Tinderbox v9.6.0 Manual - Table of Contents

Tinderbox v9.6.0 Manual - Table of Contents 1
Tinderbox v9.6.0 Manual - Table of Contents 15
Help 20
Introduction To Tinderbox 21
- The Tool For Notes 21
- Notes 21
- Anatomy Of A Note 21
- Inheritance 21
- Active Notes 22
- Discovering Emergent Structure 22
  - The Simplest Thing That Could Possibly Work 22
- Multiple Views, Multiple Perspectives 22
- A Changing Program 23
- The Main Window 23
- The Inspector 23
- Get Info 23

Learning More 25
What’s Still New 26
- Tabs 26
- Attribute Browser 26
- Inspector 26
- Get Info 26
- Badge Families and the Badge Picker 26
- Link Parking Space 26
- Sketchnote Font 26
- Color Schemes 27
- Settings and Preferences 27

Basic Concepts 28
- Notes and Links 28
  - Notes 28
- Hierarchy 28
- Links 28
- Attributes 28
  - To change an attribute’s value 29
  - To change an attribute's default value 29
  - To create a new attribute 29
  - Attribute types 29
    - String 29
    - color 29
    - File 29
    - Boolean 29
    - Date 30
    - Interval 30
    - Number 30
    - List 30
    - Set 30
    - URL 30
    - Action (System use only) 31
  - Attributes and Agents 31
  - Exporting Attributes 31
  - Suggested Values 31
- Displayed Attributes 31

How Displayed Attributes Are Shown 31
Displayed Attribute value pop-up lists 32
Displayed Attribute value Autocomplete 32
Customizing the Displayed Attributes Table 32
User Attributes 32
Aliases 32
Aliases and The Original Note 33
Intrinsic Attributes 33
Stamps 33
To create a stamp: 33
To use a stamp: 33
Organizing Your Stamp Menu 33
The Stamps Container 33
Making Notes 33
Creating A Note 33
Note Placement Shortcuts 34
Selecting A Note 34
Renaming A Note 34
Making Links 34
Text Links 35
Text Links To A Specific Passage 35
Quick Links 35
Deleting Links 35
Prototypes 35
Using Prototypes 36
Children of Prototypes 36
Inheritance 36
Built-In Prototypes 37
User Prototypes 37
Breadcrumbs 37
Using the Tab Bar 38
Using Export 38
HTML and Preview 38
More Commands 38
Footnote 38
Inspect Stamps 38
Using Maps 40
Moving Around In Maps 40
Subtitle, Caption, and Text 40
Changing fonts 41
Text In Maps 41
The Prototype Tab 41
Links 42
Dancing 42
Adornments 42
Image Adornments 42
Locking Adornments 42
To make an adornment "sticky" 43
Grids 43
Geographic Adornments 43
Image Adornments 44
Smart Adornments 44
Badges 44
User badges 44

Flags 45
Cleanup 45
Containers 45
Display Expression 46
Hover Expression 46
Summary Tables 46
Attributes and the Summary Table 47
Histograms 47
Progress Bars 48
Using Outlines 49
The disclosure triangle 49
Editing a note title 49
Separators 49
The Outline Icon 50
The Link Widget 50
Identifying prototypes 50
Focusing On Notes 50
Indenting notes in the outline 50
Organizing Notes 51
Columns 51
Filtered Outlines 51
Using Charts 53
The disclosure triangle 53
Focusing the View 53
Organizing Notes 53
Using Timelines 54
Using Attribute Browsers 55
Using Hyperbolic View 56
Using Treemaps 57
Focusing the View 57
Visualization and Treemaps 57
Using Crosstabs 59
Find 60
Agents 61
Ziplinks 62
Markdown 63
BIG NEWS! 64
Composites 65
Composite Actions 65
Built-in Composites 65
Composites and Actions 65
Command and Info 66
Taggers 67
Taggers 67
Sentiment Analysis 68
Sentiment Analysis 68
Highlighters 69
Highlighters 69
Tinderbox Fonts 70
Queries and Actions 71
Actions, Expressions, and Rules 71
Comments 71
Expressions 71
True and False 71

Actions 72
Assigning Values 72
+= and -= 72
Conditional Actions 72
Side Effects 72
Repeated Actions 72
Edicts 73
Attributes and Constants 73
Attribute References 73
Constants 73
Designators 73
next and nextItem 74
Group Designators 74
find() 75
Text Patterns 75
Regular Expressions 75
Wildcard Characters 75
Character Ranges 75
Anchors 76
The \ [backslash] Character 76
Parentheses 76
Back References 76
Local Variables 76
Functions 77
String Operators 77
+ (concatenation) 77
* (repetition) 77
.at("keyword") 77
.capitalize 78
.beginsWith 78
.bold 78
.contains 78
.containsAnyOf 78
.countOccurrencesOf 78
.deleteCharacters 78
.empty 78
.endsWith 78
.extract 79
.extractAll 79
.find 79
.textColor(color) 79
.following 79
.highlights 79
.icontains 79
.icontainsAnyOf 80
.italic 80
.jsonEncode 80
.lookup 80
.lowercase 80
.next 80
.nounList 80
.paragraph(N) 80
.paragraphCount 80
.paragraphs(N) 80

,paragraphList 80
.plain 80
.replace(pattern,replacement[,arguments]) 80
.reverse 81
.sentence 81
.show 81
.size 81
.split("pattern") 81
.strike 81
.substr(start,length) 81
.trim 81
.toNumber 81
.tr(characters,replacements) 82
.uppercase 82
.wordCount 82
.wordList 82
.words(N) 82
escapeHTML(string) 82
format(value, format-string) 82
idEncode(string) 83
urlEncode(string) 83
wordsRelatedTo() 83
String Processing 84
Failure 84
The Expect Operators 84
The Capture Operators 84
The Skip Operators 85
.extract 85
.extractAll 85
.eachLine 85
try{} 85
JSON 85
XML 86
Numeric Operators 86
Numeric Functions 86
Mathematical Operators 87
sum(), sum_if, avg and avg_if() 87
Converting Numbers to Strings 87
Color Operators 88
Color Components 88
Converting Colors to Strings 88
Converting Strings to Colors 88
Manipulating Colors 88
Date Operators 88
Date Properties 88
Date Operators 88
Date/time comparison operators 89
Locales 89
Interval Operators 89
Interval Operators 89
List and Set Operators 90
Sorting 92
List Operations 92
Building Lists 92

Logical Operators 93
Logical equality tests 93
Logical Group Operators 93
Dictionary Operators 93
Operators 94
Link Operators 94
Link Operators 94
Finding Notes With Links 94
Making Links 95
Graph operations 95
Attribute Operators 96
attribute() 96
Type 96
Note Operators 96
inside 96
descendedFrom 96
Eval(), Action(), and Update 96
create operators 97
distance() 97
distanceTo() 97
document 97
document 97
fetch 97
fetch(url,headers,attributeName,actions[,method]) 97
inheritsFrom 97
inheritsFrom() 97
hasLocalValue 97
hasLocalValue() 97
isDuplicateName 98
isDuplicateName() 98
notify, speak 98
Notify 98
Speak 98
locale() 98
locale() 98
require() 98
require("Hints") 98
runCommand() 98
runCommand 98
select() 99
stamp() 99
version() 99
version(["arg"]) 99
Fetching Information From The Web 100
Attributes for Working With the Web 100
URL 100
AutoFetch 100
AutoFetchCommand 100
ReadOnly 100
LastFetched 100
Import 101
Copy and Paste 101
Drag Data Into Tinderbox Views 101
Spreadsheets and Comma-Separated Value Files 101

Explode 101
Watching Shared Notes 103
The Notes Application 103
Evernote 103
Files and Folders 103
Tot 103
DEVONthink Pro 103
Export 104
Simple Export 104
Text Substitution 104
Exporting to HTML 104
How Export Works 104
Templates 104
Making A Template Note 104
Exporting Information 105
What will transfer to the export 105
What will transfer in part 105
Additional Export Elements 105
Export Templates 106
Paths 106
Designators 106
Placeholders 106
Placeholders In Text 107
Macros 107
Group Designators 108
Export Operators And Groups 108
Markdown 109
AppleScript 110
Feathering Your Nest 112
Favorites 112
User Badges 112
Character Badges 112
Badge Attributes 112
User Fills 112
A Distinct Look For Each Project 112
Color Schemes 112
Release Notes 113
9.6.0 113
HIGHLIGHTS 113
Hyperbolic View 113
Links Between Documents 113
Posters 113
NOTABLE 113
Actions 113
changed() 113
Fetch 113
jsonEach 113
Select 114
Attribute Browser 114
Built-In Prototypes and Hints 114
Displayed Attributes 114
Edicts 114
Email 114
Export 114

Get Info 114
Hyperbolic View 114
Import 115
Inspector 115
Links 115
Outline View 115
Map View 115
Sorting 115
Stamps 115
Tabs 115
Text 115
DETAILS 115
Actions 115
action() 116
attribute() 116
comments 116
eachLink 116
Assignment 116
Designators 116
Dictionaries and Lists 116
collect 116
date() 116
.format 116
inside() 116
Attributes 116
links 116
unlinkTo/unlinkFrom 117
AppleScript 117
Displayed Attributes 117
Export 117
Infrastructure 117
Inspector 117
Map View 117
Outline View 117
Paste 117
Print 118
Roadmap 118
Text 118
Treemap View 118
Ziplinks 118
9.5.2 118
NOTABLE 118
Actions 118
Colors 118
Displayed Attributes 118
Inspector 118
Flags 118
Import 118
Preview 119
Views 119
User Interface 119
Details 119
Actions 119
Attribute Browser 119

Brainstorming 119
Built-In Hints 119
Explode 120
Export 120
Inspector 120
Text 120
Map View 120
Prototypes 120
Stamps 120
Text 120
9.5.0 120
HIGHLIGHTS 120
Brainstorming 120
Export and Preview 120
Images 120
Map View 121
Text 121
NOTABLE 121
Actions 121
Browse Links 122
Document Settings 122
Export 122
Find 122
Gallery 122
Get Info 122
Images 122
Inspector 122
Map View 122
Outline View 122
Text 122
Watched Folders 122
DETAILS 123
Actions 123
Agents 123
Export and Preview 123
Get Info 123
Import 123
Infrastructure 123
Map View 124
News 124
Stamps 124
Tabs 124
Text 124
9.3.0 124
DETAILS 124
Actions and Attributes 124
Document Settings 125
Export: Preview Pane 125
Get Info 125
Hyperbolic View 125
Import 125
Infrastructure 125
Inspector 125
Map View 125

Outline View 125
Roadmap 125
Text 126
9.2.1 126
DETAILS 126
9.2.0 126
NOTABLE 126
DETAILS 126
Actions 126
Agents 126
Copy URL 126
Gallery 126
Import 126
Infrastructure 126
Inspector 127
Map 127
Outlines 127
Sorting 127
Startup 127
Text 127
/ Trying to make a zip link of a name enclosed in parentheses [[(name).. failed to name the new note appropriately, because the parentheses were interpreted as a designator for a note named “”. This now works as expected. 127
9.1.0 127
HIGHLIGHTS 127
String Processing 127
XML Processing 127
JSON Processing 127
Stamps 127
Functions 128
NOTABLE 128
Actions 128
Export 129
Get Info 129
Local Variables 129
Map View 129
Tabs 129
DETAILS 129
Action Notes 129
Actions 129
Autofetch 130
Browse Links 130
Displayed Attributes 130
Document Settings 130
Explode 130
Export 130
Fills 130
Hyperbolic View 130
Import 130
Infrastructure 130
Inspector 131
Links 131
Map View 131

Outline View 131
Recent Badges 131
Roadmap 131
Text 131
9.0.0 132
HIGHLIGHTS 132
AI 132
COMMAND BAR 132
ACTIONS 132
GALLERY 132
HYPERBOLIC VIEW 132
MARKDOWN 132
NOTABLE 132
Actions 133
Agents 133
Appearance 133
Badges 133
Export 133
Hyperbolic View 133
Inspector 133
Text 133
DETAILS 133
Actions 133
Agents 134
Attribute Browser 134
Composites 134
Crosstabs 134
Displayed Attributes 134
Explode 134
Export 134
Find Bar 135
Geographical Adornments 135
Get Info 135
Hyperbolic View 135
Import 135
Inspector 135
Link Actions 135
Links Pane 136
Map View 136
Markdown 136
Outline View 136
Performance 136
Roadmap 136
Stamps 136
Tagger 137
Text 137
Timeline 137
Toolbar 137
Views 137
Tinderbox 8 137
8.9.2 137
NOTABLE 137
DETAILS 138
8.9.0 138

HIGHLIGHTS 138
NOTABLE 138
DETAILS 139
8.8.0 139
HIGHLIGHTS 139
NOTABLE 139
Applescript 140
DETAILS 141
HTML Export 142
8.7.1 142
NOTABLE 142
DETAILS 142
8.7.0 143
HIGHLIGHTS 143
NOTABLE 143
DETAILS 144
8.6.1 145
NOTABLE 145
8.6.0 145
HIGHLIGHTS 145
NOTABLE 146
DETAILS 146
8.5.0 147
HIGHLIGHTS 147
NOTABLE 147
DETAILS 147
8.2.3 148
DETAILS 148
8.2.2 149
DETAILS 149
8.2.0 150
NOTABLE 150
8.1.1 151
NOTABLE 151
8.1.0 151
NOTABLE 152
DETAILS 153
8.0.4 154
NOTABLE 154
DETAILS 154
8.0.3 155
NOTABLE 155
DETAILS 155
8.0.1 155
NOTABLE 155
8.0.0 156
HIGHLIGHTS 156
NOTABLE 157
DETAILS 159
Tinderbox 7 162
7.5.5 162
MISCELLANEOUS 162
7.5.3 163
MISCELLANEOUS 163

7.5.0 164
HIGHLIGHTS 164
NOTABLE 166
MISCELLANEOUS 167
7.3.0 170
HIGHLIGHTS 170
NOTABLE 171
7.2.0 172
HIGHLIGHTS 172
NOTABLE 173
MISCELLANEOUS 174
7.1.0 175
NOTABLE 175
7.0.2 176
NOTABLE 176
7.0.0 177
HIGHLIGHTS 177
Tinderbox 6 182
6.6.5 182
DETAILS 182
6.6.2 182
HIGHLIGHTS 182
NOTABLE 183
6.6.0 184
HIGHLIGHTS 184
NOTABLE 184
6.5.0 186
HIGHLIGHTS 186
NOTABLE 187
6.4.1 190
HIGHLIGHTS 190
6.4.0 191
HIGHLIGHTS 191
6.3.2 192
HIGHLIGHTS 192
6.3.1 193
HIGHLIGHTS 193
6.3.0 195
HIGHLIGHTS 195
6.2.1 197
HIGHLIGHTS 197
6.2.0 198
HIGHLIGHTS 198
6.1.3 201
HIGHLIGHTS 201
6.1.1 202
HIGHLIGHTS 202
6.1.0 204
SUMMARY 204
6.0.4 205
6.0.3 205
SUMMARY 205
6.0.2 206
SUMMARY 206

6.0.1 207
SUMMARY 207
License, Copyright and Colophon 210
Tinderbox™ Software License Agreement 210
Limited Warranty and Disclaimer 210

Tinderbox v9.6.0 Manual - Table of Contents

- Introduction To Tinderbox
  - The Tool For Notes
  - Anatomy Of A Note
  - Inheritance
  - Active Notes
  - Discovering Emergent Structure
  - Multiple Views, Multiple Perspectives
  - A Changing Program
  - The Main Window
  - The Inspector
  - Get Info
- Learning More
- What's Still New
  - Tabs
  - Attribute Browser
  - Inspector
  - Get Info
  - Badge Families and the Badge Picker
  - Link Parking Space
  - Sketchnote Font
  - Color Schemes
  - Settings and Preferences
- Basic Concepts
  - Notes and Links
  - Attributes
  - Displayed Attributes
  - User Attributes
  - Aliases
  - Intrinsic Attributes
  - Stamps
  - Making Notes
  - Making Links
  - Prototypes
  - Inheritance
  - Built-In Prototypes
  - Breadcrumbs
  - Using the Tab Bar
  - Using Export
  - More Commands
- Using Maps
  - Subtitle, Caption, and Text
  - Text In Maps
  - The Prototype Tab
  - Links
  - Dancing
  - Adornments
    - Image Adornments
    - Smart Adornments
  - Badges
  - Flags
  - Cleanup
  - Containers
  - Display Expression
  - Hover Expression
  - Summary Tables
  - Histograms
  - Progress Bars

Using Outlines
- The Outline Icon
- The Link Widget
- Focusing On Notes
- Organizing Notes
- Columns
- Filtered Outlines

