

# TealInfo User's Manual



Program Version 3.11  
Last Updated: June 16, 2000

## Table of Contents

<b>Description .....</b>	<b>1</b>
<b>Contents .....</b>	<b>1</b>
<b>Installing .....</b>	<b>1</b>
<b>Overview .....</b>	<b>2</b>
<b>Using the Program.....</b>	<b>2</b>
<b>Folio Selection Screen.....</b>	<b>2</b>
Backing up files .....	2
Setting Preferences.....	2
<b>Folio Browsing Screen .....</b>	<b>3</b>
Details .....	3
Checkmark All/None .....	3
<b>Using Folios.....</b>	<b>3</b>
<b>Included Folio Files .....</b>	<b>5</b>
<b>Creating Folio Files.....</b>	<b>6</b>
<b>MkTIInfo Source File Format.....</b>	<b>6</b>
<b>Relative Object Placement .....</b>	<b>7</b>
<b>Special Tags .....</b>	<b>8</b>
<b>Inline Font/Underline Selection .....</b>	<b>8</b>
<b>Inline Hyperlinks.....</b>	<b>8</b>
<b>Scrolling objects .....</b>	<b>9</b>
<b>Using Object Links.....</b>	<b>9</b>
<b>Using Text Resources.....</b>	<b>11</b>
<b>Creating Commercial Folios .....</b>	<b>11</b>
<b>Appendix A - MkTIInfo Object Reference.....</b>	<b>12</b>
List .....	12
Poplist .....	12
Window.....	13
Pop Window.....	13
Outline .....	14
Checkmark .....	15
Goto .....	15
Label .....	16
Image .....	16
Rect.....	17
Tres.....	17
Password .....	18
Popedit .....	18
Pop Image .....	19
Checklist .....	19
Edit Window.....	20
Graffiti.....	20
<b>Appendix B - MkTIInfo Field Reference .....</b>	<b>21</b>
X (value) .....	21
Y (value) .....	21

W (value) .....	21
H (value) .....	21
BX (value).....	21
BY (value).....	21
BW (value).....	21
BH (value).....	21
SX (value) .....	21
SY (value) .....	21
FONT (value) .....	21
BFONT (value) .....	21
CYCLE (value).....	22
DELAY (value).....	22
TABS (value) .....	22
BTEXT (string).....	22
RECORD (value) .....	22
DATABASE (string).....	22
KEY (keyword).....	22
TARGET (string) .....	22
DEFAULT (value).....	22
<b>Appendix C - MkTInfo Style Reference .....</b>	<b>23</b>
<b>Appendix D - Revision History.....</b>	<b>27</b>
<b>Appendix E - Credits.....</b>	<b>28</b>
<b>Appendix F - Contact Info .....</b>	<b>29</b>
<b>Appendix G - Registering.....</b>	<b>29</b>
<b>Appendix H - Disclaimer.....</b>	<b>29</b>

## Description

Thank you for trying TealInfo, the powerful information display and presentation system, perfect for product catalogs, manuals, customer databases, technical data, movie guides, or any other lists, documents, or tabular information. Use it to intuitively display and organize information previously only available as freeform TealDoc files, flatform databases, data acquisition forms, or custom applications.

TealInfo browses flexible TealInfo data *folios*, which are interactive documents of organized tables, lists, and windows, complete with advanced user interface elements such as outlines, images, hyperlinks and popup lists.

*Folios* can be downloaded from the ever-growing library of available titles (most are free), or created from scratch without programming using a text editor and the command-line utility MkTlInfo.

## Contents

This archive contains the following files:

Program files:	TEALINFO.PRC MKTINFO.EXE	The TealInfo program file The TealInfo folio creation utility
Document files:	INFODOC.PDF INFODOC.HTM INFODOC.PRC REGISTER.HTM REGISTER.TXT	This document in Adobe Acrobat (PDF) format This document in HTML format (sans images) This document in TealDoc format TealPoint Registration form in HTML format TealPoint Registration form in text format
Sample folios:	SALESTAX.PDB TIPS.PDB METRICON.PDB TODOLIST.PDB PHONEBK.PDB INFOREF.PDB INFOREF.TXT	Sales Tax Calculator folio Tip Calculator and Guide folio Metric converter folio Handy, configurable to-do list folio Handy, configurable phone book folio Reference folio for folio development Source text file for reference folio

## Installing



Use the PalmPilot installer to install the program file **TEALINFO.PRC**. It will appear on the Palm Computing device after the next HotSync.

This manual is also provided in this archive as the PalmPilot document **INFODOC.PRC**, which can be viewed with our application TealDoc.

Instructions on how to use the Palm installer are in the Palm Handbook that came with your Pilot, PalmPilot, Visor or WorkPad.

To use the program, you'll also need to install some folio files, available on the TealPoint website at [www.tealpoint.com](http://www.tealpoint.com) or software archive sites like PalmGear HQ ([www.palmgear.com](http://www.palmgear.com)).

Included in this archive is INFOREF.PDB, a reference folio for *TealInfo* folio creation. SALESTAX.PDB is a calculator for sales taxes from 3-11% and TIPS.PDB is a tip calculator with a handy guide to tipping in different situations. METRICON.PDB is a metric conversion folio, containing conversions for temperature, distance, area, volume, and weight. Also included are TODOLIST.PDB and PHONEBK.PDB, good examples of useful, configurable folios that you can edit when on your Palm device to store useful information and tasks. See the section 'Included folios' below for more information.

## Overview

TeallInfo is a powerful new medium for presenting and displaying information. TeallInfo data *folios* are interactive documents of organized tables, lists, and windows, complete with advanced user interface elements such as outlines, images, hyperlinks and popup lists. Folios can be easily created with the included tools and a text editor, but look and feel like custom applications.

After installing the program file, you can run it by selecting its icon from the Applications menu.

## Using the Program

TeallInfo is easy to use, and consists of two basic interface screens, the **folio selection screen** and the **folio browsing screen**.

### Folio Selection Screen



Folio Name	Size
Computer Pinout Reference	11k
Guitar Tunings	3k
Metric Converter	76k
Nascar 2000	5k
NFL 2000-2001 Schedule	20k
Resistor Calculator	5k
The Chronicles of Narnia	33k
US Holidays & Events	8k
Wash. State Ferry Schedul	11k
World Airports	45k
Worldnet Local Numbers	21k

The first screen, the **folio selection screen**, appears when you first run TeallInfo. On it is a list of all the installed folio files in the current category. You can select or edit categories on your by clicking on the category selection button in the upper right hand corner of the screen. When you click on a listed folio, you can either open that folio for browsing, rename it, change it's category, or delete it permanently from the PalmPilot memory. To change the action performed, click on the action-selection buttons at the bottom of the screen before clicking on an entry in the list.

You can also perform similar operations with image files by changing the "Info" selector at the bottom right to "Pic". These TealPaint picture databases may be used as source imagery for folios. If you choose to Open a picture database, TealPaint will automatically be started if it is installed on your organizer. You'll need to manually load up the correct image database, however.

The following functions are accessible from this screen using the menu button silk-screened to the left of the Graffiti writing area:

#### Backing up files

The *Backup All in List* menu sets all the backup bits for folios listed in the current category. This forces the folios to be backed up into your backup directory each time you hotsync. Choosing *Backup None in List*, on the other hand, resets all the bits to zero. After hotsyncing to back your files, you'll probably want to reset the bits, or the Palm Hotsync system will scan the file looking for changes every time you hotsync, slowing down your hotsync sessions.

