition image.
It can have text, font changes, pictures and targets, like the .contents field, but it should not generally have reference buttons in it.
This field extends to the beginning of the .contents field, or to the end of the file, if the storyboard has no contents.

**** .contents

.<command>

Formatted text

* Page Formatter

Files: fmt.c fmt.cps fmt.ps

The storyboard interpreter drives a page formatter, whose job it is to decide where things go on the page, and to describe the pages by calling remote PostScript procedures in the NeWS server.

These PostScript procedures describe and manipulate windows and pages.

A HyperTIES window is comprised of a page for the control panel, a page for the definition window, and zero or more pages for the storyboard contents.

The formatter describes a page by building a display list of PostScript code associated with the page in the NeWS server. NeWS executes the display list to draw the page's image.

It puts code to change fonts and show lines of text into the display list. It also creates stamp-pad objects, used to put picture drawing code into the display list, and to render buttons and targets onto the page.

** Arguments:

*** Instantiation (create a stamp-pad)

Stamp: object body, class, name, instance ID, filepos

Target: object body (args, init), class, name, instance ID, filepos

*** Display (put a stamp)

Stamp: instance ID, position

Target: instance ID, position, size

String: string, position

* Compiling Storyboards into Forth

The storyboard interpreter is capable of compiling storyboards into Forth programs that call low level formatting commands to describe pages to the NeWS server.

The interpreter compiles storyboards by formatting them, and printing Forth words to a text output file, capturing the execution of the low level formatting commands in forth function definitions.

HyperTIES can subsequently read in the resulting Forth functions, and compile them into memory. Then they can be efficiently executed, to produce identical pages as the storyboards interpreted from disk, without the overhead of the formatter reading the storyboards and laying out the page.

The memory image of the HyperTIES Forth system, loaded with all the compiled storyboards of a database, can be saved out to disk, so that later sessions can restore the environment. This significantly improves the response time for displaying definitions and contents, without noticeably increasing the startup time.

* Objects

In addition to the storyboard files (with .st0 suffixes), there exist picture files (with .pn0 suffixes) and targets files (with .tn0 suffixes), describing the objects referred to by the author in the storyboard files.

You can think of the objects described in the .pn0 and .tn0 files as stamp pads. When an object is referred to in a storyboard, it is used to stamp a picture or a target down at some location on the page.

** The .pn0 and .tn0 files contain an object's class, name, and body.

*** class

An object's class defines its behavior.

Currently, the class is the name of a PostScript class defined in the NeWS server. In the future, some classes may be totally or partially implemented in the HyperTIES client, in C or C++.

*** name

In the storyboard files, an author can refer to an object by its name. The master index maps from the object's name to its location in a .pn0 or .tn0 file.

*** body

The interpretation of an object's body, determining its characteristics, is up to its class.

For all the currently implemented classes, the object's body contains PostScript code to be interpreted by the NeWS server. The server executes the body, which pushes onto the stack the parameters for the /new message, sent to the class implemented in the NeWS server, to instantiate the stamp-pad object the first time it is referenced.

* Data Structures

```

** index
  3 namespaces of index entries
  storyboard names (.st0)
  picture names (.pn0)
  target names (.tn0)
** entry
  name
  file, offset, length
  instance header
    instance ID
    class
    width, height
  NeWS StampDict entry
    instance ID
    object
** pile
  PileDict
** Page
  PageDict
  Canvas
  EventMgr
  DL
  Targets
** Target
  TargetDict
  TID
  Object
  Ref

* Classes
** Stamp Classes
*** Stamp
**** Methods
    /new % name id filepos => Obj
    /compile % SubClassResponsibility
    /size % - => width height
    /dlist % {PostScript code ...} => PostScript code ...
**** Instance Variables
    /Name
    /ID
    /FilePos
    /X
    /Y
    /Width
    /Height
*** Picture / Stamp
**** Methods
    /new % source width height name id filepos => obj
    /compile % x y => x y <PostScript code ...>
**** Instance Variables
    /Source
*** UnitPicture / Picture / Stamp
    /compile % x y => width height x y gsave translate scale
        <PostScript code ...>
        grestore
*** ScaledRaster / Picture / Stamp
    /new % canvas width height name id filepos => obj
    /compile % x y => width height x y gsave translate scale

```

```
        <canvas> imagecanvas  
grestore
```

```
*** Raster
```

```
*** TargetStamp
```

```
** Target
```

```
* Data Files
```

```
** master-index
```

```
** Storyboard file -- .st0
```

```
** Picture file -- .pn0
```

```
** Target file -- .tn0
```

```
* Source Files
```