Using Charts
Using Timelines
Using Attribute Browsers
Using Hyperbolic View
Using Treemaps
Using Crosstabs
Find
Agents
Ziplinks
Markdown
Composites
Command and Info
Taggers
Sentiment Analysis
Highlighters
Tinderbox Fonts
Queries and Actions
- Actions, Expressions, and Rules
- Comments
- Expressions
- True and False
- Actions
- Edicts
- Attributes and Constants
- Designators
- Text Patterns
- Local Variables
- Functions
- String Operators

        - + (concatenation)
        - * (repetition)
        - .at("keyword")
        - .capitalize
        - .beginsWith
        - .bold
        - .contains
        - .containsAnyOf
        - .countOccurrencesOf
        - .deleteCharacters
        - .empty
        - .endsWith
        - .extract
        - .extractAll
        - .find
        - .textColor(color)
        - .following
        - .highlights
        - .icontains
        - .icontainsAnyOf
        - .italic
        - .jsonEncode
        - .lookup
        - .lowercase
        - .next
        - .nounList
        - .paragraph(N)
        - .paragraphCount
        - .paragraphs(N)
        - ,paragraphList
        - .plain
        - .replace(pattern,replacement[,arguments])
        - .reverse
        - .sentence
        - .show
        - .size
        - .split("pattern")
        - .strike
        - .substr(start,length)
        - .trim
        - .toNumber
        - .tr(characters,replacements)
        - .uppercase
        - .wordCount
        - .wordList
        - .words(N)
        - escapeHTML(string)
        - format(value, format-string)
        - idEncode(string)
        - urlEncode(string)
        - wordsRelatedTo()
    - String Processing

        - The Expect Operators
        - The Capture Operators
        - The Skip Operators
        - .extract
        - .extractAll
        - .eachLine
        - try{}
        - JSON
        - XML
    - Numeric Operators
    - Color Operators
    - Date Operators
    - Interval Operators
    - List and Set Operators
    - Logical Operators
    - Dictionary Operators
    - Link Operators
    - Attribute Operators
    - Note Operators
        - inside
        - descendedFrom
    - Eval(), Action(), and Update
    - create operators
    - distance()
    - distanceTo()
    - document
    - fetch
    - inheritsFrom
    - hasLocalValue
    - isDuplicateName
    - notify, speak
    - locale()
    - require()
    - runCommand()
    - select()
    - stamp()
    - version()

Fetching Information From The Web

Import

Watching Shared Notes

Export

- Simple Export
- Exporting to HTML
- Export Templates
- Markdown

AppleScript

Feathering Your Nest

- Favorites
- User Badges
- User Fills
- A Distinct Look For Each Project
- Color Schemes

Release Notes

- 9.6.0
- 9.5.2
- 9.5.0
- 9.3.0
- 9.2.1
- 9.2.0
- 9.1.0
- 9.0.0
- Tinderbox 8
- 8.9.2
- 8.9.0
- 8.8.0
- 8.7.1
- 8.7.0
- 8.6.1
- 8.6.0
- 8.5.0
- 8.2.3
- 8.2.2
- 8.2.0
- 8.1.1
- 8.1.0
- 8.0.4
- 8.0.3
- 8.0.1
- 8.0.0
- Tinderbox 7
- 7.5.5
- 7.5.3
- 7.5.0
- 7.3.0
- 7.2.0
- 7.1.0
- 7.0.2
- 7.0.0
- Tinderbox 6
- 6.6.5
- 6.6.2
- 6.6.0
- 6.5.0
- 6.4.1
- 6.4.0
- 6.3.2
- 6.3.1
- 6.3.0
- 6.2.1
- 6.2.0
- 6.1.3
- 6.1.1
- 6.1.0
- 6.0.4
- 6.0.3
- 6.0.2
- 6.0.1

License, Copyright and Colophon

Help

What's New In Tinderbox:

- Faster. Sleeker. Better.

See the Release Notes for all the details.

- Introduction To Tinderbox
- Learning More
- What's Still New
- Basic Concepts
- Using Maps
- Using Outlines
- Using Charts
- Using Timelines
- Using Attribute Browsers
- Using Hyperbolic View
- Using Treemaps
- Using Crosstabs
- Find
- Agents
- Ziplinks
- Markdown
- Composites
- Command and Info
- Taggers
- Sentiment Analysis
- Highlighters
- Tinderbox Fonts
- Queries and Actions
- Fetching Information From The Web
- Import
- Watching Shared Notes
- Export
- AppleScript
- Feathering Your Nest
- Release Notes
- License, Copyright and Colophon

Introduction To Tinderbox

The Tool For Notes

Tinderbox is your personal content assistant, a tool for making, analyzing, and sharing notes.

Tinderbox makes it easy to capture ideas quickly, to organize them effectively, and to reorganize them as your understanding changes. In addition, your Tinderbox notes can actively work on your behalf, helping to organize themselves and to highlight notes (and relationships between notes) that might need your attention.

Tinderbox is designed to let your notes evolve and improve as you use them. You don't need to plan everything in advance; you can add new power to your Tinderbox notes when you need it.

Notes

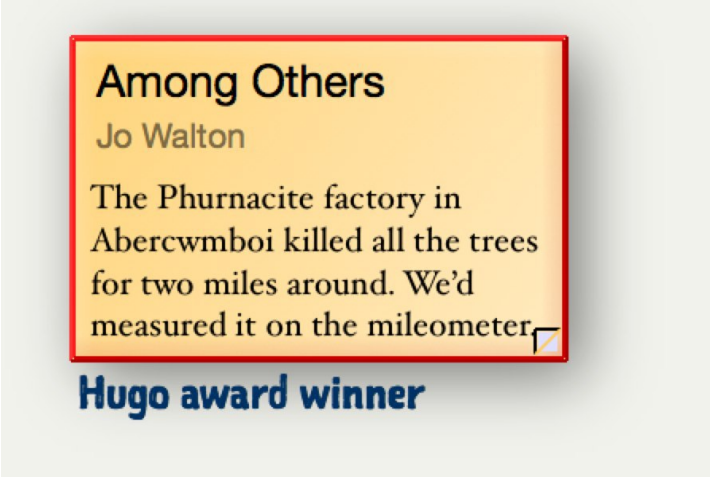
Notes are the basic elements of writing and information in Tinderbox. Each note has:

- title: the name of the note
- text
- attributes: other pieces of information about the note.

Notes live in Tinderbox documents. You must create a document before you create notes. You can create many Tinderbox documents, and individual Tinderbox documents can comfortably contain thousands of notes.

You can open several different Tinderbox documents at the same time, and you can open several different windows for the same Tinderbox document.

Anatomy Of A Note



A Tinderbox note collects a bundle of information — usually a focused bundle concerning one specific concept, task, or idea.

Internally, a note is a long list of *attributes* and *values*. For example, in the note above, we have:

attribute	value
Name	Among Others
Subtitle	Jo Walton
Color	poppy
Color2	light poppy
BorderColor	red
Caption	Hugo award winner

Tinderbox has several hundred built-in *system attributes* which have special meaning to the system. You can also add your own *user attributes*. If we need to distinguish between the name of an attribute and something else, we place a "$" before the attribute name: "Name" is a word, and **$Name** is an attribute.

Any note may also contain other notes. We call a note that contains other notes a *container;* any note may be a container.

Tinderbox also offers several special kinds of notes.

- **Adornments** decorate the background of maps to help make maps more clear. Adornments appear only in maps.
- **Separators** are simple lines that appear only in outlines and help to make long lists more clear.
- **Aliases** are notes that refer to a second note within this Tinderbox document. Aliases let the same note appear in several places in the document; most changes made to the alias are automatically applied to the original note as well.
- **Agents** are notes that search your Tinderbox document continuously for notes that meet the agent's criteria. If an agent finds a note that satisfies its criteria, it automatically places an alias of that note inside the agent. The agent can also apply an action to notes it finds. For example, one agent might find notes that represent overdue library books and automatically set their color to red.

Inheritance

Notes can *inherit* properties from another note, the note's *prototype*. Any note can serve as a prototype to another note.

If a note has a prototype, most of its attributes will be inherited from the prototype. If you do set an attribute's value for that note, the value you supply

overrides the inherited value. Inheritance lets you say, “Make this note just like that one, unless I tell you otherwise.”

Prototypes often establish the *type* or nature of a note. Notes about books might all share the prototype Book, and notes about people might all share the prototype Person. Prototypes make it easy to give related notes a special appearance, and also make it easier for Tinderbox to identify and reason about related notes. You can change a note’s prototype whenever you like.

Prototypes may themselves inherit from prototypes. Details of inheritance are discussed here.

A few *intrinsic* attributes are never inherited because inheritance would make no sense. Some attributes that aren’t inherited include Xpos, Ypos, Height, Width, and $Flags.

> To mark a note as a potential prototype, select the note and check Prototype in the Prototype inspector.
>
> To set a note’s prototype, select the note and choose the prototype from the popup menu of the Prototype inspector, OR right-click the note’s icon in outline view, OR right-click the note’s prototype tab in map view.

Tinderbox provides a variety of built-in prototypes. You can add built-in prototypes from the File menu. Built-in prototypes are created inside a top-level container named Prototypes; once created, you may change them to meet your specific needs.

Active Notes

Your Tinderbox notes can take an active role in helping to keep your work – and your attention – well organized.

Special notes called *agents* constantly scan your document, looking for other notes that match the agent’s query.

> For example: the agent Urgent Tasks might locate every task that is due within the next week.

Each agent collect *aliases* of every note that satisfies its query. Agents can sort the aliases as they prefer, and can also apply an *action* to the notes it finds.

> For example: the agent Urgent Tasks might sort tasks by due date, and its action might color overdue tasks red.

Any note can be a container that holds other notes. A container may also apply an action to notes when they are added to or removed from the container.

> For example: the container Library Books holds a note for each book you’ve borrowed from the library. When a note is created in this container, the container’s OnAdd action knows that it’s likely to be a book you have just borrowed and automatically sets its color, shape, and sets the due date two weeks from today. Of course, you can change any of these as necessary, but often Tinderbox can save your time and keep you from forgetting to note the due date entirely.

Each note can also have a list of Rules – actions that are automatically applied to the note at all times.

> For example: if a Task is marked as complete, it might automatically move itself into the container for Completed Tasks.

Discovering Emergent Structure

Many Tinderbox documents are meant for use over a span of months or years, changing frequently as you add and organize information. As your understanding grows, and as your needs change, you are bound to uncover new aspects of your work that you wish to record. Some facets that you originally captured only occasionally and informally will often take on greater importance. Some concepts you originally thought useful will turn out to be less relevant than you expected.

Over time, you may create new attributes to describe them and new agents and actions to take advantage of those attributes. This process is the discovery of *emergent structure*.

Like a conventional database, Tinderbox can handle lots of information, more information, indeed, than you might wish to memorize. Databases, however, are most effective when they are well structured from the beginning; they retrieve things quickly but make it hard to reorganize. Tinderbox is designed to facilitate reorganization. You do not need to get the structure right at first, and in fact you shouldn’t try.

The Simplest Thing That Could Possibly Work

Tinderbox beginners are sometimes tempted to invest lots of time at the start of a project, designing elaborate networks of agents and rules. Experience generally shows, however, that it’s best to keep things simple at the outset. In the early stages, you’ll have fewer notes, and you can experiment easily. Take advantage of that freedom as your work gathers way, adding and changing structure as you discover new relationships.

Of course, you may immediately recognize the usefulness of some kinds of information, and in that case you may want to create some relevant user attributes and prototypes. If you are researching the history a schoolteachers in frontier Wyoming, you’re bound to want prototypes for Teachers and for Schools, as well as prototypes for reference sources and for places. You might well want to create an attribute to hold the name of the $School at which a person taught.

Later, you might want to add more attributes; Tinderbox lets you add attributes whenever you like. Often, you’ll initially record information in the text of each note, and later you’ll move specific facts to appropriate attributes. This is neither difficult or onerous. Conversely, if you require yourself to fill in a long list of metadata whenever you add a new note, you may be tempted to delay making a note entirely until you’re confident the work is justified.

Experience suggests it’s best to record a simple note than to skip making the note entirely. Simple notes don’t take much time, and you can always delete them if they prove not to be relevant to your work.

Multiple Views, Multiple Perspectives

Tinderbox provides lots of different views of your notes. Examining your notes from new perspectives can give you fresh insights into their relationships and organization.

Maps lays all the notes in a container on a large plane, like an infinite whiteboard or like sticky notes on a big wall.

Outlines shows an outline of the entire collection of notes, or of a specific section.

Charts shows a hierarchical tree chart of the entire collection of notes, or of a specific section.

Timelines shows the timeline, indicating the $StartDate and $EndDate of each item in the container.

Treemaps show a diagram of your document as a set of nested boxes.

Attribute browser shows a list of each item in a container, arranged by the values of a chosen attribute.

> For example, if a container holds notes about business expenses, its attribute browser might group those expenses by client, and another browser might group the same expenses by city.

The View menu lets you change the type of the current view. Alternatively, right-click the current tab to change the view type.

You will frequently want to switch between different views of the same document. Tabs let you move back and forth among views; you can also open additional Tinderbox windows.

> For example: notes for a scholarly paper might use a map view for gathering facts and marshaling ideas, an outline for organizing your thoughts in a conventional form, and a second outline listing all your references an alphabetical order.

A Changing Program

Tinderbox is always changing and improving. Over ninety official Tinderbox releases have appeared to date.

Using the most current version of Tinderbox is a good idea for almost every user. Installing updates is straightforward: just download the new version and drag it into your Application folder or other convenient place on your hard disk.

Tinderbox currently comes with a year of free updates. After your year is up, updates will install as a demonstration of the new version. Of course, you may keep and continue to use the old version for as long as you like.

Because Tinderbox changes so often, some details of this document might be slightly out of date. If you notice discrepancies, let us know – there's a link at the bottom of each help page – and we'll do our best to correct them. But we always prefer to deliver improvements at the earliest possible date.

The Main Window

The main Tinderbox window contains:

1. The Tab bar, which holds tabs that bookmark different views on your document.
2. The Breadcrumb bar appears if beneath the tab bar if your current view focuses on only part of the document, and lists each ancestor of the current container.
3. The View pane contains an outline, map, chart, or attribute browser of your document, or of some part of your document.
4. The Title shows the title of the selected note.
5. The Displayed Attributes table shows some important attributes of the selected note. If the note has no displayed attributes, the table isn't shown. The **+** button lets you add new displayed attributes.
6. The Text area shows the text of the selected note. You can also use this area to see how the note would be exported.

Normally, resizing the window changes the width of the text area and the view pane proportionately. Hold down the command (⌘) key while resizing the window to hold the width of the text pane constant.

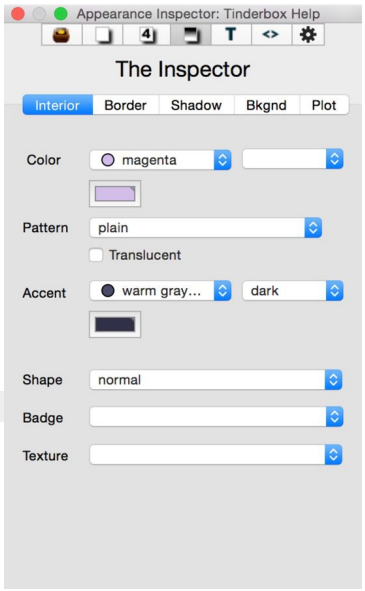
Previous versions of Tinderbox used lots of windows. You can open many windows in Tinderbox, but the tab bar usually makes this unnecessary.

The Inspector

The Inspector is a floating window that lets you examine and change properties of the selected note or notes.

To open the Inspector, select Inspector from the Window menu or press ⌘-1.

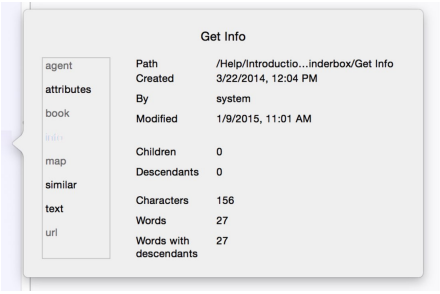
Icons at the top of the Inspector allow you to choose among six different inspectors:



- Tinderbox Inspector provides an overview of the entire document and of its agents and rules.
- Document Inspector lets you examine system attribute, create user attributes, and change colors, links, and stamps.
- Properties Inspector lets you adjust prototypes, templates, and other important properties of notes; its Quickstamp pane provides an easy way to inspect or change any attribute.
- Appearance Inspector lets you change the color, border, shadow, background, and other visual properties of a note.
- Text Inspector lets you change a note's font, size, and layout.
- Export Inspector provides access to the note's export template and markup properties.
- Action Inspector lets you edit actions, rules, queries, and sorting.