#### Setting Preferences

The following options can be set under the "Preferences" menu item.

##### Alphabetic Sorting

By default, the folio list is sorted alphabetically. You can turn off this feature to increase response time if you have many folios installed.

##### Check Version Number

By default, TeallInfo will automatically check the version numbers of any folios you open, warning you if any were made with a newer version of TeallInfo. You can turn off this feature, but note that you may encounter unpredictable results running newer folios with an older copy of TeallInfo should the new folios require features not supported in your version.

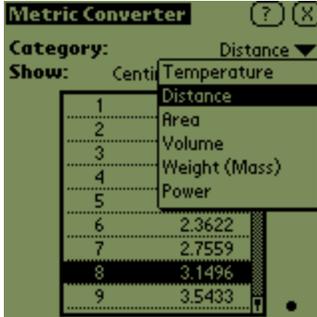
##### Hide Folios Starting with ~ (tilde)

In certain applications it may be desirable to create a series of interlinked folios where some of the sub folios are hidden from view and only accessible from other folios. To do this, name the folios starting with a tilde character (~) and set this option in preferences.

### Run in greyscale

When set, this options runs TealInfo in greyscale mode on a monochrome PalmPilot, automatically remapping TealPaint color images to shades of grey. TealInfo support four-shade greyscale on basic PalmPilot models and 16-shade greyscale on Palm IIIx, and Palm V models running OS 3.3 or higher.

## **Folio Browsing Screen**



When you open a folio for browsing, you are brought to the **folio browsing screen**, whose interface depends on the folio. The one common element is the close button in the upper right corner of most folios, but some may override this button as well. In any case, you can always close a folio and return to the Selection Screen by selecting the *Close* menu item.

### **Details**

The *Details* dialogue allows you to set two parameters for the current folio. The *Backup* bit determines whether the file is backed up during each hotsync, while the *Private* item determines whether the file is listed in the folio Section screen when private records are globally hidden in the Palm security settings.

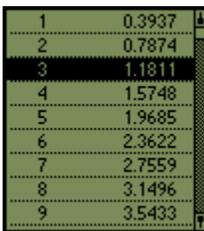
### **Checkmark All/None**

Sets or clears, respectively, all entries in presently shown CHECKMARK and CHECKLIST items. Use this to quickly reset or set a commonly-used task list, for instance.

## **Using Folios**

TealInfo folios can have many interface elements already familiar to PalmPilot owners. These include:

### Lists



Vertical lists of selectable elements, scrollable with a scroll bar if there are too many selections to fit on one view.

### Windows



Freely-wrapping scrolling text windows

### Checkboxes

Fancy Lists with selectable boxes for each item

### Popup Lists



Fields which show the current selection but pop up a selection list when you click on them.

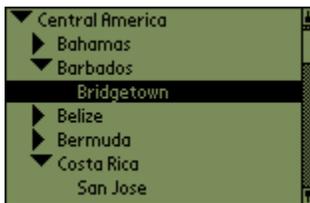
### Edit fields



Regions where you can click inside and enter or edit text. Unlike the system equivalent, these may take on more advanced formatting when not being edited.

### **In addition, TealInfo offers these newly-expanded features**

#### Outlines



Like lists, but featuring an outline-like tree hierarchy, subsections of which can be collapsed (hidden) or expanded (shown) under user control. To expand or contract an outline item, tap on the control triangle next to the item name.

#### Tables

12	0C	014	FF
13	0D	015	CR
14	0E	016	S0
15	0F	017	S1
16	10	020	DLE
17	11	021	DC1
18	12	022	DC2
19	13	023	DC3
20	14	024	DC4
21	15	025	NRK
22	16	026	SYN
23	17	027	ETB

A variant of windows supporting multiple columns with or without vertical and horizontal grid lines.

#### Popup Windows

Pushbuttons that bring up a window of text when tapped.

#### Goto Buttons

Buttons that open up another folio, if present, or a Doc file in TealDoc

#### Images

Picture imbedded in a folio or stored from TealPaint image databases

#### Popup Images

Pictures which display themselves when a button is tapped

#### Animations

Page-flipped images, providing simple cycling animation.

## Cycling Text

Windows with changing text messages for warnings or advertising messages.

## Passwords



Ability to ask for passwords when entering a folio for security or registration purposes.

## Hyperlinks

Specially-marked text that you can tap on to jump to other folios or popup windows, pictures, and lists.

## Object Linking

Ability to link the contents or behavior of a window, image, or list to the current selection of one to eight other lists, outlines or checkboxes.

## **Included Folio Files**

### INFOREF.PDB

This folio provides a useful quick reference for building TeallInfo folios. It lists the FIELD and STYLE tags supported by each TeallInfo object. Chose an object type from the popup list at the top of the folio. Its corresponding FIELDS and TAGS will appear in the windows below.

### SALESTAX.PDB

A simple calculator for sales taxes from 3-11% in ¼ percent increments. Choose your tax rate from the popup and pre-tax subtotal in the list on the left.

### TIPS.PDB

A simple tip calculator, choose the pre-tax total from the left to calculate an appropriate tip. Also includes a helpful tip guide giving general guidelines for common situations.

### METRICON.PDB

A metric conversion folio. Choose conversion type and units from the popups at the top of the folio to display an appropriate conversion table.

### TODOLIST.PDB

A handy to-do list folio. Click on the 'Edit' button to insert your own entries. Unchecked items appear on lines preceeded by a minus (-), while checked-off items are preceeded by plus (+) signs. A line with neither symbol will simply appear as an un-checkable label. To load more than one copy of this folio, rename the original folio on the **Folio Selection Screen** and use the Palm Install tool to install the TODOLIST.PDB file again.

### PHONEBK.PDB

A handy phone book folio. Click on the 'Edit' button to edit the current page of phone number entries. Use a single tab stroke to separate data intended for each nine and the 'return' stroke to separate lines. Change the list of categories using the 'categories' button, entering each category on a separate line.

## Creating Folio Files

Like TealDoc files, TealInfo folios are created first as text files that then get converted to a pilot-recognizable format using a conversion program. For TealInfo folios, the MkTlInfo conversion tool is included for this purpose.

On a PC, run MkTlInfo from a DOS box with the following format:

```
MKTLINFO <input text file> <output .PDB file> <folio title> <image database>
```

If any of the file names or titles have spaces, they should be enclosed in quotes. The image database (optional) is the name of a TealPaint image database in the current folder to insert into the folio to be used as an embedded source of imagery.

You can also run MkTlInfo without arguments and you'll be prompted to input the necessary file names.

If you're planning to make your own folios, we recommend downloading the TealInfo Dev Kit, available on our web site. It contains sample source text files and documentation to help in making folios, as well as a folio decompiler for reverse-converting unprotected folios.

When making a folio, you'll no doubt need to quickly preview changes, particularly when positioning on-screen objects. We recommend using POSE, the Palm OS Emulator, available at Palm's web site <http://www.palm.com>. This program simulates a Palm organizer on your computer desktop, allowing you to nearly instantly load new version of a folio for quick previewing.

