

global/Font
Current font.

global/FontName
Current font name.

global/FontSize
Current font size.

global/TiesRootDirectory
String containing root directory of TIES database.

global/FilePos
Array containing [directory file-name offset length] of the last object defined.

global/LastTID
TargetID of last Target defined.

global/LastSelectionTime
Last time a reference button was selected. (UpTransition)

global/Pile
The current Pile dictionary. (also on the dict stack)

global/Pile/PileID
local/PileID
The pile's PileID

global/Pile/PileID
local/ParentPileID
The pile's parent's PileID

global/Pile/Win
local/Win
The pile's Window object.

global/Pile/Win/WinPileID
local/Win/WinPileID
The PileID of the Window's pile.

global/Pile/Win/PageTrackers
local/Win/PageTrackers
A dictionary mapping the TargetID of every target on this page tracking PageNumber to its reference. A /PageTrack message is sent to each of these targets when the page number changes.

global/Pile/Win/Pages
local/Win/Pages
Array of canvases of Pages in this Pile.

global/Pile/Win/PageNumber
local/Win/PageNumber
Index into Pages of currently displayed page. (-1 if none)

global/Pile/Win/PageDict
local/Win/PageDict
Dictionary mapping all this Pile's canvases to Pages.

```

    canvas => Page

global/Pile/Win/PageDict/<canvas>
local/Win/PageDict/<canvas>
local/Page
    Page dictionary that keeps track of one canvas.

global/Pile/Win/PageDict/<canvas>/Can
local/Win/PageDict/<canvas>/Can
local/Page/Can
local/Can
    Canvas corresponding to this page. Same as <canvas>.

global/Pile/Win/PageDict/<canvas>/EventMgr
local/Win/PageDict/<canvas>/EventMgr
local/Page/EventMgr
    Process that handles start interests of targets on canvas
    and repairs damage on canvas by executing display list.

global/Pile/Win/PageDict/<canvas>/DL
local/Win/PageDict/<canvas>/DL
local/Page/DL
    Display list that draws this page. Executed by EventMgr.

global/Pile/Win/PageDict/<canvas>/Targets
local/Win/PageDict/<canvas>/Targets
local/Page/Targets
    Dictionary mapping TargetID's of all targets on this page to
    their references. Currently only the TargetID's are actually used.

global/Pile/PageWidth
local/PageWidth
    The width of each page in this pile.

global/Pile/PageHeight
local/PageHeight
    The height of each page in this pile.

global/Pile/Can
local/Can
    The canvas of the pile's current page.

global/Pile/Page
local/Page
    The pile's current Page dictionary.

global/PileDict
    Dictionary mapping all PileID's to Pile dictionaries.
    PileID => Pile

global/PileDict/<PileID>
    Pile dictionary.

global/StampDict
    Dictionary mapping InstanceID's to stamp-pad objects.

global/StampDict/<InstanceID>
    Stamp-pad object, subclass of Stamp.

```

global/StampDict/<InstanceID>/FilePos
The FilePos where the definition of this Stamp lives.

global/StampDict/<InstanceID>/Name
A Stamp's name.

global/StampDict/<InstanceID>/ID
A Stamp's Instance ID.

global/TargetDict
Dictionary mapping TargetID's to Target objects.

global/TargetDict/<TargetID>
Target object, subclass of LiteItem.

global/TargetDict/<TargetID>/TID
A Target's TargetID.

global/TargetDict/<TargetID>/Name
A Target's name.

global/TargetDict/<TargetID>/Ref
A Target's reference.

global/TargetDict/<TargetID>/TargetPileID
The PileID of the pile the target lives in.

global/TargetDict/<TargetID>/ID
The InstanceID of the stamp that created this target.

global/TargetDict/<TargetID>/FilePos
The FilePos where the definition of the Stamp that created this target lives.

global/RasterDict
Dictionary mapping raster file name to cached canvas.

global/RasterDict/<RasterFileName>
Cached canvas read from RasterFileName.

Rename:

FmtWin => TIESWindow
WinDict => PileDict
CanvasDict => PageDict
CanDict => Page
ImageDict => StampDict
Image => Stamp
Target => TargetStamp
Canvas => Raster
ScaledCanvas => ScaledRaster
Touchable => Target
TouchableMenu => MenuTarget
TouchablePageControl => PageTrackerTarget
TouchablePopup => PopupTarget
TouchScrollbar => ScrollbarTarget

```

TouchSlider => SliderTarget
TouchText => TextTarget
TextCanvasItem => TextCanvasTarget
AnimaTouchable => AnimatedTarget

initialize => do_initialize
find_font => do_find_font
use_font => do_use_font
get_font_metrics => do_get_metrics
get_window => do_get_rect
do_make_window => do_make_pile
use_window => do_use_pile
set_window_name => do_set_pile_name
def_file_pos => do_def_file_pos
def_target => do_def_target
put_target => do_put_target
def_image => do_def_picture
put_image => do_put_picture
measure_image => do_measure_stamp
put_string => do_put_string
get_page_size => do_get_page_size
end_page => do_end_page
start_line => do_start_line
end_line => do_end_line
do_free_images => do_free_stamps

.free-images => .free-stamps

```

Add:

```

do_use_pile(id)
    Sets the current Pile
do_use_linked_pile(name)
    Follow a named link to a child pile
    (Sets current Pile)
do_link_parent_pile(name)
    establish a link from parent pile to this pile, of a
    given name. (DefinitionPile, ControlsPile)
do_use_parent_pile()
    Follow the link back to this pile's parent
    (Sets current Pile)
do_def_local(name)
    Define the name in the pile's local dictionary to
    be the TargetID of the last created target.
do_def_global(name)
    Define the name in the global userdict to
    be the TargetID of the last created target.

```

Associate pile names with PileID's in the server. (Left Contents, Left Definition, Left Controls, Right Contents, Right Definition, Right Controls...) Name is an argument to .new-pile. Add another argument to new_pile: Class of the window to use. (ContentsWindow, DefinitionWindow, ControlsWindow) (borders, labels, etc)

Make a client side array of pile structs, indexed by PileID.
 Keep track of the current pile.
 Keep the following per-pile info in the client:
 PageWidth

PageHeight
Article return stack

Make functions to record all this new stuff.

Make formatting the contents not dependant on "found-contents?"

Only start-contents when you find the .contents

If we hit eof without finding .contents, then don't do anything.

So we don't have to remember found-contents? in compiled
definitions, etc.

Make it so that executing compiled null contents is a noop.

redefining the control panel seems to break things...

When a pile gets zapped, do a "PileDict PileID null put", and check
for null piles's, after looking them up by their pileid, before doing
stuff to 'em.