Get Info

The **Get Info** popover provides detailed information about any selected note. To open it, select a note and choose Get Info (⌘⌥-I) from the Note menu.



Drag the background of the Get Info window to tear off the popover, creating a window that will remain open until you close it.

Panes of the Get Info popover include:

- agent: displays an agent's query and actions
- attributes: lets you browse and edit all the attributes of the selected note
- book: explores attributes relevant to notes about books; for example, if $ISBN is set, this pane will try to locate an image of the cover, look up its author, and offer links to the volume in various catalogs and booksellers.
- info: displays often-used information about the node, such as its word count, child count, and number of descendants.
- map: if the selected note has an $Address or a $Latitude/$Longitude pair, this tab displays the location in a map.
- repetition: scans a note, section, or the entire document for words that are repeated more than once. Applied to a draft, this can be a valuable guide to overused or imprecise terms that might benefit from closer attention.
- similar: identifies notes with text similar to the selected note.
- text: displays the test of note note.
- url: if the note has a $URL, displays a preview of that URL.
- words: displays a word cloud for the selected note, section, or the entire document.

Learning More

Tinderbox is a deep and rich program, designed to assist you in complex projects that last months or years. Different people use Tinderbox in different ways. Nearly everyone believes that they only use a small fraction of Tinderbox's potential.

Some good ways to learn more about Tinderbox include:

- **Getting Started With Tinderbox**, in the Tinderbox Help menu, provides a walkthrough that describes some great ways to use Tinderbox.
- The Tinderbox Way, a book about the design and philosophy of Tinderbox written by Tinderbox designer Mark Bernstein.
- aTbRef (A Tinderbox Reference), a comprehensive reference to Tinderbox in the form of a Tinderbox document and Web site, by Mark Anderson.
- The Tinderbox Forum, an extraordinarily helpful and active online community of Tinderbox users
- The Tinderbox site offers a wealth of case studies, screencasts, and examples.
- The Tinderbox Cookbook provides many examples of Tinderbox actions and expressions.
- Periodically, Tinderbox Weekend hosts small conferences where Tinderbox enthusiasts can learn more about Tinderbox.

What's Still New

Tabs

Tabs at the top of each Tinderbox window let you switch instantly between different views of your notes.

The tab bar greatly reduces Tinderbox's appetite for creating vast numbers of windows. If you prefer, you may easily convert a tab into a separate window, but many users find tabs a faster and more convenient alternative.

Attribute Browser

The Attribute Browser is an entirely new major view which lets you explore your notes in new and powerful ways. In essence, the Attribute Browser shows you the contents of any container in your document, categorized by the values of a chosen attribute.

For example, suppose you have a container Expenses that contains notes about travel expenses. Important attributes for each expense record include the City where the expense was incurred, the Project to which this cost should be assigned, and the Amount spent. An Attribute Browser could let you see all the expenses for each city. Another attribute browser might show all the expenses that exceeded $100, organized by project.

Inspector

The Tinderbox Inspector lets you control dozens of attributes of the notes you have selected. To open the inspector, select Inspector from the Window menu or press ⌘-1.

The Inspector contains seven panes:

- **Tinderbox Inspector** provides an overview of the document and of its agents and rules.
- **Document Inspector** lets you inspect and modify the document's attributes, its symbolic colors, its link styles, and its stamps.
- **Properties Inspector** lets you change the selected notes prototype, declare it a prototype, template, or separator, and make it locked or sticky. The Quickstamp pane, formerly a separate palette, also appears here.
- **Appearance Inspector** lets you change the note's appearance, border, and background. For containers, the appearance inspector also lets you modify the container's background and its plot.
- **Text Inspector** lets you change colors and fonts used for the note's text, title, subtitle, caption, and hover expression.
- **Export Inspector** modifies the note's export name, template, and the note's markup.
- **Action Inspector** modifies the note's OnAdd action, as well as an agent or a smart adornment's query.

Get Info

The **Get Info** popover provides a wealth of information about the selected note. The extensible architecture of this window is designed to facilitate adding new information for new kinds of notes. Not all information is available for every note; Tinderbox highlights those panes that appear relevant.

You may tear off the Get Info popover to convert it into a window.

Currently-available panes include:

- **agent:** the query and action of the selected agent.
- **attributes**: explore and edit the entire range of attributes for the selected note.
- **book**: if the note has an $ISBN, provides access to its cover, ISBN-10 and ISBN-13 codes, and links to the book's bibliographic records and reviews.
- **info:** the note's modification date, word count, and other useful metadata.
- **map:** if the note has an $Address or if its $Latitude and $Longitude have been specified, provides a map of the note's location.
- **similar:** lists other notes with similar text.
- **text:** display's the note's text
- **URL:** if the note has a URL, previews the URL in a small browser window.

Badge Families and the Badge Picker

Tinderbox users may now add both individual custom badges and entire families of badges. A new badge picker lets you select from a much larger repertoire of symbols.

See: User Badges.

Link Parking Space

The link parking space is found in at the left edge of the tab bar, near the top of the main Tinderbox window.

When linking two notes in Tinderbox, select the source of the link and drag the link into the parking space. Then, locate the destination of the link — perhaps in a different tab. Finally, drag the link out of the parking space and into the destination.

When editing text, you may put a *text link* into the parking space. Simply select the link text and choose Note > Park Link (⇧⌘-L).

To make a Web link, select the link text and choose Note > Make Web Link (^⌥⌘-L).

Sketchnote Font

Tinderbox now includes the **Sketchnote** font, a superb font designed by Mike Rhode, the author of *The Sketchnote Handbook*.

Many handwriting fonts are idiosyncratic and hard to read, but Sketchnote is designed to be legible as well as informal. It's a very sophisticated font with thousands of adaptive letterforms.

Sketchnote is the default font for note captions, and is used in several Tinderbox Color Schemes.

Color Schemes

The Document Settings window now provides a selection of built-in color schemes. These schemes build on and extend an older Tinderbox feature and adjust many aspects of the Tinderbox document, not only color.

All Tinderbox color schemes include ten new colors, named simply "0" through "9", intended for use in indicators and dashboards. Often, these represent a visually uniform selection of grays that complement the overall color scheme, but in some cases they might be shades of the scheme's dominant color or hues ranging from red to green.

The primary purpose of color schemes is to define precisely what color corresponds to Tinderbox's symbolic color names. The Standard scheme, for example, defines "blue" as a fairly dark navy, but other schemes let "blue" denote sky blue, blue-gray, or other shades entirely.

Color schemes may include font settings and other attribute defaults as well as color definitions.

To save a color scheme as a file, open the Colors pane in the Document Inspector. Choose **Save Color Scheme…** from the action menu.

To use a color scheme file, simply drag that file into a document's view pane.

Settings and Preferences

The Document Settings window remains in its customary place in the Edit menu, and retains its shortcut ⌘-8.

Tinderbox Preferences (⌘-,) is now reserved for settings that affect all documents. At present, only the Registration pane appears in Tinderbox Preferences.

- If you're looking for Document Preferences, see **Document Settings** in the Edit menu.
- If you want to change the initial settings for new documents, consider making a "starter" document or stationery pad that contains the settings and infrastructure you prefer for the task at hand. For ready access to your starter documents, you can keep them in the Favorites folder inside the Tinderbox Support Folder, thus making them available in the File>Open Favorites menu.

Basic Concepts

- Notes and Links
- Attributes
- Displayed Attributes
- User Attributes
- Aliases
- Intrinsic Attributes
- Stamps
- Making Notes
- Making Links
- Prototypes
- Inheritance
- Built-In Prototypes
- Breadcrumbs
- Using the Tab Bar
- Using Export
- More Commands

Notes and Links

Notes

Tinderbox is a personal content assistant, a tool for making, visualizing, analyzing, and sharing notes.

Notes are the basic units of writing and information in Tinderbox. Each note has:

- title: the name of the note
- text
- attributes: other pieces of information about the note.

A Tinderbox document is a collection of notes. You must create a document before you create notes. You can create many Tinderbox documents, and individual Tinderbox documents can comfortably contain thousands of notes.

You can open several different Tinderbox documents at the same time.

Hierarchy

Notes can contain other notes. This allows you to arrange notes in a hierarchy.

A note that contains other notes is called a *container*. You could create chapters containing sections containing divisions; or topics containing subtopics; or any other hierarchy that fits your work.

You don't need to create a hierarchy in your Tinderbox document; you can leave all the notes at the same level.

In this manual, we use several terms to describe relationships in the hierarchy. A note's *container* is often called its *parent.* Other notes in the same container are *siblings;* the first note (in outline order) in a given container is the *eldest sibling,* and the last note is the *youngest sibling*. Notes immediately inside a container are called its *children;* a note's *descendants* include its children and also the descendants of the children. Notes that are not inside any container are *top-level notes.*

Links

A link is a connection between one note and another. Links allow you to make connections between ideas, and to quickly move between linked notes. A link can lead from a note as a whole, or from selected text within a note (like linked text on the Web).

You can also create a link to a URL—an external document on the Internet. Following the link will open that location in your Web browser.

Clicking on a text link will follow that link. To edit inside a text link, option-click inside the text link.

Attributes

In addition to its title and text, each Tinderbox note has a long list of attributes that describe aspects of the note: its name, color, position, shape, and size. You can define your own attributes as well. Attributes let you (and Tinderbox) put meaningful information in a known place.

For an informal note, you might simply write information in the note's text:

> Read Gibbon's *Decline and Fall of the Roman Empire*

In other contexts — research notes for a dissertation, say — you might prefer to be a bit more formal:

> Author: Edward Gibbon
>
> Title: The Decline and Fall of the Roman Empire

Author and Title are examples of attributes; by putting the name of the author in a specific place, it is easy for us (and for our Tinderbox agents) to distinguish a book by Gibbon from a book that is about baboons and gibbons.

You don't have to define Tinderbox attributes before you begin making and using notes. Tinderbox is designed to encourage experimentation and evolutionary change in your work. As your needs change and your understanding grows, you can add, change, and delete attributes whenever you like.

Each attribute has a value; values can be numbers, character strings, boolean (true/false) values, dates, colors, files, and so forth. Whenever you rename, edit, or change a note in Tinderbox, you are actually changing the value of one or more attributes of that note.

There are two types of attributes:

- System attributes: information built into all Tinderbox documents such as the color of the note, or its width and height. Tinderbox keeps this information about every note, and you may view, use, and change it.
- User attributes: you may add your own attributes that every note will have. For instance, you could add the attribute "Priority," and assign every note a priority level from one to five. Or in a bibliography, create the attribute "Type" and tag the note for each source as either "primary" or "secondary."

A few System attributes are

- Read-only attributes: information such as the date and time the note was created, or the date and time it was last modified. Other read-only attributes, such as TextLength and ChildCount, describe properties derived from the document's current state and are inherently read-only. Tinderbox keeps this information about every note, and you may view and use it, but you cannot change it.

Tip: attribute names are case sensitive—"Height" is not the same as "height."

You can inspect and change the value of any attribute in a note's Get Info... window (choose Get Info from the Note menu). Inherited values are in gray; values set specifically for this note are in black, and read-only values are italicized.

To change an attribute's value

There are several ways to change an attribute's value for a selected note or notes:

- The Inspector provides direct access to many of the most-commonly-used attributes. For example, the Appearance inspector lets you change $Color — the note's Color — `$BorderColor` — the color of the note's border — and many other attributes related to the note's appearance.
- The Quickstamp inspector allows you to change the value of that attribute for one selected note or for many selected notes.
- The Get Info window for a note shows all the note's attributes and allows you to change the value of any attribute.

To change an attribute's default value

The default value of an attribute is the value Tinderbox uses if no specific value has been set for that note.

The System Attributes panel of the Document Inspector lets you change the default value of attributes. If you set the default value of Color to red, then all newly-created notes will be red, and all notes for which no specific color was chosen will also become red.

To create a new attribute

- Open the User Attribute pane of the Document Inspector
- select the User tab
- click on the action button and select New User Attribute
- A new attribute will be created and given the temporary name "NewAttribute"
- Change the new attribute's name, type, and default as needed.

Attribute types

Attributes can be of various types, to contain various kinds of data:

String

Any sequence of text. "Yes," "Marie Hancock-Lowickie, Jr.," and the first paragraph of the Gettysburg Address would all be valid values for a String attribute. `$Name` is an example of a string attribute.

color

Color values are accepted in these formats:

- named colors. The name (string) of a color that either comes pre-defined in Tinderbox, or which you have defined for this Tinderbox document using the Colors pane of the attribute palette. Names set as text strings are stored as strings. Examples: "blue", "red", "minty-fresh-green", "subtle yellow".
- hexadecimal: six hexadecimal digits, the format used for describing colors in HTML. A hexadecimal value should be preceded by a number/hash sign sign. Example: `#A482BF`
- hue-saturation-value: hue—a value from 0 to 360 degrees; saturation and value—from 0 to 100 percent. Preceded by HSV, and enclosed in parentheses. Examples: HSV(0,100,50), HSV(240,80,80). HSV values are stored in the six-digit hexadecimal form (as above).
- RGB: Red, green, and blue, with intensity levels represented as integers from 0 to 250. Examples: rgb(0,0,0) and rgb(68,153,68). RGB values are stored in the six-digit hexadecimal form (as above).

`$BorderColor`, which controls the color of a note's main border, is an example of a Color attribute.

File

The pathname to a file. File-type attributes are used mainly or such System attributes as `$File`.

Boolean

A true/false value. `$HideDisplayedAttributes` is an example of a Boolean attribute.

If setting the value in an action, use the keywords `true` and `false`. Note that these keywords are case-sensitive and are never quoted. The number 0, the empty string, and the date "never" are considered to be false; other values are considered to be true.

Date

A date and time. When entering a value for a date, you may use your local Macintosh date formats and a variety of other standard forms. You can also use the terms yesterday, today, and tomorrow, and add or subtract units of time. Time (in hours/minutes only) is always part of Date data even if not displayed. For example, these are all acceptable values for a date attribute (on a month/day ordered Mac):

- 10/25/2002
- 10/25/02
- 10/25 (assumes the current year)
- November 15, 1955
- Nov 15 55
- today
- today - 7 days
- yesterday + 4 months (expressing date values relative to the current time is often convenient when defining agents and actions.)
- Wednesday (interpreted as next Wednesday)
- 11/25/2002 12:43

On non-US Macs, the day/month order might vary as might the allowed delimiters for day/month/year/time. Tinderbox defers to the current users "international" System Preferences for allowable formats. Internally, the data is stored in a locale-independent format so a date set on a US-locale Mac will still be correct is the document is subsequently opened on a French-locale Mac.

If sharing files cross-locale do be aware of where dates represented in code as string literals which thus fix a locale type. In such cases it is better to store that date in a Date type attribute in a note somewhere and refer to the attribute, ensuring the date is locale independent.

`$Created` is an example of a Date type attribute.

Interval

The length of time between two days. Intervals are typically entered as *hh:mm:ss* –

03:16

denotes an interval of three minutes and sixteen seconds. Intervals may also be entered as

3h30

which describes an interval of three and one half hours. Intervals may also include a number of days:

7 days 03:15:00

which describes an interval of seven days, three hours and fifteen minutes.

Number

Either an integer or a floating point (decimal) number, positive or negative. `$ChildCount` is an example of the Number type.

List

Lists are strings separated by semicolons, and are useful for lists of topics, categories, and tags.

- Status: urgent;delegatedToMartha
- Topics: programming;computer languages;courses;news
- Tags: tips and techniques; readme; TuesdayMeeting

Lists can be assigned with Stamps, Quickstamp, the Get Info Window, or the Displayed Attributes table. They may also be set by agents and other actions.

Lists can be sorted.

In many respects, List-type attributes can be considered a special form of String-type attribute (a string with semicolon delimited values), meaning that Lists can be coerced to/from Strings.

`$TimeLineBandLabels` is an example of a list attribute.

Set

Sets are best though of as a special form of List. Sets only hold one instance of any given value; passing a List with duplicate values to a Set will remove duplicates.

A Set, unlike a List, can't be sorted.

In many respects, Set-type attributes can be considered a special form of String-type attribute (a string with semicolon delimited values), meaning that Sets can be coerced to/from Strings.

`$KeyAttributes` is an example of a Set attribute.

URL

URL type attributes store URLs.

When displayed as a displayed attribute, URL type attributes show a globe icon to the left of the value edit box; if a URL value is set, clicking the icon

opens the URL in the users default web browser.

In many respects, URL-type attributes can be considered a special form of String-type attribute.