### ***MkTlInfo Source File Format***

MkTlInfo takes plain text files with the following basic format: Objects are defined by a line with the name of the OBJECT, followed by lines containing FIELDS describing the object. Each field is followed by one or more VALUES on the same line, though TEXT fields may have VALUES on subsequent lines as well. VALUES can be numbers, expressions, keywords, or quoted text, depending on the type of FIELD.

```
(OBJECT1) (ID)
(FIELD1) (VALUE1A) (VALUE1B...)
(FIELD2) (VALUE2A) (VALUE2B...)
...

(OBJECT2) (ID)
(FIELD1) (VALUE1A) (VALUE1B...)
(FIELD2) (VALUE2A) (VALUE2B...)
...
```

where (OBJECT) is a keyword defining an on-screen interface element such as a list, label, or button, and (FIELD) is the name of an adjustable parameter for that object. Each (FIELD) is followed by one or more values, depending on the particular (FIELD).

Comment lines are indicated with a pound (#) sign. An lines beginning with a '#' are ignored by the folio converter.

EXAMPLE: This example creates a simple form with one scrolling object

```
#
# This is a simple scrolling list
#

LIST
  X 10
  Y 20
  W 140
  H 130
  TEXT "The Man Trap"
      "Charlie X"
      "Where No Man Has Gone Before"
      "The Enemy Within"
      "Mudd's Women"
      "What are Little Girls Made of?"
      "The Naked Time"
  STYLE HRULE
  FONT 1
```

### **Relative Object Placement**

When placing objects in a folio, it's often convenient to define an object's coordinates relative to another object, making them easy to align or move as a group. To help do so, numerical values can be entered as simple math expressions, and a number of predefined symbols can be used in the expressions, substituting for numbers:

X Y W H

Coordinates of the current object (must be defined on a previous line to be used)

BX BY BW BH

Coordinates of the current object's activation button

PREVX PREVY PREVW PREVH

Coordinates of the previous object

PREVBX PREVBY PREVBW PREVBH

Coordinates of the previous object's activation button

For instance, to define a column of checkmark objects, one might write:

```
CHECKMARK
  X 10
  Y 15
  W 100
  H 12
  TEXT "One"
CHECKMARK
  X PREVX
  Y PREVY+12
  W PREVW
  H PREVH
  TEXT "Two"
CHECKMARK
  X PREVX
  Y PREVY+12
  W PREVW
  H PREVH
  TEXT "Three"
```

**SPECIAL NOTE:** mathematical expressions currently support five operators: addition (+), subtraction (-), multiplication (\*), division (/), and modulus (%). Expressions are evaluated strictly from left to right, currently ignoring any mathematical precedence rules. Thus, 2+3\*5 is evaluated as 25, not 17.

## **Special Tags**

### **PNTFILE**

While images for IMAGE and POPIMAGE objects can remain as separate TealPaint databases, it's often more convenient to ship a folio as a single file with the images imbedded inside. An entire TealPaint image database can be included inside a folio with PNTFILE tag. For example,

```
PNTFILE PICTURES.PDB
```

will imbed the image database "Pictures.pdb" (should be in the current folder) into a created folio. Any references to the picture database by IMAGE or POPIMAGE objects should then use the name of the folio as it appears on the PalmPilot instead of the image database name. Only one picture database can be imbedded in a folio.

### **OUTFILE and OUTNAME**

MkTIInfo supports the use of special tags in the source file to pre-specify output and database (as seen on PalmPilot) file names. This latter name is especially important if the database is required to have a specific name because it's referred to by links from other folios or from internal references to imbedded TealPaint images.

Examples:

```
OUTFILE MYFOLIO.PDB  
OUTNAME "My little folio"
```

### ***Inline Font/Underline Selection***

TeallInfo supports changing fonts and underlining inside a body of text for a WINDOW or POPWINDOW object. Other objects may be less predictable. Font changing is done via HTML-like tags imbedded with the text of the object. For example:

```
TEXT
```

```
"This is some <$FONT=1>bold</$FONT> text."  
"This is some <$UNDERLINED>underlined</$UNDERLINED> text."
```

The <\$FONT> tag selects a font, while the </\$FONT> tag reverts it back to the original font. Unlike HTML, tags of the same type cannot be nested, so a </\$FONT> tag will always revert back to the original font, even if preceded by two <\$FONT > tags.

Note that the spacing of text in a TeallInfo object is governed by the original font defined in the object definition. Thus, you usually don't want to change fonts to one taller than the original font, or that text will get clipped by subsequent text lines.

### ***Inline Hyperlinks***

You can also insert hyperlinks in text to open a named folio or activate a named button to pop up a window or picture. For example:

```
TEXT
```

```
"Just tap <$LINK="My Folio">here</$LINK> to go to the index folio."  
"Just tap <$LINK="Info">here</$LINK> to show more information."
```

The link text can be either be the name of a folio to open or the name of a button to activate. The actual text of the button is used, not the ID of the button's object. Note that you can place the button entirely off-screen by setting it's BX field value to a large value, say 162, leaving the hyperlink as the only way to activate it.

## Scrolling objects

Teallinfo supports scrollable, oversized, text and image objects. The oversized width of text objects are defined by creating columns wider than the objects, while their heights are defined by the number of lines of content. Image widths and heights are defined by their source imagery.

```
WINDOW
  X 10
  Y 10
  W 50
  H 50
  TABS 75
  TEXT
    "This is an example of a text window with" \
    "horizontal scrolling"
```

Text objects appear with on-screen scrolling bars. In addition, for both types of objects, scrolling can be done by mapping the hardware up/down scrolling buttons and/or combinations for the four application launch buttons to vertical and/or horizontal movement. This is done using one of the `BUTTON_SCROLL` style flags.

By mapping scrolling behavior to different buttons for different objects, some folios can be made which operate largely pen-free.

## Using Object Links

Links allow one to link the `TEXT` content of an object to the selections of one or more other objects. Basically, you need to create multiple alternate `TEXT` blocks for the object to be controlled, and then specify which other objects do the controlling.

To link an object, first identify the objects you're linking-to by giving each of them a unique ID name (with no spaces) following the object tag. For instance, the following identifies a list object as object 'choice\_list'.

```
LIST choice_list
  X 5
  Y 5
  ...
```

Then, you can refer to this object in another object using the `LINKS` tag

```
WINDOW info
  LINKS choice_list
  TEXT "text one"
  TEXT "text two"
  TEXT "text three"
```

An object can only be linked to objects ("Linkees") which appear before it (the "Linker") in a text file. The "linker" object must also have multiple `TEXT` fields defined for it. The one that is actually used at any given moment is determined by the current selections of all of its "Linkees", multiplied out using the formula (example for 3 objects, 1, 2, and 3):

$$\text{index} = ((\text{value1} * \text{max2}) + \text{value2}) * \text{max3} + \text{value3}$$

where `max1`, `max2`, and `max3` represent the respective number of selections objects 1, 2, and 3 can take on. In this example, you must have  $(\text{max1} * \text{max2} * \text{max3})$  number of `TEXT` fields to cover all the possible values of `value1`, 2, and 3.