Action (System use only)

The Action data type is only used internally. `$AgentAction`, `$DisplayExpression`, `$OnAdd`, `$Rule` and `$TableExpression` are all Action type attributes. These attribute all hold strings of Tinderbox action code and are evaluated as such when used.

In many respects, Action-type attributes can be considered a special form of String-type attribute.

Attributes and Agents

Agents often use attributes to locate notes accurately. If you define a user attribute Author, agents will locate books in your notes written by Booker T. Washington and ignore notes on Seattle, Washington.

Agents can also set the value of attributes. An agent could look for items that are due within a week and automatically set the priority to "urgent."

Exporting Attributes

The value of any attribute can be exported to HTML using the expression ^value($ *attribute_name*)^. For example, if we have stored the location of an image in user attribute ImageName, we might export:

`<img src="">`

Suggested Values

When working with displayed attributes for string, set, and list attributes, Tinderbox provides a pull-down menu of all the values currently used in the document. The attribute inspectors also allow you to add **suggested values**; these values will be available in menus even if they are not currently used in the document. Suggested values can be useful when an attribute's vocabulary should be limited -- for example, the anticipated values of $Status might be

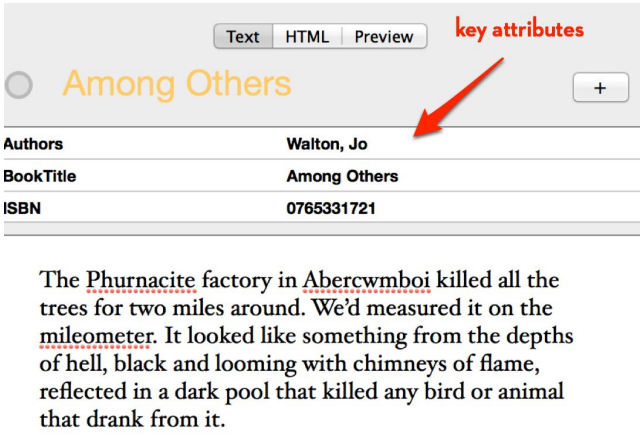
`"planned;in progress;overdue;completed"`

Suggested values are entered as semi-colon delimited lists.

Displayed Attributes

Tinderbox keeps track of hundreds of attributes for each note, but a few attributes of each note are likely to be of particular interest to you. If you are making notes about books, for example, displayed attributes might include author, title, and ISBN number. Displayed Attributes were formerly called "key attributes".

A note's displayed attributes appear at the top of its text pane, and at the top of the text window.



You choose the displayed attributes by setting the value of the attribute `$DisplayedAttributes` — just enter the names of the attributes you'd like to make displayed attributes, separating them with semi-colons.

To add an attribute to the note's displayed attributes,

- Press the Displayed Attributes Button (labelled "+") in the upper right-hand corner of the text pane
- Add the displayed attributes you prefer

Drag attributes to up or down to rearrange them. Select any attribute and press Delete to remove it.

Notes often inherit displayed attributes from their prototype.

How Displayed Attributes Are Shown

Attribute data types are discussed in more detail further below, but it is worth noting how attribute data types are displayed or edited slightly differently when a displayed attribute:

- String data types appear as a simple text field.
- Boolean data appear as a checkbox.
- Color attributes display the color name and a swatch.
- Date type attributes are shown as the day, month and year plus hours and minutes, formatted according to the user's System Preference using the 'short' data format. If a date is entered manually and the time is omitted, the current system time is appended. Seconds are not used. (To choose a different date format, use $DisplayedAttributesDateFormat.)
- Numeric attributes are displayed as numbers.
- List and Set attributes are shown as a single string with semicolons separating individual values.
- URL attributes show a globe icon to the left of the text field. Click the icon to open the URL in your web browser.
- Email attributes show an email icon to the left of the text field. Click the icon to compose an email in your default email application, addressed to the email recipients in the field. The body of the email is typically the Text of this note. However, if $EmailTemplate contains a path to a template note, that template is applied to this note and the results are used as the email body. This allows you to conditionally text from other notes, or to interpolate the of this note's attributes in the body of the email.
- File attributes show a folder to the left of the text field. Right-click the folder for a contextual menu of operations on that file; click the folder to open the file in the appropriate application.

Displayed Attribute value pop-up lists

For String, List and Set data types, a disclosure triangle at the right end of the attribute edit box opens a pop-up list which shows all discrete values for that attribute in the current document; for list and sets the listing is of all discrete list values across the document. As value lists may grow long, they are limited to 99 values. For such attributes no pop-up list control if shown - so the absence of a disclosure triangle indicates many values. Listings are case sensitive ('dog' and 'Dog' will be separate entries).

Clicking a list item sets a String attribute to that value or with Lists and Sets it is appended to existing values.

Displayed Attribute value Autocomplete

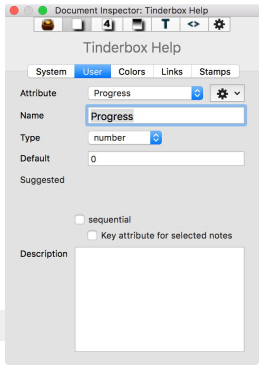
Where value lists are offered, autocompletion is available. As the user types a value it is completed with the nearest possible case sensitive match. Up and Down keys can be used whilst editing to scroll other possible list values for the stored value list.

Customizing the Displayed Attributes Table

To adjust the size of the text in the displayed attributes table, choose the size you prefer from the **Window ▸ Displayed Attributes** menu.

The $**DisplayedAttributesFont** and **$DisplayedAttributesFontSize** attribute determines the typography of the displayed attributes table for any given note. This overrides the document-wide setting, allowing you to have some notes with larger or smaller displayed attribute tables.

User Attributes



Tinderbox documents are created with several hundred attributes, most of which have a specific meaning to Tinderbox itself. For example, **Height** and **Width** describe the size of the note in the Tinderbox map view. These built-in attributes are called *System Attributes.*

You can add your own *user attributes* to any Tinderbox document. To define a new attribute, open the Document Inspector and switch to the User pane.

In the Inspector, you can use the action menu to create or delete user attributes, or select any use attributes you have already defined. The inspector lets you choose to type of attribute – some of the choice included number, string, URL, or list – and set the default value that the attribute should use. A text field labeled "Description" invites you to write a brief explanation of the way you intend to use the new attribute; this may save confusion amongst your colleagues and collaborators.

Aliases

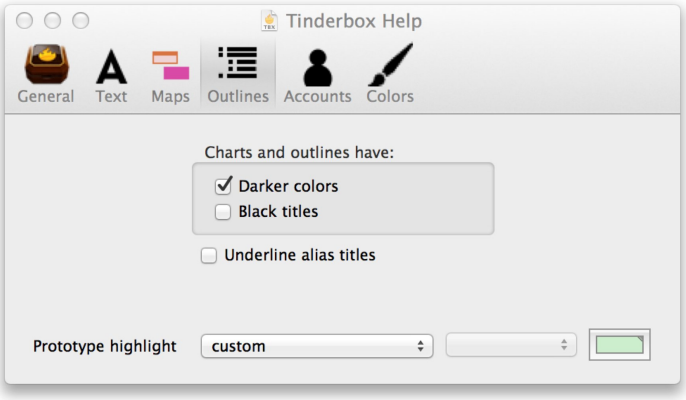
Aliases are notes that refer to a second note within this Tinderbox document. Aliases let the same note appear in several places in the document; most changes made to the alias are automatically applied to the original note as well.

Agents create and hold aliases of notes that match the agent's criteria. You can also create aliases manually.

To make an alias of a note:

- Select the note.
- Choose **Make Alias** from the Edit menu.

Aliases look like other notes, but the name of an alias is drawn in italics. Because some languages do not use italic typefaces, you can also request that the names of aliases be underlined using a checkbox in the General pane of Document Settings.



Aliases and The Original Note

The attribute values of an alias are almost always identical to the values of the original note. For example, the $Color of an alias is the color of the original. If you make the original note red, the alias will be red; if you make the alias green, the original note will also be green.

If the original note is deleted, all its aliases are also deleted. Deleting an alias, on the other hand, has no effect on the original note.

Intrinsic Attributes

A few attributes are *intrinsic.* Aliases have their own values of intrinsic attributes. Intrinsic attributes are never inherited from prototypes.

Intrinsic attributes concern properties of the alias that are necessarily distinct from the original. For example, moving an alias does not move the original note, so $Xpos and $Ypos are intrinsic. Other intrinsic attributes include $Height, Width, $ID, $Container, $Created, $Modified, $Creator, $RuleDisabled, $DisplayExpressionDisabled, $IsPrototype, $Flags, $MapScrollX, $MapScrollY, $TimelineBand,and $AgentPriority.

A number of read-only attributes are intrinsic because their value may differ from the value of the original. These include $OutlineOrder, $OutlineDepth, $SiblingOrder, $InboundLinkCount, $OutboundLinkCount, $WebLinkCount, $TextLinkCount, and $PlainLinkCount.

In expressions and actions, an alias refer to the intrinsic attributes of its original note using the designator `original`. For example, if `this` note is an alias, `$Height` refers to the position of the alias and `$Height(original)` refers to the height of the original.

Stamps

Stamps let you assign frequently-used values to your selected notes. For example, you could defind a stamp to turn notes red, or to set the value of $DueDate to a week from today.

To create a stamp:

- Choose **Inspect Stamps…** from the Stamps menu.
- Press the “+” button to create a new stamp.
- Name the stamp.
- Enter the action the stamp performs, and press Return

Most stamp actions are simple. For example:

```
$Color="red"
```

sets the color of the selected notes to red.

To use a stamp:

- Select the note(s) to which you want to apply the stamp
- Select the stamp you want from the **Stamps** menu.

Organizing Your Stamp Menu

Most Tinderbox documents only need a handful of stamps, but occasionally you might need more. For example, qualitative analysis of interview data often involves numerous coding choices, and stamps can speed this task considerably.

If your stamp name contains exactly one colon, it will be placed in a submenu of of the main Stamps menu. For example, if a stamp is named

```
Color:red
```

Tinderbox will add a stamp “red” to the submenu **Color** in the Stamp menu.

The Stamps Container

Stamps may now be stored in the Hints container, created automatically by selecting **Built-In Hints** from the File menu. Each stamp is stored as an individual note in /Hints/Stamps.

If you use File ▸ Built-In Hints to create or update the hints container, stamp notes are automatically created for each of your existing stamps.

Notes in the Stamps container with names in parentheses are treated as commentary rather than stamps. For example, the note (What’s This?) explains the container and contains no stamp.

Notes in the Stamps container with names beginning with a period are not listed in the Stamps menu, but may be used with the stamp(stampName) action.

To create stamps that appear as a submenu, continue to name them menuName:stampName.

Making Notes

Creating A Note

In any view,

- select an existing note (by clicking on it)
- press Return

  or

- right-click in the window, then choose Create Note from the contextual menu

or, in a map view,

- double-click the map background

Or, in any view,

- choose **Create Note** from the Note menu.

To finish creating the note,

- type a name for the new note

Note Placement Shortcuts

In map view, pressing [Return] places the new note to the right of the selected note. If there is not sufficient space, Tinderbox will find a suitable place for the new note.

Press ^-[Return] in map view to create a new note to the left of the selected note.

Press ^⌥-[Return] in map view to create a new note beneath the selected note.

Selecting A Note

Click on any note to select it. If a note is already selected, it will be deselected and the note you click will be selected.

In outline and chart view, shift-click a note to extend the current selection to include all the notes between the selected note and the note you shift-clicked. In map view, shift-click simple adds the chosen note to the current selection.

⌘-click a note to add that note to the current selection. ⌘-click a selected note to deselect it.

Click in the background of the view to deselect all notes, or press ^⌘⌥-A (Edit>Deselect All).

Renaming A Note

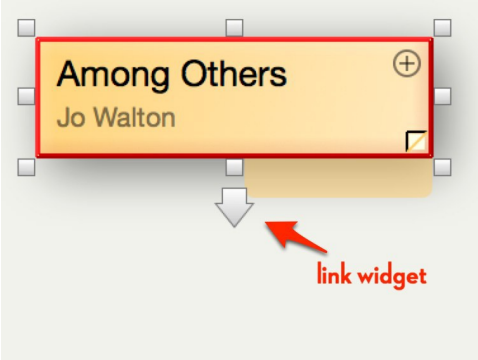
To change a note's name, select that note. Then

- Press ⇧⌘-Return and enter the new name, OR
- Click on the title and hold until the edit cursor appears, OR
- Edit the title in the right-hand pane

If you wish, Tinderbox may compute the *display expression* in place of displaying the same name. For example, a container of tasks might display the number of tasks remaining to be done. Display expressions are entered in the Title pane of the Inspector (⌘-1).

Making Links

In map view, a *link widget* appears beneath each selected note.



To make a link, click the link widget and drag the link to its destination.

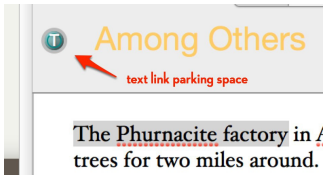
If the destination is not conveniently to hand, drag the link to the *parking space* in the upper right-hand corner of the window.

After the link is parked, locate the destination. Then, click the parking space and drag the link to its destination.

Shortcut: In map view, make a new note that is linked to an existing note by clicking the link widget and dragging to the place where you'd like to create the new note.

Text Links

Links may also be anchored to any span of text inside a note. To make a text link, select the text you wish to link. Then, click on the *text link parking space* above the text pane and drag the link to its destination — or to the main parking space.



Clicking on a text link follows the link in Tinderbox, or, if it's a Web link, opens the link in your browser. If you want to edit linked text, option-click in the link to set the insertion point without following the link.

Text Links To A Specific Passage

Text links may link a specific place in a note.

To specify a specific destination of the link:

1. Select the source of the link and drag the text link to the link parking space at the left side of the Tinderbox window.
2. Select the destination, and scroll the text pane as needed to make the destination text visible.
3. Drag the link out of the parking space and click in the text pane at the destination location.

When the text link link is followed, Tinderbox will scroll the text so the destination text is visible.

Quick Links

You can make a text link to a note by typing two left-brackets and the first letter of the note's name. For example, if you type [[A, Tinderbox will display a menu of notes in your document that have names that begin with "A". Choose the note to which you want to link from the menu; Tinderbox will insert a text link in the text. If you change your mind, press **Esc** to cancel the quick link, or simply click outside the menu without choosing a destination.

Deleting Links

In map view, selecting a note highlights the note's outbound links. Click any link's ⓘ button for information about the link or to edit it, or the link's ⓧ button to delete the link. You can also view outbound links by selecting a note and choosing View ▸ Browse Links; select a link in the Browse Links popover and press Delete to delete it.

Prototypes

Often, the easiest way to describe a note is to explain how it differs from another note. We say that the first note serves as the *prototype* of the second: it shares all the properties of the prototype, except where we specify otherwise.

For example, *Oliver Twist* might be prototype for *Great Expectations*: Great Expectations inherits all the properties of its prototype, save for `Title` and `PublicationDate` which we specify explicitly.

Any Tinderbox note can serve as a prototype for other notes. Prototypes let you specify the default value for an entire class of notes. Whenever Tinderbox checks an attribute that you haven't specifically set, it will use the value from the prototype. Change an attribute in a prototype, and you change it for the notes that use that prototype.

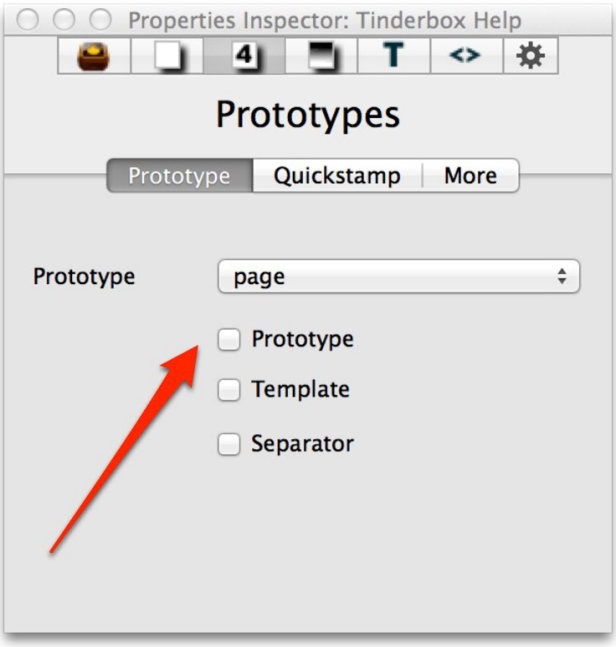
> A few attributes are inherent and are never inherited. For example, moving *Oliver Twist* in the map should not move *Great Expectations*. more on Inherent attributes.