EXAMPLE:

```
LIST id_difficulty
  X 90
  Y 15
  W 60
  H 100
  TEXT
    "Easy"
    "Hard"
LIST id_speed
  X 10
  Y 15
  W 60
  H 100
  TEXT
    "Off"
    "Slow"
    "Med"
    "Fast"
WINDOW
  X 10
  Y 120
  W 140
  H 20
LINKS id_difficulty id_speed
  TEXT "This is Easy and Off"
  TEXT "This is Easy and Slow"
  TEXT "This is Easy and Med"
  TEXT "This is Easy and Fast"
  TEXT "This is Hard and Off"
  TEXT "This is Hard and Slow"
  TEXT "This is Hard and Med"
  TEXT "This is Hard and Fast"
```

**ADVANCED TIP:** The maximum value for each Linkee is normally determined by that object's current contents. If the contents of a Linkee can change, the math will not come up predictably if it is not the first item linked, because the MAX value will be variable. This commonly occurs with two Linkees when the first Linkee is a category, and the second is a sub-category whose contents depend on the first Linkee's value.

```
LINKS palm_models options
```

If this is the case, use the MAXVAL style flag to fix a maximum value for that second LINKEE. Place the MAXVAL tag in the linkee's object definition, not in the object containing the LINK field referencing the linkee. Set MAXVAL equal to the real maximum number of selections the LINKEE need have, and place extra entries in the LINKER object to pad out the unused combinations, so that you have *max1* groups of MAXVAL2 TEXT fields.

```
LINKS palm_models options
  TEXT "1. Pilot 1000"
  TEXT "2. Pilot 5000"
  TEXT "3."
  TEXT "1. PalmPilot Personal"
  TEXT "2. PalmPilot Professional"
  TEXT "3."
  TEXT "1. PalmIII"
  TEXT "2. PalmIIIx"
  TEXT "3. PalmIIIe"
  TEXT "1. PalmV"
  TEXT "2. PalmVx"
  TEXT "3."
  TEXT "1. PalmIIIc"
  TEXT "2."
  TEXT "3."
```

For more info on how to do multiple dependencies, download our TEALINFO DEV KIT for examples.

## **Using Text Resources**

Text Resources allow a folio to minimize duplicated text. Use the TRES tag to define an object with a single TEXT field. Be sure to give the TRES object a unique ID. Then, other objects can use this resource's text as its own object by simply referring to this ID preceded by an '@' (at signed) instead of adding its own quoted text. Note that adding a TRES object uses about 50 bytes of overhead, so space savings will only occur with text of sufficient length, but text resource still allow easy editing of commonly-used text.

### **EXAMPLE:**

```
TRES id_notfound
  TEXT "Sorry, no information is available about this device."

WINDOW
  X 5
  Y 14
  W 80
  H 100
  LINKS id_device id_class
  TEXT "4-5 Miles"
  TEXT "3 Miles"
  TEXT @id_notfound
  TEXT @id_notfound
  TEXT "15 Miles (estimated)"
```

## **Creating Commercial Folios**

When creating a commercial folio, you may wish to use some of TeallInfo's security features to keep your folio from being illicitly copied, decompiled, or used. As a first step, you can use the PASSWORD tag to lock the folio to a keyword. You can set up the keyword to be required to open the folio, or only if someone tries to decompile it. With the former option, using a numerical keyword or one that is not easily remembered, can keep the folio from being casually copied by someone unless they keep the keyword with them.

As a second level of protection, you create a custom version of the folio for every customer, using a key they wouldn't want to give out, like their credit card number.

Finally, if you collect the customer's HotSync user name, you can create a folio keyed to only work with their PalmPilot using the PASSWORD tag with with the LOCKOUT style option but an empty ("" ) TEXT field. See the TeallInfo Object Reference in the Appendix for more information.

## **Using TeallInfo for Forms Applications**

With a little work, TeallInfo can be used for simple forms applications using POPEDIT, EDITWINDOW, POPLIST, CHECKLIST, and CHECKMARK objects. For the latter three objects, the REGISTER style flag is typically used to keep any choices permanent.

To load the data back to the desktop, the folio "backup" option should be set in the details dialog for the folio. The folio will then be copied back to the user's Palm backup folder on HotSync. At that point, the folio decompiler "UntlInfo" (in the TeallInfo Dev Kit) can be used to convert the folio back to text format suitable for parsing by a database script or converter program.

Future versions of TeallInfo may include tools to help make such forms functionality available to general users and other non-programmers.

## Appendix A - MkTInfo Object Reference

### List

Description:

A scrollable, selectable list of text items, supporting optional columns and vertical and horizontal ruled lines.

Required Fields:

X Y W H TEXT

Optional Fields:

STYLE FONT TABS LINKS DEFAULT MAXVAL

Supported Styles:

VERT\_RULE HORIZ\_RULE INVERTED NO\_BORDER BOLD\_BORDER NO\_SCROLL  
ALIGN\_RIGHT ALIGN\_CENTER ALIGN\_LEFT\_START ALIGN\_RIGHT\_START  
ALIGN\_CENTER\_START ALIGN\_LEFT\_END ALIGN\_RIGHT\_END ALIGN\_CENTER\_END  
BUTTON\_SCROLL BUTTON\_SELECT STROKE\_FIND REGISTER

Notes:

Lists are defined by multi-line text fields, where each line of the text field defines an item in the scrolling list. In addition, up to 8 columns can be defined with the TABS field. By default, entries in each column are left-justified. The ALIGN\_CENTER and ALIGN\_RIGHT styles can override this justification. In turn, the ALIGN\_XXX\_START and ALIGN\_XXX\_END styles can be used to set alternate justification for the first or last columns, respectively, overriding the overall settings for those columns only.

### Poplist

Description:

A scrollable selectable item list triggered by popup button, resembling the PalmPilot category selector

Required Fields:

X Y W H TEXT BX BY BW BH

Optional Fields:

STYLE FONT TABS LINKS DEFAULT BFONT MAXVAL

Supported Styles:

VERT\_RULE HORIZ\_RULE INVERTED NO\_SCROLL BOLD\_BORDER ALIGN\_RIGHT  
ALIGN\_CENTER ALIGN\_LEFT\_START ALIGN\_RIGHT\_START ALIGN\_CENTER\_START  
ALIGN\_LEFT\_END ALIGN\_RIGHT\_END ALIGN\_CENTER\_END BUTTON\_SCROLL  
BUTTON\_SELECT STROKE\_FIND FILLED REGISTER SQUARE\_BUTTON

Notes:

A popup list appears when the user taps on its corresponding Popup trigger. The trigger shows the name of the currently selected item, and its bounds are defined by BX BY BW and BH, while X, Y, W, and H define the bounds of the popup list of items. The button bounds must be large enough to contain the longest selectable value in order for it to draw correctly.

## **Window**

### Description:

A scrollable text area that auto-wraps lines of text if necessary. Columns and line rulings are supported. This object can also be used to represent a table of data by setting up columns using the TABS field, and inserting tab characters to separate columns of data on each line.

### Required Fields:

X Y W H TEXT

### Optional Fields:

STYLE FONT TABS LINKS CYCLE DELAY

### Supported Styles:

VERT\_RULE HORIZ\_RULE INVERTED NO\_BORDER BOLD\_BORDER NO\_SCROLL  
ALIGN\_RIGHT ALIGN\_CENTER ALIGN\_LEFT\_START ALIGN\_RIGHT\_START  
ALIGN\_CENTER\_START ALIGN\_LEFT\_END ALIGN\_RIGHT\_END ALIGN\_CENTER\_END  
BUTTON\_SCROLL

### Notes:

A window holds a block of text which automatically word-wraps if longer than one line. Normally, to take advantage of this feature, paragraphs should be entered as one long line. As MktIInfo only supports a 4000 character long line, this is typically done by using the continuation mark (\) with the TEXT fields. See the TealInfo field reference below for more information.

Since text in a window is not selectable, WINDOW objects have no value and thus cannot be linked-to.

## **Pop Window**

### Description :

A scrollable auto-wrapping text window which comes up only when a specified trigger button is pressed.

### Required Fields:

X Y W H BX BY BW BH BTEXT TEXT

### Optional Fields:

STYLE FONT TABS LINKS BFONT

### Supported Styles:

VERT\_RULE HORIZ\_RULE INVERTED BOLD\_BORDER NO\_SCROLL ALIGN\_RIGHT  
ALIGN\_CENTER ALIGN\_LEFT\_START ALIGN\_RIGHT\_START ALIGN\_CENTER\_START  
ALIGN\_LEFT\_END ALIGN\_RIGHT\_END ALIGN\_CENTER\_END BUTTON\_SCROLL FILLED  
SQUARE\_BUTTON

### Notes:

The trigger button is defined by BX, BY, BW, and BH. Its label is defined by the field BTEXT.

## Outline

### Description:

A scrollable, selectable, hierarchial item list, this object shows a list of data in outline tree form, where subsections of the tree may be collapsed and hidden from view when desired.

### Required Fields:

X Y W H TEXT

### Optional Fields:

STYLE FONT LINKS DEFAULT TABS MAXVAL

### Supported Styles:

HORIZ\_RULE INVERTED NO\_BORDER BOLD\_BORDER BUTTON\_SCROLL  
BUTTON\_SELECT STROKE\_FIND REGISTER

### Notes:

Used much like a normal scrolling list, OUTLINE objects are displayed like an outline tree with items, sub-items, and sub-sub-items. Subsections can be hidden or shown by tapping on control triangles on the list. To define an outline list's contents, create a multiline TEXT field, precede each line with one or more '>' or '<' characters to define it's level in the tree. Root items are bare. Subitems to root entries are preceded by one '>', sub-items to sub-items are preceded by '>>', etc. If the item in the list should appear initially when the folio is first opened, then use a '<' character instead of a '>'. A '<' character forces the control on the previous line to default to the open state.

To set up an OUTLINE object for horizontal scrolling, use a TABS field with a single value specifying the virtual width of the scrolling area in pixels.

The following example sets up an OUTLINE object with three levels of items:

```
OUTLINE
X 10
Y 20
W 100
H 120
TEXT "Restaurants"
    "Hotels"
    "Taxis"
    "Museums"
    "Art"
    "Nature"
    "Marine"
    "Land"
    "Nightclubs"
    "Nice"
    "Seedy"
```

## **Checkmark**

### Description:

A checkmark control like standard PalmPilot checkmarks.

### Required Fields:

X Y W H TEXT

### Optional Fields:

STYLE FONT LINKS DEFAULT MAXVAL

### Supported Styles:

INVERTED REGISTER

### Notes:

When used as a link for another object, checkmark objects assume a value of 0 when unchecked and 1 when checked.

## **Goto**

### Description:

A button that closes the current folio and opens another one specified by name. Can also open a specified Doc file in TealDoc.

### Required Fields:

BX BY BW BH BTEXT

### Optional Fields:

STYLE BFONT TARGET

### Supported Styles:

INVERTED

### Notes:

A GOTO object with no TARGET resource replaces the standard close button in the upper right. Otherwise, TARGET, followed by the name of a folio in quotes specifies the folio to open when this control is tapped. Note that the TARGET must exactly match the name of the target folio as *it appears in TealInfo*, including capitalization and spacing

## ***Label***

### Description:

A simple text label.

### Required Fields:

X Y TEXT

### Optional Fields:

STYLE FONT

### Supported Styles:

INVERTED ALIGN\_CENTER ALIGN\_RIGHT

### Notes:

The label is drawn so that either the Upper Left, Upper Right, or Upper Center of the text is aligned at the designated coordinate.

## ***Image***

### Description:

An embedded TealPaint image or subrectangle of an image.

### Required Fields:

X Y W H SX SY DATABASE RECORD

### Optional Fields:

STYLE LINKS CYCLE DELAY

### Supported Styles:

NO\_BORDER BOLD\_BORDER BUTTON\_SCROLL

### Notes:

The bounding box for the image must be a multiple of 8 pixels wide. SX and SY define the offsets into the source image from which to grab the subrectangle. SX must also be a multiple of 8 pixels wide, and the source image must not be wider than 160 pixels.

DATABASE defines the name of the image database. It is case sensitive and must match the name of the Image Database as it appears in TealInfo or TealPaint. If the image is imbedded in the folio, then DATABASE should hold the name of the folio instead. RECORD defines the image number of the database to use. The first image in a database is image 0.

To create simple page flipping animation, set the CYCLE parameter equal to the number of frames of animation. TealInfo will step forward through the image databases, looping from image RECORD to image RECORD+CYCLE-1, stopping between from frames by an amount specified by DELAY, in tenths of a second.

## **Rect**

### Description:

A bare graphic rectangle, either filled or not, used as a graphic embellishment

### Required Fields:

X Y W H

### Optional Fields:

STYLE

### Supported Styles:

FILLED ROUND\_BORDER BOLD\_BORDER

## **Tres**

### Description:

A text resource

### Required Fields:

TEXT

### Optional Fields:

(none)

### Supported Styles:

(none)

### Notes :

This object does not appear on screen, but serves as a central location where reused text may be placed and pointed-to by other TEXT fields. It is also used to break-up TealInfo OBJECTS from exceeding the Maximum Object Text Size (See TEXT field for more info). There is about a 50-byte overhead for using a TRES object, so you won't save much space on only a few short lines of text.

See the field descriptions in the next section for details on how to use text resources.

## ***Password***

### Description:

The presence of this object in an folio forces causes a passkey to be entered when the folio is opened. This can be used to generate password-protected folios for security or sales-demo purposes.

### Required Fields:

KEY TEXT

### Optional Fields:

STYLE

### Supported Styles:

LOCKOUT REGISTER

### Notes:

The password is defined by the KEY field. When the LOCKOUT style is defined, the folio cannot be entered until a correct key is entered. Otherwise, the password request can simply be dismissed, functioning only as a registration reminder.

When the REGISTER style is defined, the password is saved once it's entered. If not, the password is asked-for every time. You can have multiple PASSWORD objects in a folio, but this is not particularly useful.

If you provide empty TEXT ("") and do not set the LOCKOUT style keyword, then the passkey will not be required to read the folio, but one will need to enter it before the folio can be decompiled using the UnTIInfo program in the TealInfo Dev Kit. In this use, the PASSWORD tag should appear at the top of the list to prevent decompiling the tags above it.

If you provide empty TEXT ("") and **do** set the LOCKOUT style keyword, then the folio will only open on a Palm whose Hotsync user name matches the passkey. Use this to make secure folios that cannot be beamed to any other Palm and be used.

## ***Popedit***

### Description:

An editable text popup.