Prototypes are an advanced feature—you may not need to use them—but they're powerful time-savers for complex projects.

To create a prototype,

- create a new note, or select an existing note to serve as a prototype
- Open the Properties Inspector (⌘-1)
- Select the Prototype pane
- Check the checkbox "Prototype"

This new note can now serves as a prototype for other notes.



In outline view the note icon for notes which can serve as prototypes is surrounded by a light green circle.

To use a prototype,

- Select a note in map view, and right-click on the Prototype Tab beneath the note, OR
- Select a note in outline view and right-click on the note's icon, OR
- Select a note, open the Properties inspector, and choose a prototype from the Prototype pull-down menu.

Your note will now use the selected prototype: it will inherit the values for most of its attributes from the prototype note. When you change the value of any of those attributes in the prototype, the value of that attribute in all of the notes that use that prototype will change as well.

Using Prototypes

Any note can serve as a prototype for other notes. You don't need to define prototypes in advance, and you should feel free to create new prototypes as your needs change.

Prototypes save typing. As you take notes, you may find that you sometimes repeat yourself, setting up a series of similar notes that describe similar sorts of things — people, or references, or issues you want to discuss with your staff. Prototypes can help you capture the common elements, making it easier and more pleasant to create notes.

A note can only have one prototype, but each prototype can be used by many different notes.

Prototype inheritance in Tinderbox is efficient; don't worry about memory or performance when using prototypes.

Children of Prototypes

If a prototype is a container, then notes that use the prototype will "inherit" copies of the prototype's children. For example, if Prototype Article contains separate notes for Author Information and Article Text, then making a note into an instance of Prototype Article will create new notes inside the article to hold the author information and the text of the article.

Note that these "inherited" notes are created at the time the prototype is assigned; adding or removing children to the prototype at some later time will not affect notes that already use the prototype.

Inheritance of prototype notes is controlled by the attribute, `$PrototypeBequeathsChildren`. By setting this attribute to false for a specific prototype, passing of children to new instances of the prototype may be disabled. By setting the default value of `PrototypeBequeathsChildren` to `false`, the feature may be disabled throughout the document (unless specifically overridden by a prototype that wants to use it)

Inheritance

Tinderbox notes *inherit* from their prototype; when you give a note a prototype, you're telling Tinderbox, "this note is just like its prototype, except for the differences I tell you about."

Whenever Tinderbox looks up the value of an attribute, it reviews the following checklist:

- If the note has a value for that attribute, that's the value.

Otherwise,

- If the note has a prototype, and if the prototype has a value, then we inherit the prototype's value.

Otherwise,

- If the prototype itself has a prototype, we inherit that value.

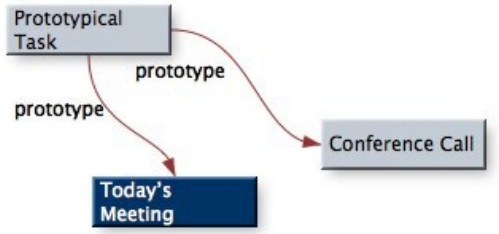
Otherwise,

- We use the default value for that attribute.

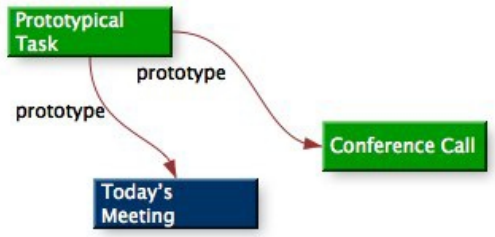
Note that inheritance has nothing to do with the document hierarchy; a note's prototype is not necessarily its container. For example, *Oliver Twist* and *Great Expectations* might be two notes in a document. Each has the prototype "Book". One is inside the container "Living Room Books," the other is inside "Books I've lent to friends."

A note's own values always override inheritance. For example, suppose that note Prototypical Task is gray. We create a new note called Today's Meeting that inherits from Prototypical Task. Initially, Today's Meeting inherits everything from its prototype, so it's gray, too. But if we set the Color of Today's Meeting to blue, it turns blue. Other tasks remain gray.

Now, we make yet another note, Conference Call, which also inherits from Prototypical Task. It, too, is gray, because Prototypical Task is gray.



But perhaps we'd like all tasks to be green; we change the Color of Prototypical Task to green. Now, Conference Call turns green, because it inherits its Color from the prototype. Today's Meeting remains blue, because you gave it a specific color; a note's own values always take precedence over inheritance.



Built-In Prototypes

Tinderbox offers a small number of specimen prototypes built into the application that you can added to any Tinderbox document.

- select the Built-In Prototypes submenu of the File menu.
- choose a prototype to add.
- the selected item is added to a 'Prototypes' container in the current document.

If Tinderbox detects an existing 'Prototypes' note (case-sensitive), the newly added prototypes note is added to that container. Otherwise a new 'Prototypes' container is created at root level in the current document.

User Prototypes

Tinderbox lets you add your own built-in prototypes, helping you create new documents more easily.

Your own built-in prototypes are stored as top-level notes in a Tinderbox document named Prototypes.tbx in subfolder "prototypes" of the Tinderbox support folder.

~/Application Support/Tinderbox/prototypes/Prototypes.tbx

These prototypes will appear at the bottom of the Built-In Prototypes menu. Choosing one of these prototypes will add it to your document's Prototypes containwatcher.

If the prototypes container already holds a prototype with the chosen name, the command has no effect.

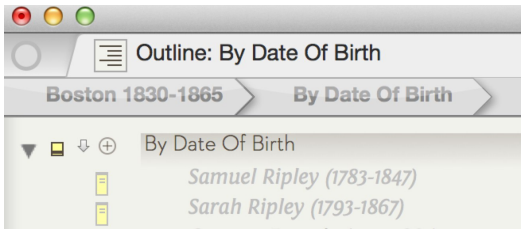
The new prototype shares most attributes with the prototype in the chosen support folder. Exceptions include:

- User attributes that do not exist in your document
- Intrinsic attributes such as Xpos and Container, that are never inherited
- Readonly attributes

The new prototype will have the same initial Height and Width as the original, though you are free to change it.

Breadcrumbs

When a view shows only one section of a document, such as a focused outline, a *breadcrumb bar* appears above the view.



The breadcrumb bar shows each ancestor of the note's presented in the view. In the example above, the notes are contained in the agent named "By Date Of Birth", and this agent in turn is contained in "Boston 1830-1865".

Click on any item in the breadcrumb bar to change the focus from the current container to one of its ancestors. For example, click on "Boston 1830-1865" to see both the agent "By Date of Birth" and any other notes contained inside "Boston 1830-1865."

Using the Tab Bar

Tabs at the top of each Tinderbox window let you switch instantly between different views of your notes. Each tab represents a specific view of a specific section of your document.

Click a tab to select it.

Click the **+** button at the right edge of the tab bar to add a new tab.

Slide tabs left and right to reorganize them in the tab bar.

Pull a tab down to open the tab in its own window.

When the mouse points at a tab, the tab's close box becomes visible. Click the close box to delete the tab.

Right-click the tab for a useful menu, allowing you to change the view type or open a new window.

The View menu's Tab submenu allows you to switch to the next tab (⌥⌘-]), the previous tab (⌥⌘-[), or to create a new tab.

The *link parking space* is found at the left edge of the tab bar.

Using Export

For many export needs, exporting work from Tinderbox is as simple as selecting an option from the **File** menu.

- **Export ▸ As Text** will export your work in a variety of text formats, from plain text to Scrivener, Word, and RTF. You can choose to export an entire document or just some selected notes.
- **Export ▸ As Outline** lets you export the entire document as an outline in a variety of formats, with a variety of outline styles.

At times, however, you may need to export a specialized format used by another program, or you may want to include the values of specific attributes in your exported work. Or, you may want to export different kinds of notes with different formats. Tinderbox's HTML export provides a flexible way to export information just the way you want it. HTML Export can also be configured to create XML, OPML, LaTeX, and a variety of other documents.

When using HTML Export, you create a special **template note** that shows Tinderbox how you want things to be exported. Tinderbox provides a selection of **Built-In Templates** in the File menu to help you get started. Templates contain *export placeholders* that are replaced by information taken from the exported note. Placeholders begin with the caret sign **^**; common placeholders include

^title^

^text^

^value(*expression*)^

^include(designator *,template*)^

Here, an *expression* is typically an attribute value like $Width or $Address, and a *designator* is the name of another note or a keyword like **this** or **parent** that designates a specific note.

HTML and Preview

Tinderbox's text pane provides two additional views for working with export. These are normally hidden to conserve screen space; to display them, choose **Show Text Pane Selector** from the **Window** menu.

The **Text Pane** is already familiar; it shows the text of your note.

The **Export Pane** shows you exactly how Tinderbox would export the selected note.

The **Preview Pane** shows you how the exported note would be rendered in a Web Browser.

More Commands

Footnote

The **Footnote** commands let you create a new note for any word or phrase you have selected in the text pane. The name of the newly-created note is taken from the selected word or phrase. A text link leads from the selected text to the footnote, and a basic link is created from the footnote back to the currently-viewed note.

After you a footnote, the current selection shifts to the newly-created footnote; after writing the footnote, use the Note menu's Navigate command (⌘-return) to follow the basic link back to the note you were editing

The Note menu's **Add Footnote As Sibling** commands adds a new note as a sibling of the current note. The **Add Endnote** commands adds the footnote inside a container named "Notes" that is created as a sibling of the current notes; many notes can share the same Notes container.

Inspect Stamps

Stamps apply a Tinderbox action to the currently-selected notes. Stamps are valuable for actions that you may want to do frequently, but that aren't suitable for automatic application through agents or OnAdd actions. For example, if you often want to make notes red, you could define a Stamp named "red" with the action `$Color="red"`.

To make or edit your stamps, choose **Inspect Stamps…** from the Stamps menu or open the Document Inspector and select the Stamps pane.

To use a stamp, select one or more notes and choose the Stamp you want to use from the Stamps menu.

You can move stamps to a different document by dragging them out of the stamp inspector and dropping them in the new document's view pane. Note that, if the destination document already has a stamp with the same name, the drag will be ignored. If the stamp action refers to user attributes that don't exist in the new document, those actions will have no effect until the attribute is defined.

Using Maps

The Tinderbox map view shows all the notes inside a container. You can arrange notes and containers as you like, perhaps placing related notes close together.

Map views provide great scope for expressing tentative, preliminary, or provisional relationships among your notes, and for discovering emergent structure. Tinderbox provides a host of visual dimensions — shape, color, borders, badges, and many more — that you can use to express the properties of notes.

To move a note in the map, simply drag it. Hold down the shift key while dragging to move it only horizontally or only vertically. As you move the note, guidelines will indicate opportunities to align the note with other notes in your map.

Hold down the option key and drag a note to make a copy of the note. Holding down both the shift and option keys while dragging will make an alias of the note.

If a note has a Fill or Texture, hold down the shift and command keys to move the fill image.

To resize a note in the map, select it and drag any of the resize handles. Shift-drag the resize handles to preserve the note's shape or aspect ratio while changing its size.

To place a note inside another note or into a container, simply drop it inside another note or on a container. You can also drag notes out of a container, or drag the background of a container to adjust the part of the container that is visible.

To create a note in a map view

- double-click where you want to place the note, OR
- select a note and press Return (↩) to make a new note,

  OR
- choose Create Note… (⌘K) from the Note menu,

  OR
- right-click where to want to place the note and choose Create Note… from the menu

To rename a note in map view

- select it and ⇧⌘-Return

  OR
- select it and press Function-Return (fn↩)

  OR
- click and hold the note for a moment, and edit the note's name in the map.

Moving Around In Maps

Use the hand cursor to move the window. Simply click and drag the map background.

Use the horizontal and vertical scroll bars to move around in a view window. Use the scroll wheel, or two-fingered drag on the trackpad, to scroll the window.

To scroll directly to a specific note, simply type the first characters of the note's name.

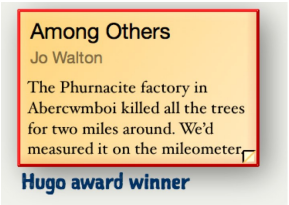
To see an overview of the entire map, press and hold the ctrl, option, and command keys simultaneously. To zoom to a different part of the map from the overview, move the mouse cursor to the area of the map in which you are interested before you release the ^⌘⌥ keys.

Subtitle, Caption, and Text

A note always displays its title — either its $Name or, if it has a display expression, the result of evaluating $DisplayExpression. If the title does not fit in the available area, Tinderbox will draw a truncated title.

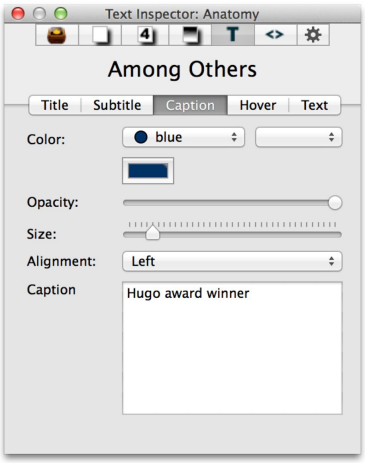
If there is sufficient space, Tinderbox will also draw that note's subtitle beneath the title. If the note has a caption, the caption is drawn beneath the note.

If there is even more space, Tinderbox will draw the note's text beneath the title and subtitle.



- Details on displaying note text in the map

Subtitles and captions may be entered and modified in the Text Inspector.



You may also edit subtitles and captions by clicking them while a note is selected in the map. When editing the title, use tab to move to the subtitle and caption.

The appearance of the subtitle is controlled by the attributes $SubtitleColor, $SubtitleSize, and $SubtitleOpacity. The subtitle is most easily added or changed in the Subtitle panel of the Inspector (⌘-1).

Changing fonts

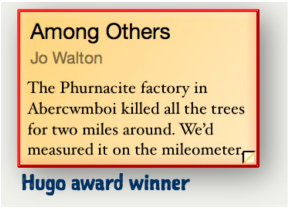
To change the typeface used for a field in a note

- Select the note.
- Click on the field you want to change until it becomes editable.
- Choose Format>Font>Show Fonts or press ⌘-T.
- Select the font and size you prefer.

In Tinderbox, italic fonts are usually reserved for aliases.

Text In Maps

Normally, Tinderbox displays the note's title in the map. If a note is sufficiently large, Tinderbox will display some or all of the note's text as well as the title.



The text display is a scaled representation of the note's text. A suitable size is normally chosen automatically, but may be overridden by setting the attribute `MapBodyTextSize`.

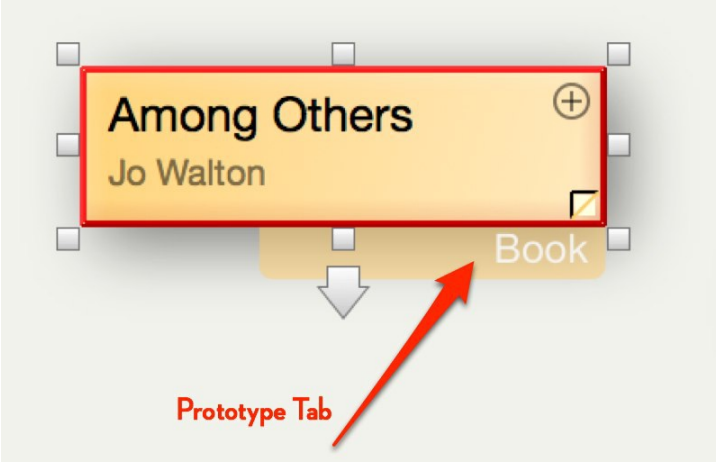
If $MapBodyTextSize is 0, Tinderbox chooses a suitable font size.

If $MapBodyTextSize is -1, no body text is drawn.

Otherwise, $MapBodyTextSize is the font size, in points, for drawing the note's body text.

The Prototype Tab

When a note is selected in map view, a Prototype Tab appears beneath the note and shows the name of the note's prototype.



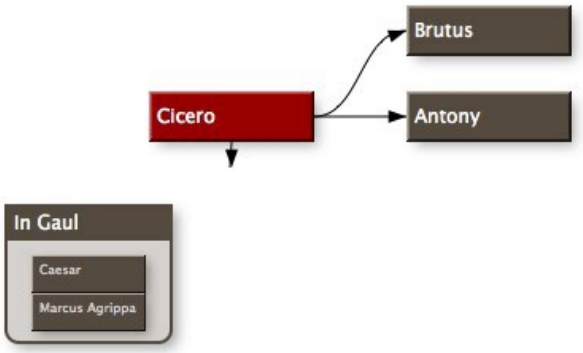
If the document has no prototypes, no prototype tab is displayed.

To change a note's prototype, right-click the prototype tab for a menu of available prototypes.

Links

A map view shows the links between notes.

If both the source and destination of the link are in the map, the link is drawn as a line between the boxes. If the source or the destination of the link are outside the map, the link is drawn as a short arrow departing from or arriving at a note.



In this map, Cicero is linked to Brutus, Antony, and Caesar. Because Caesar is inside the container "In Gaul", only the start of the link to Caesar appears in the map.

To make a link, select a note. The selected note's arrow-shaped *link widget* will appear beneath the note in map view, or next to the note icon in outline and chart view. Drag a link out of the link widget, and connect it to any other note.

In map view, you can create a new linked note by dragging the link from the link widget and releasing the link where you want to create the new note.

If the destination of the link isn't handy, drag the link to the *link parking space*. Then, locate the destination and drag the parked link from the parking space to its destination.

Dancing

In map view, View ▸ Arrange ▸ Dance (⇧⌘-D) initiates an automated layout of the view based on a physical simulation.

- Each link among notes in the map is treated as a spring that pulls linked notes together.
- All notes exert a gravitation attraction for other notes.
- Notes that overlap repel each other.
- At the beginning of the simulation, each note is subject to a random force, much as if it were heated. This force is reduced progressively over time. This process, known as simulated annealing, helps the simulation from getting tangled up in local minima.

Not all maps will benefit from automatic layout; the famously tangled map of Mary-Kim Arnold's "Lust", for example, does not. Performance may be unsatisfactory in maps with more than a few dozen notes. Nonetheless, this may prove useful in many cases.

Adornments

Adornments are labels that you can add to the background of a map view to help organize the map. You can adjust the size, placement, color, and text color of an adornment. Maps may have many adornments, or none. Adornments do not appear in any other views. Adornments can serve to label neighborhoods in a map view, but they do not affect the hierarchy or the links of any notes.

To create an adornment

- choose Create Adornment… from the Note menu.

  OR

- right-click in the background of a map view and choose Create Adornment….

Image Adornments

You can use adornment to place images in the background of the Tinderbox map.

To create an image adornment

- drag an image file into the Tinderbox map

Locking Adornments

It it often convenient to lock the position of large adornments. The attribute `$Lock` fixes an adornment's (or note's) position in the map, preventing it from being inadvertently moved.

To lock (or unlock) an adornment

- select the adornment
- click the Lock icon,

  OR

- select the adornment
- click the **Lock** checkbox in the More pane of the Properties Inspector.

To make an adornment "sticky"

If an adornment is sticky, objects lying on the adornment will stick to the adornment when the adornment is moved. This makes it much easier to work with large, complex maps, allowing you to move whole sections of notes together.

To toggle 'sticky'

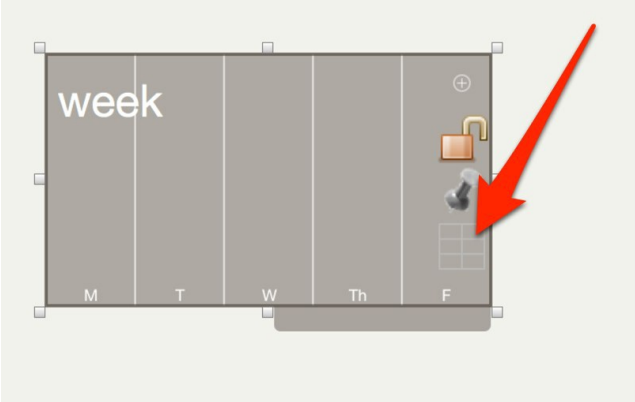
- select the adornment
- click the pushpin icon to make the adornment sticky,

  OR

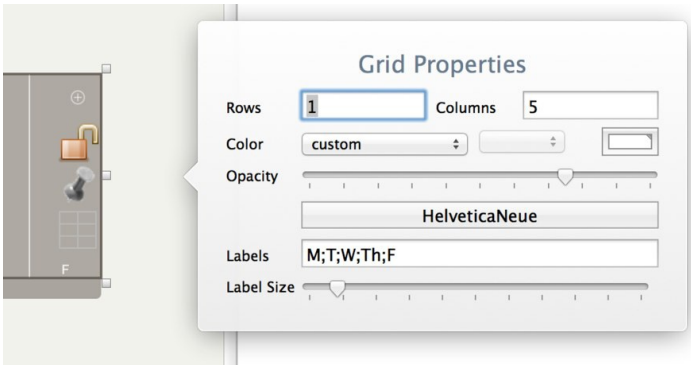
- select the adornment
- click the **Lock** checkbox in the More pane of the Properties Inspector.

Grids

Adornments may have an optional *grid,* letting you divide the adornment into equally-sized rows, columns, or into both rows and columns. For example, you might choose a grid of five columns, labelled “Monday” through “Friday”



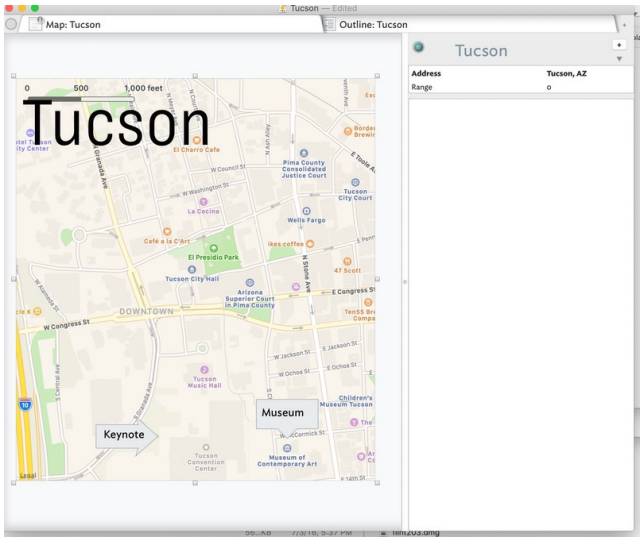
To add or change a grid, click on the adornment’s Grid Button. The grid popover lets you select the color and appearance of the adornment grid. If you wish to add labels to the grid, enter the labels as a list, separated by semicolons, from left to right and from top to bottom.



Note that grid properties are controlled by corresponding attributes, such as $GridColumns, $GridRows, $GridColor, and $GridLabels. Thus, grid properties may be inherited from prototypes or altered by rules or actions.

Geographic Adornments

Geographic Adornments are adornments that depict maps.



An adornment is geographic if all of the following are true:

- it is sufficiently large
- its latitude and longitude are not zero

To make a geographic adornment, simply make a large adornment and set its $Address. The map will show the area around the designated location. (This may take some time and requires internet connectivity.)

The scale of the map is determined by the adornment's $Range attribute, which represents the approximate size of the map in kilometers.

If a note lies on a map adornment and has a latitude and longitude, it will be placed near the corresponding place on the map adornment.

Image Adornments

Images may be inserted into the background of a map view as image adornments. Like all adornments, image adornments do not appear in any other view and don't affect the hierarchy or the links of any notes. Adornments are never exported.

To create an image adornment, drag an image file into a Tinderbox map view.

Images are inserted and displayed at 'actual size'. Resizing the adornment will crop the image. If a shape is set for the adornment, the image is cropped by the shape's boundaries.

The $Opacity attribute is supported for image attributes, allowing image adornments to be translucent.

Smart Adornments

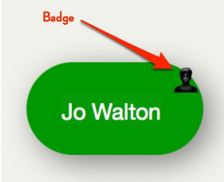
Tinderbox "smart adornments" have a query, just like agents.

The smart adornment searches within its map—that is, amongst its siblings in the document outline—for notes that satisfy its query. Matching notes will be moved to the smart adornment, and notes that do not match the query will be moved off the adornment. (Locked notes and adornments are never repositioned by a smart adornment.)

Smart Adornments are handy for automatically organizing containers to which you are frequently adding notes—especially if these notes are added automatically. When you add notes yourself, you naturally place them appropriately, but when notes are being moved around your Tinderbox document by agents or inserted from email, smart adornments can help keep things organized.

Badges

Badges are iconic images used to distinguish some Tinderbox notes.



To add a badge to a note

- Select a note
- click the badge (if the note has one) or the badge button in the note's upper right-hand corner
- Choose a badge from the popover

User badges

Tinderbox comes with a collection of built-in badges for a variety of uses. In addition, you can easily add your own badges.

Badges are .png image files stored in the Tinderbox support folder. A size of 32px x 32px is recommended; larger images will be scaled automatically.

To add a new collection of badges, simply add a folder of badges to the support folder. To add an individual badge, simple add it to the badge folder; it will automatically be added to the User badge collection.

The badge folder is located in your Library folder:

`~Library/Application Support/Tinderbox/badges/`

You can view the support folder by choosing **Reveal Support Folder in Finder** from the Help menu.

Flags

One common Tinderbox task is qualitative analysis of existing materials, such as letters, surveys, diaries, and personal papers. An important preliminary step in this work is coding —identifying occurrences of special interest for the study. For example, if we were analyzing a collection of nineteenth-century diaries to study what people recorded about food and drink, we might want to code where the food was consumed. We might mark every passage that discussed eating at home with the code P1, eating at the residence of another family member with the code P2, eating at a pub with P3, and so forth. We might also note places where money is discussed: C1 might indicate that the writer paid for their meal, C2 that someone else explicitly paid for the writer's meal, and so forth.

Flags offer a convenient and flexible way to foreground selected codes in map view. $Flags is a new set attribute; when not empty, small "flags" are displayed above the note in map view. (Flags do not appear in other views).

For simple coding tasks, using $Badge may be adequate. Flags provide a wider range of visual cues, and new flags can be improvised quickly when coding needs change.

Note that $Flags is a list attribute.

Specifying Flags

Flags are described using a concise textual shorthand. (A visual flag editor is planned for the future.)

The simplest flag is simply a color. The flag "red" looks like this:

For horizontal stripes, list the colors separated by hyphens: red-white-blue

For vertical stripes, list the colors separated by the vertical bar character: blue|white|red

red|white|blue|white|red

Diagonal stripes use the '/' symbol: yellow/black

A checkered flag uses the '$' symbol: black$white

A variety of symbols may be overlaid on a flag. A cross, for example, uses the '+' sign: white+red

white+yellow/black

For a saltire, use an asterisk "*": white*light blue

For a diagonal line, use the percent sign: white%light blue/red

The ">" symbol adds a chevron:

The "}" character adds a pall.

The pall and chevron work together: white}green>blue

Finally, the period "." adds a short textual annotation. A1.red

The text color is normally white, but may be specified: black:C.lighter blue

The text may be an emoji: .green

The color "none" represents a transparent flag. black:T1.none

Web-style colors #RRGGBB and #RGB may be used in place of symbolic colors. #FF0 displays a bright yellow flag.

Cleanup

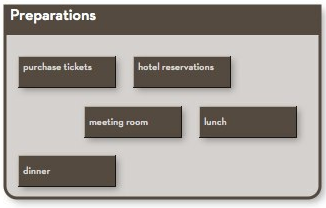
The Cleanup commands appear in the View menu, and are also available by right-clicking in the background of the map.

You can arrange the notes in a grid, a staggered grid, a row, a column, or a box.

Like Smart Adornments, automatic cleanup can help organize cluttered maps.

Containers

In map view, containers display a miniature view of their contents.

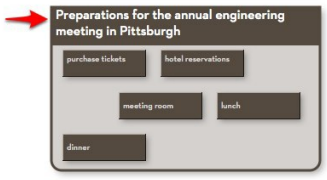


Containers are always drawn with a distinctive shape with the title bar at the top and rounded corners at the bottom. (Agents have rounded top corners and a title at the bottom).

The relative scale of the interior of a container is set by the attribute `$InteriorScale`. Typically, $InteriorScale has a value of 1.5; notes inside a container are drawn ⅔ the size of normal notes.

You can drag notes into the body of a container, or drag them out of the container to move them into the current map. Dragging a note into a container makes it the first child of the container; to add a note as the last child of the container, hold down the ⌘ and shift keys while dragging the note.

Normally, container titles are a single line, drawn at the top of the container. If you wish to expand the size of the title bar, simply drag its lower edge (for containers) or upper edge (for agents).



Display Expression

In maps and other views, Tinderbox typically displays the note's title ($Name) to identify the note.

Sometimes, you may want to display additional information: how much text the note contains, who is responsible for a project, or when a task is due.

The attribute $DisplayExpression lets you change the note's label without changing the value of its $Name.

The $DisplayExpression attribute is simply an expression, just as used in Tinderbox rules, container actions, and agent actions. Whenever Tinderbox needs to display the note, it evaluates the note's $DisplayExpression and displays the result. If $DisplayExpression is empty, Tinderbox displays the note's $Name.

For example:

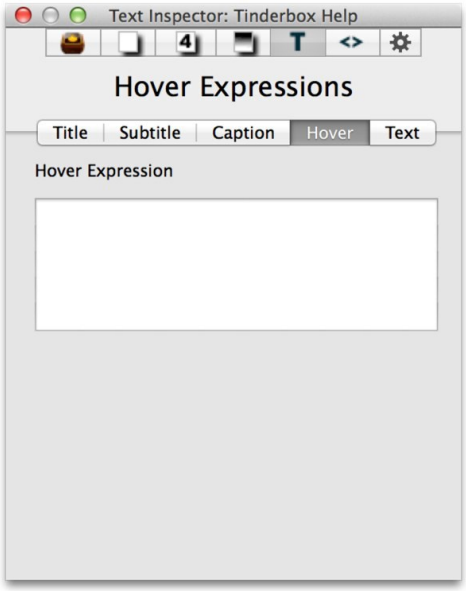
DisplayExpression	typical display
$Name+" ("+$WordCount+" words)"	Chapter 3 (7500 words)
$Name+":"+format($DueDate,"l")	Return books to library: 12/30/2008

Hover Expression

You may display additional information about a note when the cursor hovers over the note. The hover expression is only displayed in Maps and Outlines.

The attribute $HoverExpression lets you specify the information that will be shown while hovering. A note may have its own hover expression, but often will inherit hover expression from a prototype.

To add or change the hover expression, open the Text inspector, and switch to the Hover pane



The $HoverExpression attribute is simply an expression in action code, just as used in Tinderbox rules, container actions, and agent actions. If $HoverExpression is empty, no hover expression is displayed.

For example:

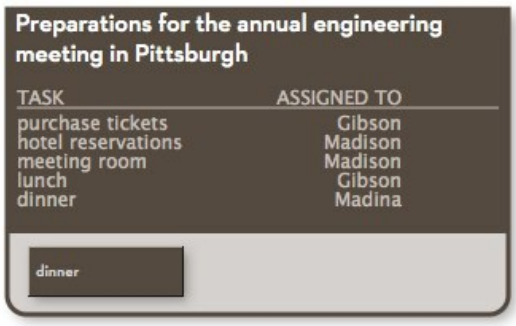
HoverExpression	typical display
$WordCount+" words" "	7500 words
$WordCount+" words\n\n+$Text"	7500 words Now is the winter of our discontent…

The attribute $HoverOpacity allows you to customize the opacity of the displayed hover expression on a scale from 0 (invisible) to 1 (opaque). The default value is 0.5.

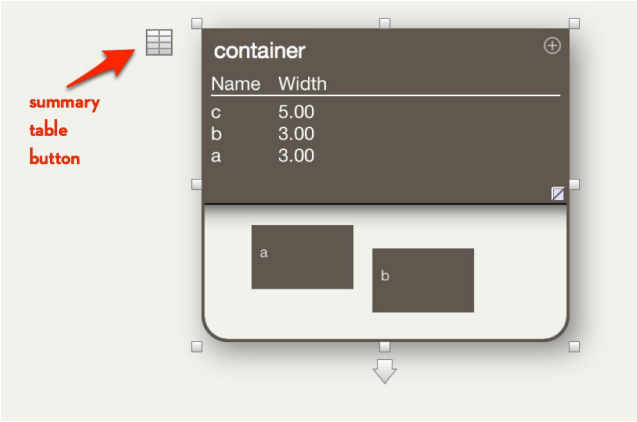
For more information on actions and expressions, see Actions, Expressions, and Rules.

Summary Tables

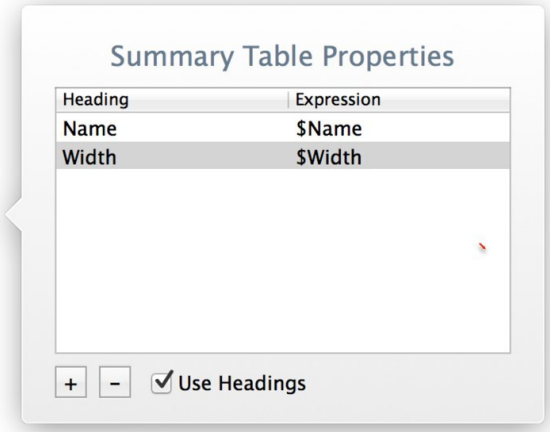
Containers and agents may display a *summary table* of their contents if their title bar is tall enough to accommodate it.