### Required Fields:

BX BY BW BH BTEXT TEXT

### Optional Fields:

X Y W H BFONT FONT STYLE LINKS

### Supported Styles:

FILLED SQUARE\_BUTTON

Notes:

An editable text area that can be used either to save notes on its own or to modify text resources shared with other objects. The text editing window is brought up with a tap on the trigger button specified by BX,BY,BW, and BH. The POPEDIT object can be used to modify any TRES resource used by any other object, but unpredictable results may result if used to modify text used by other than WINDOW or POPWINDOW objects.

SPECIAL NOTE: When the contents of a POPEDIT window can change because the POPEDIT object is linked to other objects, the contents of the POPEDIT window **must** reside in text resource objects, or the text pre-wrapping system will get confused and mis-wrap the text after you edit one of the TEXT blocks near the top of the list of possible entries.

### **Pop Image**

Description:

An embedded TealPaint image or subrectangle of an image.

Required Fields:

BX BY BW BY BTEXT X Y W H SX SY DATABASE RECORD

Optional Fields:

STYLE LINKS CYCLE DELAY BFONT

Supported Styles:

NO\_BORDER BOLD\_BORDER FILLED SQUARE\_BUTTON BUTTON\_SCROLL

Notes:

This object has the same image and usage restrictions as an IMAGE object. As a small bonus feature, however, color images shown in a full-screen POPIMAGE objects will appear in 16 grey levels on a Palm V-style display (Palm V, Visor), even under OS 3.1, which does not otherwise support the 16-shade mode.

### **Checklist**

Description:

A scrollable, selectable list of checkmarked text items.

Required Fields:

X Y W H TEXT

Optional Fields:

STYLE FONT TABS LINKS DEFAULT MAXVAL

Supported Styles:

VERT\_RULE HORIZ\_RULE INVERTED NO\_BORDER BOLD\_BORDER NO\_SCROLL  
ALIGN\_RIGHT ALIGN\_CENTER ALIGN\_LEFT\_START ALIGN\_RIGHT\_START  
ALIGN\_CENTER\_START ALIGN\_LEFT\_END ALIGN\_RIGHT\_END ALIGN\_CENTER\_END  
BUTTON\_SCROLL BUTTON\_SELECT STROKE\_FIND REGISTER

Notes:

Checklists are defined by multi-line text fields, where each line of the text field defines an item in the scrolling list. Each line should start with either a plus (+) character for items that start up prechecked or minus (-) character for those which start out unchecked. Normally, the checklist will revert back to its initial state when the object is changed or the folio is exited. If the REGISTER style flag is used, the folio is not in flash memory, \*and\* the text for the checklist is stored in a TRES text resource, checked entries are permanently saved even if you leave TealInfo and return.

### **Edit Window**

Description:

An editable text window. Often used to make folios to be used as forms for data acquisition.

Required Fields:

X Y W H TEXT

Optional Fields:

STYLE FONT TABS LINKS

Supported Styles:

VERT\_RULE HORIZ\_RULE INVERTED NO\_BORDER BOLD\_BORDER ALIGN\_RIGHT  
ALIGN\_CENTER ALIGN\_LEFT\_START ALIGN\_RIGHT\_START ALIGN\_CENTER\_START  
ALIGN\_LEFT\_END ALIGN\_RIGHT\_END ALIGN\_CENTER\_END NO\_SCROLL  
BUTTON\_SCROLL

Notes:

Similar to a combination POPEDIT/WINDOW object, this is an editable text area that can be used either to save notes on its own or to modify text resources shared with other objects. It normally displays as a WINDOW object, but brings up an editing area when the window itself is tapped on. The editing area's overlays the defined window area, giving a clean, seamless editing mode.

Commonly, EDITWINDOW objects are used as single-line text fields, but multiple-lines are also supported. When used in multi-line mode, the editing and display modes do not line up as well as in single-line mode.

Otherwise, EDITWINDOW objects follow the same limitations as POPEDIT objects, including the requirement that Edit Windows linked to other objects must place all of its text in Text Resources to maintain proper formatting after editing.

### **Graffiti**

Description:

Places the graffiti shift indicator on screen.

Required Fields:

X Y

## Appendix B - MkTlInfo Field Reference

The following list details the fields used to describe the objects in Appendix A. Fields requiring a numerical value or expression such as 20+5 are followed by "(value)". Fields requiring a text string in quotes are followed by "(string)". Lastly fields requiring specialized text keywords (not in quotes) are followed by "(Keyword)".

### ***X (value)***

Defines the object's pixel offset from the left edge (0) of the screen.

### ***Y (value)***

Defines the object's pixel offset from the top edge (0) of the screen.

### ***W (value)***

Defines the width of the object in pixels (screen is 160 pixels wide).

### ***H (value)***

Defines the height of the object in pixels (screen is 160 pixels high).

### ***BX (value)***

Defines the pixel offset of a trigger object or button from the left edge of the screen.

### ***BY (value)***

Defines the pixel offset of a trigger object or button from the top edge of the screen.

### ***BW (value)***

Defines the width of a trigger object or button.

### ***BH (value)***

Defines the height of a trigger object or button.

### ***SX (value)***

For image objects, defines the horizontal offset into the source image from which the displayed image is grabbed. Must be a multiple of 8

### ***SY (value)***

For image objects, defines the vertical offset into the source image from which the displayed image is grabbed.

### ***FONT (value)***

Defines the font to be used for an object's body text. 0=standard font, 1=bold font, 2=large font, 3-5=standard system symbol fonts, 6=large number font. Use a program like *FontApp* (available at [www.palmgear.com](http://www.palmgear.com)) to preview the correct fonts.

### ***BFONT (value)***

Defines the font to be used for an object's activation button

**CYCLE (value)**

This field causes WINDOW objects to cycle through their list of TEXT fields. CYCLE indicates the number of TEXT blocks to cycle through, starting with whatever field would normally be shown. There must be at least that number of TEXT fields present. For image objects, CYCLE indicates the number of images in the database to cycle through, starting with the imaged defined by RECORD.

**DELAY (value)**

Indicated the rate at which cycling (if turned on) occurs. This value is in tenths of seconds between frames. Thus, a value of 10 will cause cycling a one change per second.

**TABS (value)**

The TABS field sets tab stop positions within an object in pixels. Up to 63 tab positions may be specified, creating 64 columns. Each tab value represents the left edge of a column measured in pixels from the left edge of the screen.

If you create columns wider than the width of the object, the object will appear with a horizontal scroll bar, which can be moved to view the full width of the window, list, or outline object.

**STYLE (keyword) (keyword)...**

Followed by one or more style keywords like *ALIGN\_LEFT* and *NO\_BORDER*, the STYLE field determines alignment and other miscellaneous properties for screen objects.

**BTEXT (string)**

Defines text for a button or trigger.

**RECORD (value)**

Used for IMAGE objects, this defines the number of the image (starting at 0) to be shown.

**DATABASE (string)**

Defines the name of a TealPaint image database to serve as a source for imagery.

**KEY (keyword)**

Defines the password that must be entered to unlock a folio with a PASSWORD object.

**TARGET (string)**

Defines the folio to open when a GOTO button is pressed. Multiple alternate TARGET fields can be used if the GOTO object is linked to other objects. In this use, the folio opened depends on the selection of the object (typically a list) linked-to.

**DEFAULT (value)**

Defines the initial value for a list, outline, or checkmark object (starting at 0)

**TEXT (string) (string)...**  
(string) (string)...

Defines text for this object. Text items should be quoted, with text for separate columns appearing on the same line as separate quoted groups of words. To insert a quote in a string, use two quotes together ("" ) inside the string.

Text for multiline text objects (such as lists and outlines) should appear as more quoted text on subsequent lines without the TEXT field name. Only TEXT fields allow information for the field to follow on subsequent lines after the field name itself.

A backslash '\' character at the very end of a line indicates that the text on the next line is a continuation of the current one. It should be used to break up a line of text but have it treated as if it were entered as a very long line of text, such as for a paragraph to be auto-wrapped by a WINDOW object. MkTInfo has a maximum input line length of 4000 characters.

There is a maximum total text size of 32k for all TEXT blocks in a single screen OBJECT. To overcome this limit, text can be externally-referenced to TRES objects, effectively splitting up the text into one OBJECT per TEXT block.

#### EXAMPLE:

```
TEXT
  "This is a line of text."
  "This is another line of text, followed by a blank line."
  " "
  "This is a paragraph of text, using the backslash " \
  "character as a line-continuation marker so that " \
  "TrealInfo will know to re-wrap this block as if it " \
  "were entered as a single line. Note the extra " \
  "spaces at the end of each line."
```

#### LINKS (value) (value)...

The LINKS field can be used to link the contents of an object to current selection in one or more other LIST, POPLIST, OUTLINE, or CHECKMARK objects, each of which is identified by its ID value which follows the object TAG. More information appears in the section below.

#### MAXVAL (value) ...

The MAXVAL field specifies the largest number of items a variable list can contain. Use this field when linking to objects whose contents can themselves change.

## Appendix C - MkTInfo Style Reference

The following are keywords supported by the STYLE field, where applicable.

#### INVERTED

Draw the object with reversed colors.

#### HORIZ\_RULE

Draw horizontal ruling lines between rows.

#### VERT\_RULE

Draw vertical ruling lines between columns. Note that vertical rules also have a subtle effect on the behaviour of tab characters in text. Without vertical rules, tabs function as they do in a word processor, advancing the cursor to the next column to the right. If text extends past a tab stop, then next tab brings the cursor to the next tab stop to the right. When vertical rules are turned on, however, text between tabs is treated as data in a cell, truncated if necessary to fit in the available space before the next tab stop, so data never overflows into a neighboring column.

#### NO\_SCROLL

Don't allow a scroll bar to appear and don't reserve pixels for one when word wrapping.

#### NO\_BORDER

Don't draw a border around the object.

#### BOLD\_BORDER

Draw a thicker border around the object.

#### ROUND\_BORDER

Draw a rounded border around the object.

#### ALIGN\_RIGHT

Align all text to the right of its column.

#### ALIGN\_CENTER

Center all text in its column.

#### ALIGN\_LEFT\_START

Align the first column to the left, ignoring the global setting.

#### ALIGN\_RIGHT\_START

Align the first column to the right, ignoring the global setting.

#### ALIGN\_CENTER\_START

Align the first column in the center, ignoring the global setting.

#### ALIGN\_LEFT\_END

Align the first column to the left, ignoring the global setting.

#### ALIGN\_RIGHT\_END

Align the first column to the right, ignoring the global setting.

#### ALIGN\_CENTER\_END

Align the first column in the center, ignoring the global setting.

#### LOCKOUT

PASSWORD: Require the key to be correct to enter the folio.

#### BUTTON\_SCROLL

Do page-by-page scrolling of a text object when scroll buttons are pressed. Default settings scrolls vertically with the Page Up/Down scrolling buttons. Using variations on the BUTTON\_SCROLL keyword, it's possible to map combinations of the four applications buttons (Datebook, Address, To Do List, and Memopad) to vertical and/or horizontal scrolling for objects.

When mapped to a folio in this way, the default behavior of the application buttons used is temporarily overridden only while the folio is open, and for POPIIMAGE, POPTXT, and POPLIST objects, only when the popup is shown.

When overriding such buttons, it's important to indicate the new button mapping in the folio to avoid confusing the user.

**Other Supported Variations include:**

BUTTON\_SCROLL\_A

Vertically scroll with Page Up/Down buttons

BUTTON\_SCROLL\_B

Vertically scroll with Date/Addr buttons

BUTTON\_SCROLL\_C

Vertically scroll with Addr/ToDo buttons

BUTTON\_SCROLL\_D

Vertically scroll with ToDo/Memo buttons

BUTTON\_SCROLL\_E

Horizontally scroll with Page Up/Down buttons

BUTTON\_SCROLL\_F

Horizontally scroll with Date/Addr buttons

BUTTON\_SCROLL\_G

Horizontally scroll with Addr/ToDo buttons

BUTTON\_SCROLL\_H

Horizontally scroll with ToDo/Memo buttons

BUTTON\_SCROLL\_I

Horizontally scroll with Date/Memo buttons

BUTTON\_SCROLL\_J

Vertically Page Up/Down, Horizontally Date/Addr

BUTTON\_SCROLL\_K

Vertically Page Up/Down, Horizontally Addr/ToDo

BUTTON\_SCROLL\_L

Vertically Page Up/Down, Horizontally ToDo/Memo

BUTTON\_SCROLL\_M

Vertically Date/Addr, Horizontally ToDo/Memo

BUTTON\_SCROLL\_N

Vertically ToDo/Memo, Horizontally Date/Addr

BUTTON\_SCROLL\_O

Vertically Addr/ToDo, Horizontally Date/Memo

## BUTTON\_SELECT

Move the current selection up or down when scroll buttons are pressed. Default settings move the current select with the Page Up/Down scrolling buttons. Using variations on the BUTTON\_SELECT keyword, it's possible to map combinations of the four applications buttons (Date, Address, To Do, and Memo) to scrolling movement as well. When mapping a single button to a POPLIST object, the object acts like the category selector in the standard application, cycling through available choices. As with the BUTTON\_SCROLL variations, when overriding the application buttons, it's important to indicate the new button mapping in the folio to avoid confusing the user.

### **Other Supported Variations include:**

#### BUTTON\_SELECT\_A

Change selection up/down with the Page Up/Down buttons

#### BUTTON\_SELECT\_B

Change selection up/down with the Date/Addr buttons

#### BUTTON\_SELECT\_C

Change selection up/down with the ToDo/Memo buttons

#### BUTTON\_SELECT\_D

Change selection down (looping) with the Date button

#### BUTTON\_SELECT\_E

Change selection down (looping) with the Addr button

#### BUTTON\_SELECT\_F

Change selection down (looping) with the ToDo button

#### BUTTON\_SELECT\_G

Change selection down (looping) with the Memo button

## STROKE\_FIND

Move the current selection to the next line whose first char matches an entered stroke.

## FILLED

RECT: Draw it filled-in  
POPLIST: Draw its button filled-in  
POPEDIT: Draw its button filled-in  
POPIIMAGE: Draw its button filled-in  
GOTO: Draw its button filled-in

## REGISTER

PASSWORD: Save a registration key once entered.