To define a Summary Table for a container, expand the container's title bar and click on the container's summary table button.



In the Summary Table popover, use the "+" button to add rows, the "-" button to delete rows, and edit the row names and expressions as you prefer.



Attributes and the Summary Table

The contents of the summary table are determined by the attribute `$TableExpression`, which should evaluate to a string using the vertical bar character "|" to separate columns. For example:

```
$Name+"|"+$AssignedTo
```

will display the name of each note and the person to whom the note is assigned.

The table heading is taken from the attribute TableHeading; for example:

```
TASK|ASSIGNED TO
```

Tinderbox displays one line for each child of the agent or container, until it runs out of space or lists each child.

If the table expression for a child note evaluates to the empty string "", that row of the table is skipped.

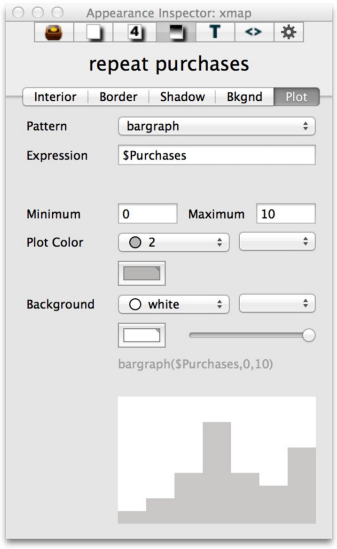
If the table expression for a child note evaluates to the string "-", a single hyphen, that row of the table is replaced by a horizontal rule.

Histograms

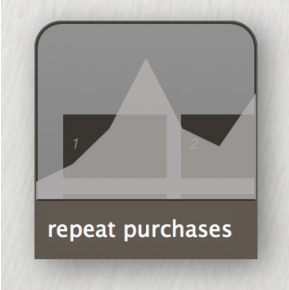
Tinderbox containers can display a variety of graphs to describe their contents. For tabular summaries of the contents of a container, see Summary Tables.

Here, an agent has gathered notes concerning the number of repeat purchases made by selected customers, and sorted them by the time since their most recent purchase. The newest customers come first; since they're new, few have had occasion to repurchase the project. Interestingly, though older customers have repurchased more often, it's clear that some improvement made about half-way through the product's life makes those users more eager to buy more.

To add a plot of histogram to a container, select the container and open the Plot pane of the Appearance Inspector.

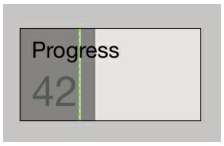


First, choose the sort of plot you prefer: a line graph or plot, a histogram, and x/y plot, or a pie chart. Then, choose what attributes (or expressions based on attributes) you wish to display. You may also set the axis limits, the colors of the data and background, and the opacity of the background. If the plot background is not completely opaque, the container's contents are drawn behind the plot.



Progress Bars

Tinderbox notes may display horizontal or vertical progress bars to track the state of any numerical attribute. The progress bar is specified in the note's Pattern attribute and is most easily set using the Plot pane of the Appearance inspector.



Progress bars may be horizontal (bar) or vertical (vbar) and fill the body of the note. The progress bar always has a **value** which is to be represented by the bar; it may be a fixed number or an attribute expression. For example, if the value is $WordCount, the progress bar will represent the number of words in the note's text.

Progress bars may have optional *minimum* and *maximum* values. If no minimum is specified, Tinderbox assumes that 0 is the minimum. If no maximum value is specified, Tinderbox assumes that 100 is the maximum.

Progress bars may also display a *target line* for any chosen value that lies between the minimum and maximum values.

Using Outlines

An outline view shows the hierarchical structure of the document as an indented outline. It is easy to restructure the hierarchy of the document in an outline window, by dragging parts of the hierarchy to other parts of the hierarchy.

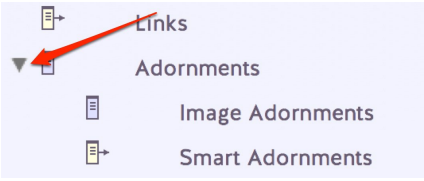
The outline view does not show any of the links in the document; use maps to explore links.

Outlines are best when you understand the structure of your information, and when it's easy for you to put everything in its proper place. If you frequently find yourself puzzling over where to put a note, consider working in Map view for a while. You can always reorganize later.

You can have several outline windows open in different tabs or even in different windows, each focusing on different portions of your document.

The disclosure triangle

A container — a note that contains other notes — appears in an outline view with a small triangle to its left.



When the triangle points down, the outline lists the children of the note. When the triangle points right, the children are hidden. A container which is hiding its children is sometimes said to be *collapsed,* and a container that displays its children is said to be *expanded.*

- click the disclosure triangle to expand or collapse the item.

  OR

- with a note selected, ⌥→ expands the item, and ⌥← turns it collapses it.

- choose Expand from the View window to turn the expand the selected item.
- hold down the ⌘ key while you click a disclosure triangle to expand or collapse that item and also all its younger siblings.
- hold down the ⌥ key while you click a triangle to toggle that triangle, and also expand or collapse all the descendants of that note.

Editing a note title

To edit a note's title in an outline, simply click and hold the title. After a brief delay, a frame will appear around the title and you may revise the title at you wish. The click-hold delay is the same duration as that used to trigger a file rename in Finder

Several key shortcuts are available when using the in-place note editor:

- Escape (⎋)will cancel editing and restore the previous title.
- Return (↩) will confirm the new title and terminate editing. Press Return a second time to create a new sibling note.

You may also finish editing by:

- clicking on the note's icon,

OR

- selecting a different note

Separators

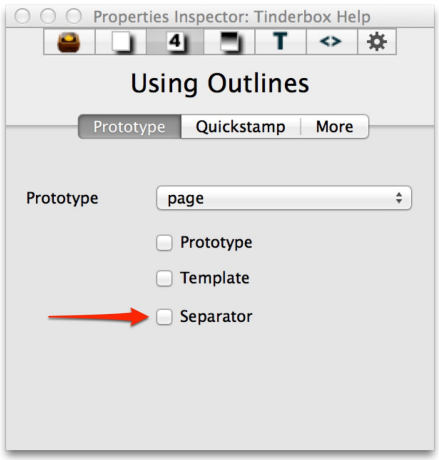
A *separator* is a horizontal line in the outline view that may be colored and titled like other notes via the Note and Colors menus.

To add a separator:

- choose Create Separator… from the Note menu

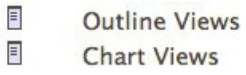
  OR

- select any note
- open the Prototype pane of the Document inspector and check **Separator**.



The Outline Icon

The “icon” that appears next to the disclosure triangle in the outline conveys a host of information in a very small space.



- The Color of the icon’s border is the color of the note.
- The “title bar” of the icon indicates whether the note is a container (title at the top) or an agent (title at the bottom). If an agent is turned off, it's title bar is hollow.
- The text lines inside the icon indicate whether the note contains any text, and roughly how long the text is.
- The interior color of the icon reflects how recently a note has been edited. A newly-created note is light blue. Over the course of a day, the note “dries” to a neutral white, and then over the span of a year it gradually yellows.

Right-click the icon to displays the note’s Prototype menu.

Double-click the icon to *focus on* or “hoist” the note, focusing the window on the note and its descendants.

The Link Widget

Focusing or “hoisting” a note focuses the view on it and its descendants. Expanding the view returns the focus to the the note’s container. These commands are available in chart and outline views.

- select Focus View from the View menu

  OR

- select Expand View from the View menu

Identifying prototypes

In outline view, the note icon for notes which can serve as prototypes is now surrounded by a light green circle. This helps distinguish notes intended to serve as prototypes. See the Prototypes section of the Notes chapter for more information.

Focusing On Notes

You can drag notes in an outline window to rearrange the document’s hierarchy.

To move a note:

- Drop a note on the right half of another note to make it a child of the note you drop it on.
- Drop the note while it is over the left half of another note to make it a sibling of the note you drop it on.

As you drag, a blue highlight indicates where the dragged note would be moved. You can’t drag a note onto an alias, an agent, or onto one of its own descendants.

Indenting notes in the outline

- Press Tab (→|) to indent the selected note one more level in the outline. If the note cannot be indented, Tab has no effect.
- Hold down the Shift key while you press Tab (⇧→|) to unindent the selected note out one level in the outline.

Indent (⌘-]) and Unindent (⌘-[) are also available in the Note menu.

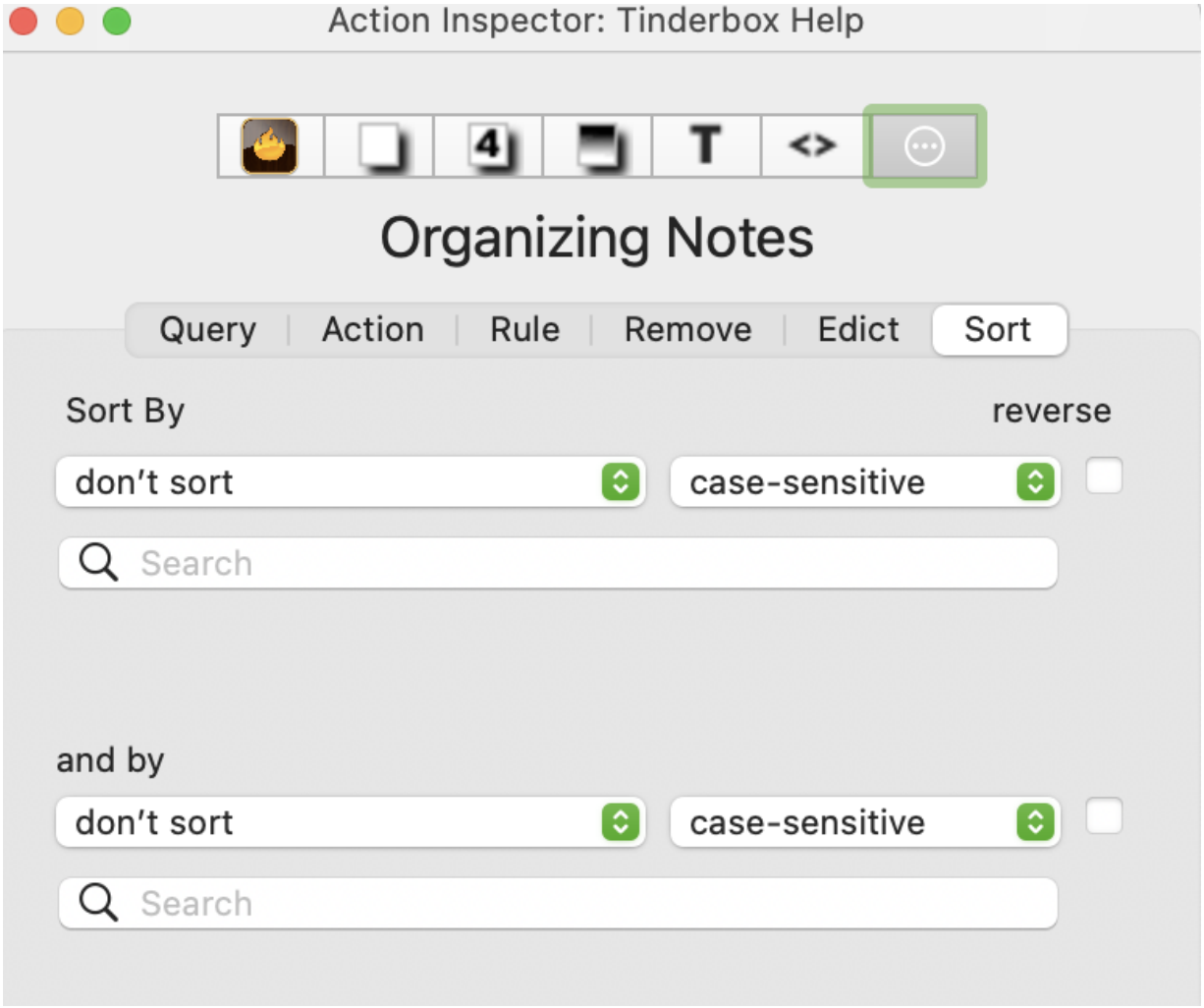
Organizing Notes

Outlines may display a checkbox to the left of each item.

> To display checkboxes, choose **Use Checkboxes** from the View menu.

The checkbox of an item is associated with the item's `$Checked` attribute. Checking an item sets the value of $Checked for that item to `true`.

Any container may be **sorted** by choosing a sort method in the Sort pane of the Action Inspector.



You can sort by any attribute, though sorting by Name is most common. Sorting may be case-sensitive or case-insensitive. The sort method **library sorting**, permits notes to be sorted as they might be filed in a library. By default, library sorting ignore initial words "a", "an", and "the". The note **/Hints/library sorting/** , if present and not empty, contains a list of words that will be ignored for library sorting, each on a separate line. You can customize this note for different languages and filing practices. Library Sorting also respects some local rules for handling dipthongs and diacritics; in English, for example, accented characters are filed as if they were not accented. Library Sorting is always case insensitive.

Columns

Outlines may be **filtered**, allowing you to see only those notes that meet a specified criterion. For example, you could show only notes created in the last month, or only notes that mention "Roosevelt", or only notes that received a grade of A or A-. These notes and their ancestors will appear in the filtered outline; all other notes will be hidden.

To set the filter for an outline view, select an outline view and choose **Filter** from the **View** menu.

When a view has a filter, notes appear if they satisfy the filter expression, or if a descendant of that note satisfies the filter expression. When the filter is active, notes cannot be collapsed and new notes cannot be created.

If the filter expression is empty or invalid, every note satisfies the filter expression.

The filter bar displays the number of items that match the filter's query. Not that this is not identical to the number of items displayed in the outline, since an item appears in a filtered outline if it matches the query or if one of its descendants matches the query.

Filtered Outlines

Outlines may be **filtered**, allowing you to see only those notes that meet a specified criterion. For example, you could show only notes created in the last month, or only notes that mention "Roosevelt", or only notes that received a grade of A or A-. These notes and their ancestors will appear in the filtered outline; all other notes will be hidden.

To set the filter for an outline view, select an outline view and choose **Filter** from the **View** menu.

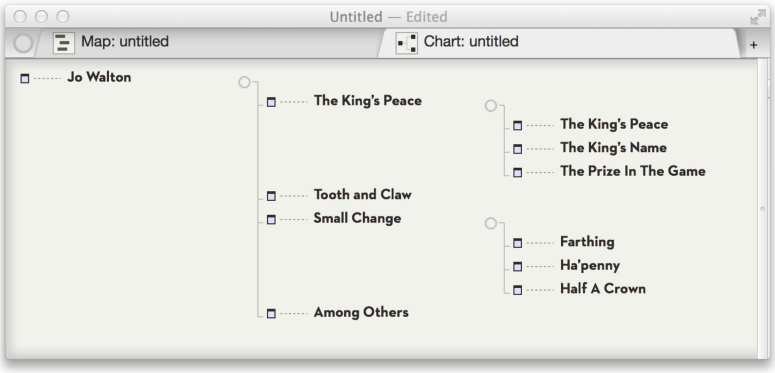
When a view has a filter, notes appear if they satisfy the filter expression, or if a descendant of that note satisfies the filter expression. When the filter is active, notes cannot be collapsed and new notes cannot be created.

If the filter expression is empty or invalid, every note satisfies the filter expression.

The filter bar displays the number of items that match the filter's query. Not that this is not identical to the number of items displayed in the outline, since an item appears in a filtered outline if it matches the query or if one of its descendants matches the query.

Using Charts

A chart view shows the hierarchical structure of the document. A chart is also handy for seeing many notes in one window. It is easy to re-structure the hierarchy of the document in a chart window, by dragging parts of the hierarchy to other parts of the hierarchy.



Notes in chart view display their badges, and long titles will wrap to multiple lines. A chart view does not show any of the links in the document.

The disclosure triangle

As in outlines, a disclosure triangle allows you to expand to collapse an container or agent.

Focusing the View

Focusing a chart view lets you focus on a specific part of your work. It may be useful to keep several different charts open in separate tabs or even separate windows.

To focus on a note in chart view,

- Double-click the icon of the note to be focused on, OR
- Select the note and choose Focus View from the Note menu

Organizing Notes

You can drag notes in a chart window to rearrange the document's hierarchy.

- Drop the note on the right half of another note to make it a child of the that note.
- Drop the note while on the left half of another note to make it a sibling of the that note.

If you drop the note elsewhere in the chart window, it will snap back to its starting location. The drag highlight indicates where the dropped not will move. Notes cannot be moved inside one of their descendants, inside an agent, or inside an alias.