## SQUARE\_BUTTON

Draws an activation button with square corners, not default rounded ones

## Appendix D - Revision History

### Release 3.11 6/16/00

- Added support in installing TealInfo in Flash RAM (broken in 3.06)
- Added support for displaying font-change tags in POPLIST labels
- Added cleaner display of tabbed items POPLIST labels
- Added support for linking from popup objects (they now close correctly)
- Added support for launching TealPaint from chooser screen with correct image database
- Fixed SQUARE\_BUTTON style flag rendering (did not work)
- Fixed display of underlines and hyperlinks in INVERTED objects

### Release 3.06 5/7/00

- Added imbedded images stored inside TealInfo folio database
- Added support for oversize scrolling images (using BUTTON\_SCROLL)
- Added imbedded hypertext links to activate buttons or go to folios
- Added HTML-like tags for underlining within Window text
- Added HTML-like tags for changing fonts within Window text
- Added support for linking to TealDoc with GOTO objects
- Added linking of GOTO object targets, making their targets selectable
- Added support for displaying color TealPaint images in greyscale on mono pilots
- Added BUTTON\_SCROLL support for horizontal scrolling
- Added BUTTON\_SCROLL support for mapping application buttons
- Added BUTTON\_SELECT support for mapping application buttons
- Added EDITWINDOW object for in-place text field editing
- Added GRAFFITI object to show graffiti state indicator
- Added SQUARE\_BUTTON style option for un-rounded activation buttons
- Added "Checkmark all"/"Checkmark none" Menu items
- Added support in MkTIInfo for relative placing of objects using symbols (PREVX..etc) for numerical values
- Added simple math support in MkTIInfo for calculating object placement
- Added support for imbedding output file names in source file
- Added PASSWORD option to create folios locked to a specified HotSync User Name
- Added support for CmdGoto command from external programs to launch folios
- Added color launch icon for Palm IIIc
- Changed BUTTON\_SELECT option to wrap over at ends
- Changed checklist "uncheckable" lines to align on left edge
- Changed STROKE\_FIND to scroll the selected item to the top of the page
- Changed default close button to more attractive look.
- Increased speed of response when selecting folios in folio list
- Fixed display of over-long POPLIST text (clipped with ellipsis)
- Fixed word wrap in tables with many columns

### Release 2.25 2/9/00

- Fixed error trying to change category of folio in flash
- Fixed detection of folios in Visor flash memory

### Release 2.22D 1/2/00

- Added compatibility with Palm OS 3.5 pre-release build 25 (for color devices)
- Updated conversion tools to support long file names.

### Release 2.22C 12/5/99

- Added image drawing compatibility with BackdropEZ hack

### Release 2.22B 11/6/99

- Fixed MkTIInfo bug with extended lines

### Release 2.22 10/27/99

- Added PASSWORD option to protect folios from decompiling only (use empty TEXT)
- Added memo-taking and text-editing capabilities with POPEDIT object.
- Added CHECKLIST screen object for scrollable task lists.
- Added POPIMAGE screen object for pop-up images.
- Added ability to selectively set default OUTLINE controls to expanded position with '<'
- Added preference option to hide specially-named folios from the chooselist.
- Added easy interdependent multi-link folios using MAXVAL field
- Added independent font control for buttons and popups using BFONT tag.
- Added tick sounds on object screen tap
- Added reverse-color buttons using FILLED tag
- Added horizontal-scrolling OUTLINE objects using TABS tag.
- Added support for "STICKY" LIST, CHECKMARK, and OUTLINE controls using REGISTER tag.
- Added icon for launcher list view
- Removed X placement limitation on IMAGE objects

- Fixed graphic display glitches with columnar tables and lists.
- Fixed initializing of DEFAULT object states for linked objects
- Fixed delay opening folios with RECT objects.
- Added warning for deleting or renaming folios in flash memory
- Added click sound to folio selection
- Increased MkTInfo maximum input line width from 1024 to 5000 characters.
- Included configurable to-do list folio
- Included configurable phonebook folio

Release 1.59 5/4/99

- Fixed updating of objects usings scroll buttons on older palmpilots.
- Fixed vertical drag bar movement when horizontal scroll bar present.
- Fixed vertical button scroll when horizontal scroll bar present.

Release 1.56 3/19/99

- Fixed crashing opening folios created with buggy linux MkTInfo

Release 1.55 2/19/99

- Added Horizontal scrolling objects
- Added optional scrolling of objects using page up/down buttons
- Added optional selection of list items using page up/down buttons
- Added optional selection of list items by first letter graffiti stroke
- Added support for up to 64 columns per object
- Added pre-scrolling to show default selections
- Changed dragbar scrolling to one-page increments when clicking in container area
- Made TRES structures much more space efficient
- Removed limitation on number of TRES objects
- Fixed checkmark objects to act correctly as link source
- Fixed scroll-bar activation with OUTLINE objects
- Fixed clickable area on right margin of non-scrolling objects
- Fixed popup window/list refresh after menu selection
- Fixed dragbar drawing glitch

Release 1.32 1/5/99

- Fixed category conflict with TealDoc
- Fixed scrolling limit overrun

Release 1.30 12/29/98

- Added arrow buttons to scroll bar
- Added support for embedded quotes in TEXT resources using ""
- Fixed hanging in MkTInfo
- Fixed unexpected page closing when using custom close buttons

Release 1.20 12/25/98

- Added NO\_SCROLL STYLE to WINDOWS and LISTS
- Fixed intermittent crash with popup windows
- Fixed crash with very small windows
- Fixed scroll selection bug

Full Release 1.11 12/22/98

- Fixed registration system
- Added DEFAULT field value support

Beta Release 1.10 12/21/98

- Fixed drag bar updating on unbordered objects
- Added support for text links and resource IDs
- Fixed various MkTInfo bugs

Initial Beta Release 1.00 12/15/98

## Appendix E - Credits

Manual by Vince Lee, Tex Tennison, Sara Houseman, and Sara Houseman

## Appendix F - Contact Info

TealInfo by TealPoint Software  
©1999-2000 All Rights Reserved.

TealPoint Software  
454 Las Gallinas Ave #318  
San Rafael, CA 94903-3618  
We look forward to hearing from you.

Please visit us at [www.tealpoint.com](http://www.tealpoint.com), or email us at [contact@tealpoint.com](mailto:contact@tealpoint.com).

## Appendix G - Registering

Currently, you may register by snail mail, electronically through our website at [www.tealpoint.com](http://www.tealpoint.com) with a credit card and a secured server, or through PalmGear HQ at 1-800-741-9070. For the first option, the enclosed registration form is provided for your convenience. You may use this form or simply send the following to the address above.

- Product Name
- E-Mail Address
- HotSync User ID (Pilot Name Required for Passkey generation. It can be found on the main screen of the HotSync application on the Pilot as "Welcome \_\_\_\_\_" or in the corner on a PalmIII or higher)
- Check (drawn off a US Bank) or Money Order for (\$16.95 US) to TealPoint Software

## Appendix H - Disclaimer

We at TealPoint Software are committed to providing quality, easy-to-use software. However, this product is provided without warranty and the user accepts full responsibility for any damages, consequential or otherwise, resulting from its use.

This archive is freely redistributable, provided it is made available only in its complete, unmodified form with no additional files and for noncommercial purposes only. Any other use must have prior written authorization from TealPoint Software.

Unauthorized commercial use includes, but is not limited to:

- A product for sale.
- Accompanying a product for sale.
- Accompanying a magazine, book or other publication for sale.
- Distribution with "Media", "Copying" or other incidental costs.
- Available for download with access or download fees.

This program may be used on a trial basis for 30 days. The program will continue to function afterwards. However, if after this time you wish to continue using it, please register with us for the nominal fee listed in the program.

Thank you.