Using Timelines

Timelines display all the notes in a container which have a `$StartDate`. If a note has both a `$StartDate` and an `$EndDate`, both the start and end of the note are drawn.

Dragging notes in the timeline will change their `$StartDate`, their `$EndDate`, or both.

Timeline notes are arranged in vertical bands, according to `$TimelineBand`. Drag notes to a new band to change `$TimelineBand`.

Links among notes visible in the timeline are drawn in the timeline; other links are ignored.

Using Attribute Browsers

The Attribute Browser shows you the contents of any container in the current document, categorised by the values of a chosen attribute. By default the scope is whole-document.

The scope and appearance of the view are set via the Attribute Browser controls at the top of the view pane. By default all expect the first row of controls are hidden. The remainder of the controls can be revealed/re-hidden via a disclosure triangle on the control panel.

The Attribute Browser can use column view and as in other views the column data is editable.

For Set and List data types, categories are still single value notes list in every category (value) for which the note has a value of combination of values. Thus a note with more than one value may list several times.

Optionally, categories can show a count of the number of the items. By default this is for the view's chosen attribute but can also be for any of the attributes also displayed via column view If enabled, this summary figure is shown at the right end of the category bar. For Number-type attributes, a range of mathematical optional are also offered.

Dragging a note to a new category changes the note's value for that listed attribute to that of the category. For multi-value attributes, regardless of the category the dropped note receives all of the values for the attribute for the note listing after the dropped note. In such circumstances it can be useful to list the attribute as a column view item to see the full range of values allotted.

Items listed are drawn in `$Color` using `$DisplayName`. Badges are shown, and a link-drag widget is shown after the item name. The icon used for each item is as per Outline view, indicating the degree of text, recentness of edit and in/outbound links.

A link widget for dragging links is drawn to the right of the title of the selected note.

Using Hyperbolic View

The hyperbolic view (**View ▸ Hyperbolic**) shows notes that are linked to or from a specific note — the ***focus*** note. The initial focus note is the selected note when the view was activated.

Starting at the focus note, Tinderbox identifies all the notes that link to that note, and all the notes to which the selected note links. This is repeated until all notes reachable from the selected note, and all notes from which the selected note can be reached, are included. Those notes are then arranged in a view with the selected note at the center. Notes linked to that note are arranged radially around that note, and notes linked to those notes are arranged radially around them.

Finally, this tree is projected onto a hyperbolic plane; notes near the center are larger, and notes further from the center are progressively smaller.

Dragging the background of the hyperbolic view moves all the notes, letting you examine different parts of the link structure.

Clicking a note selects that note.

Right-clicking or control-clicking a note displays a menu that lets you see and follow links from that note, select or focus on that note, and performs various other common activities.

Double-clicking a note makes that note the focus and redraws the view.

To make a link, drag any note and then drag the link to its destination, or to a link parking space.

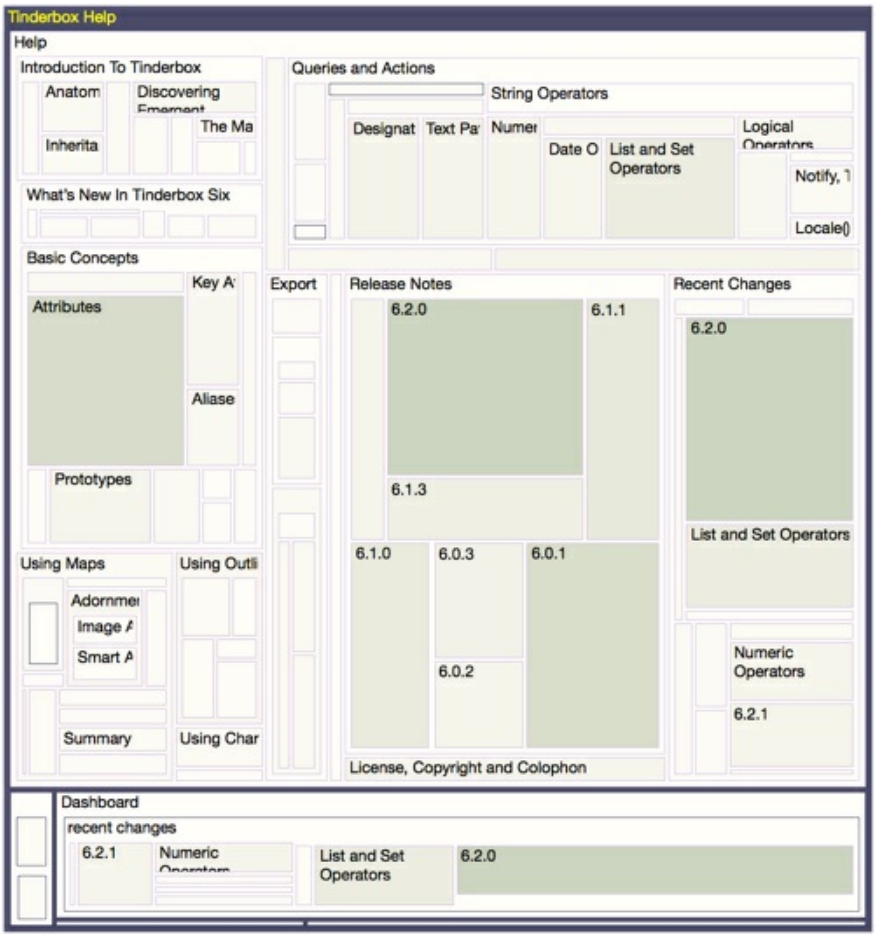
To make a new note linked to an existing note, drag a link from the starting point and release the mouse over any vacant place.

Hovering the mouse over a note will reveal its full name as a tooltip. This is especially useful out near the rim.

Web Links and Prototype links are ignored in hyperbolic view.

Using Treemaps

A **treemap** shows the hierarchical structure of the document as a set of nested rectangles. Treemaps can display even more of the document hierarchy than chart view, permitting hundreds of notes tp be browsed in a single view.



To select any note in a treemap, click on the note's title.

Focusing the View

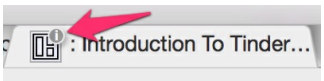
Focusing a treemap view lets you focus on a specific part of your work.

To focus on a note in chart view,

- Double-click the icon of the note to be focused on, OR
- Select the note and choose Focus View from the Note menu

Treemaps do not display adornments or separators. In addition, if a note is too small to display at the current window size and scale, that note will be omitted.

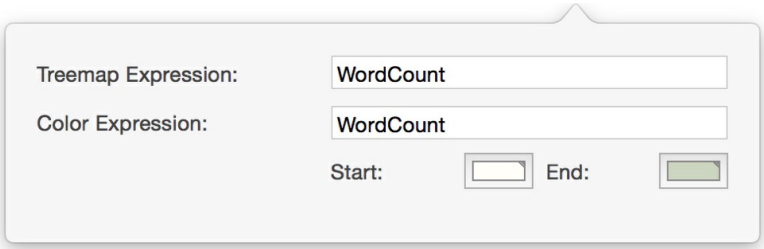
Visualization and Treemaps



New treemaps assign equal weight to each note they display. You can instead set the area of each note to reflect some attribute or combination of attributes of the note. In addition, each note can be colored to reflect an attribute or combination of attributes of the note.

To customize a treemap visualization, click on the "i" button in the treemap tab.

This will display the treemap popover.



The **Treemap Expression** adjusts the relative area of each note. If the value of a note's $TreemapExpression is less than or equal to zero, the note will be omitted entirely; otherwise, the note's area will be proportional to the value of TreemapExpression. TreemapExpression is frequently simply an attribute value, such as $Price or $WordCount.

The **Color Expression**, if specified, adjusts the color applies to each note. The expression is evaluated for each note in the view, and is chosen from a

linear gradient between two colors, the **Start Color** and the **End Color**. If the Color Expression is $WordCount, then notes with the smallest word count will use Start Color, those with the largest $WordCount will use End Color, and intermediate notes will use a blend of StartColor and EndColor.

Using Crosstabs

portion of your document. It displays a table in which one attribute's values are listed horizontally and the second attribute's values are displayed vertically. The following attribute types are supported: numeric, string, date, boolean, list and set.

The attributes to be viewed are displayed in two buttons at the top of the view. Press either button to change attributes. The popover window also lets you set the maximum number of rows or columns to use; setting the limit to the upper bound (25 rows or columns) asks Tinderbox to display a separate row or column for each unique value.

The notes viewed in the crosstabs view may be all the notes in the document, or may be restricted to those inside a specific container. Select the desired container from the Container popup menu.

The crosstabs view may be further restricted to those notes which satisfy a query.

The Style popup lets you choose to display either the count of notes in each cell or the percentage of notes. An optional heatmap will color cells in proportion to the number of notes they contain, using the system accent color for sparsely-populated cells and the complement of that color for heavily-populated cells.

At the bottom of each cell that is not empty, Tinderbox shows the display attribute of one note in the cell. The display attribute is typically $DisplayName, but may be any attribute.

Hover the mouse over a crosstabs cell to browse through other notes in that cell. The first note in outline order is viewed by placing the mouse at the left edge of the cell and the last note in outline order is viewed by placing the mouse at the right edge of the cell. Click to select the current note and to see its text and displayed attributes. You can also use the ↑ and ↓ keys to select the next and previous notes in the cross tabs cell.

In the center of each crosstabs cell is a bar that shows the relative location of each note in the cell in the document's outline. The left edge of the bar corresponds to the first descendant of the chosen container, and the right edge corresponds to the last descendant of the chosen container. A vertical line representing each note appears in the note's outline color.

It is sometimes useful to make an agent that collects the items corresponding to interesting cells of the table. Right-click or control-click the interesting cell and choose Copy Agent For This Cell from the contextual menu. Tinderbox will place an agent that gathers notes in this cell onto the clipboard; choose a map or outline view in another tab and paste the agent where you like.

You can apply an action or a stamp to all the notes in a cell. Right-click or control-click the crosstabs cell, and choose Perform Action. Enter the action you want to perform, or the name of the stamp you want to apply, and then press [Return] to perform the action.

Find

In the view pane, choose **Edit ▸ Find** (⌘-F) to locate notes that contain a particular word or phrase. A bar will appear at the top of the view pane, allowing you to search the notes' names, text, or both.

Find searches for regular expressions which can describe a range of textual patterns.

Press return after entering your search pattern to view a list of results. Results that are visible in the current view are listed first; select them to select the corresponding note. Results that are not currently visible in this view are listed separately; double-click to modify the view to make them visible. Shift-double-click will open the note in a new tab.

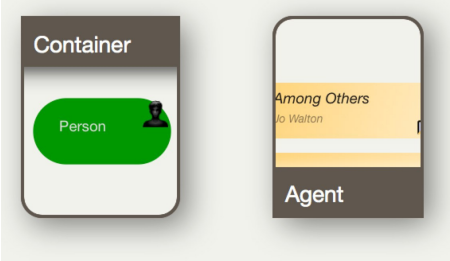
While entering search terms, press **F5** for suggestions of related words.

Drag any item from the list of found notes into the view pane to make an alias to that note.

Agents

Tinderbox agents scan your document constantly, looking for notes that meet criteria you have specified. When an agent finds a note that meets its criteria, it creates an alias of that note inside the agent.

Agents are drawn in maps with a characteristic shape. The title bar of an agent appears at the bottom of the note, while the title bar of an container appears at the top.



To create an agent:

- Right-click in the background of a map and select “Create Agent”, OR
- Select a note in the current view and select “Create Agent” from the Note menu. The agent will be created near the selected note.

The agent’s behavior is determined by a number of special attributes used by agents. The most significant of these are:

AgentQuery	the criteria for which the agent is looking. If a note satisfies the query, the agent will contain an alias of that note. If a note no longer satisfies that query, the agent will remove the note’s alias from its children.
AgentAction	an optional action to be performed on the newly-created alias of any note that matches the agent’s query.
AgentPriority	The agent’s relative priority, controlling how frequently the agent updates itself.
CleanupAction	The method used by the agent to arrange its children. The default value, `grid`, arranges the children in a rectangular grid.

Agent queries and actions are written with simple but powerful Tinderbox commands. For details, see Queries and Actions.

To inspect or change an agent’s query, action, or other properties:

- Select the agent, choose Get Info (⌘⌥-I) from the Note menu, and select the **agents** pane, OR
- Open the Action inspector and choose the Query, Action, or Sort pane as required.

Ziplinks

Ziplinks facilitate rapid linking to old and new notes while brainstorming and writing *zettelkasten* or speedy notecards. In older versions of Tinderbox, they were called “quicklinks”.

To make a link to a named note, precede the name of the note with two brackets and follow its name with a two brackets

    [[like this]]

If there is no note with this name in the document, Tinderbox will create one as the sibling of the note you are editing.

If you wish to make a ziplink to a note in a different container, you may enter the path to the destination in the ziplink:

    [[/container/note]]

The source text of the new link anchor is typically the name of the destination note, but you can choose a different anchor text to be replace the [[bracket]]:

[[note|explanation]]

You may also set the text of the quicklink note:

    [[note|explanation::see this]]

If the destination already exists, the new text is appended to the note.

As you are typing a ziplink, Tinderbox will show a list of related note names to the left of the text pane. Click on any of these notes to make a link, or option-click to copy the destination to the ziplink while leaving the ziplink open for further editing. Alternatively, press the up-arrow or down-arrow keys to select the note you prefer, then press [Return] to make the link or press ⌥[Return] to copy the link destination to the ziplink while leaving the ziplink open for further editing.

The ziplink popup menu displays possible ziplink destinations as you type.

[[shows a list of all siblings of this note. Containers are bolded.

[[/ shows a list of all top-level containers

[[/D shows a list of all top-level notes that begin with a “d”

[[/Dog/ shows a list of all notes in /Dog

[[/Dog/R shows a list of all notes in /Dog that begin with “r”

[[D shows a list of all notes with names that contain a “d”

The tooltip for each item in the ziplink popup menu displays that item’s full path.

Cick on any item in the ziplink popup menu to make that link. Option-click (⌥-click) on any item to autocomplete that item while leaving the ziplink open for additional editing.

The Ziplinks popup menu now takes into account your locale, so *étude* will sort appropriately for French users and Ångstrom will sort correctly in Norway.

To disable ziplinks in a note, set $Ziplinks to false.

Ziplinks are always created using the link type ***untitled**.

Markdown

Some writers prefer to use **styled text**, others prefer **Markdown**. Tinderbox works the way you prefer.

To use Markdown for a note, use the built-in prototype Markdown. This sets $HTMLMarkdown to **true,** informing Tinderbox that you plan to use Markdown in this note.

If $HTMLPreviewCommand is **CommonMark**, Tinderbox previews the text using the CommonMark Markdown standard (cheat sheet and documentation). CommonMark is significantly faster for large documents than the original Markdown script. CommonMark is now adopted by the built-in Markdown prototype in new documents, though the older Markdown remains available. Set $HTMLPreviewCommand to **Markdown** to use the classic Markdown processor. To use your own Markdown processor, enter its command path in $HTMLPreviewCommand.

To switch from editing Markdown to previewing the formatted text, switch to the Preview pane by:

- Selecting Preview from the Text Pane Selector at the top of the text pane, or
- Choosing **Preview In Text Pane (⌘⌥-E)** from the Windows menu.

The appearance of formatted previews may be customized through a note in the built-in hints, /Hints/Preview/style. This note contains styles that are applied by the built-in Markdown template. This template can itself be modified by editing /Hints/Preview/template.

Alternatively, Tinderbox automatically communicates with the *streaming preview* of the application Marked2 (https://marked2app.com). When you select or edit a Markdown note in Tinderbox, it will automatically be sent to the Marked2 streaming preview window.

You can drag Markdown files into Tinderbodx views to create notes. Tinderbox formats the imported notes using your chosen font. If you hold down the Option key ⌥ while dragging a Markdown file into Tinderbox, the file will be imported as plain text.

Holding down the option key:

1. BIG NEWS!

   This is *important*.

Not holding down the option key:

BIG NEWS!

This is *important*.

Composites

A **Composite** is a group of Tinderbox notes that work together to describe something larger than themselves. For example, when taking notes in a conference, an individual talk might be a composite of notes for the title, the speaker, the content of the talk, and action items you want to follow up on. Each note has its own text and attributes, but you may sometimes want to treat the composite as an object – for example, you might want to move all the notes in the composite to a new map location.

Tinderbox notes form a composite when they touch other notes.

Clicking on a note that is part of a composite selects the entire composite. You select an individual note by command-clicking it. You can remove a note from a composite by command-clicking to select the note and then dragging it away from the other notes.

When a composite is selected, the text pane displays the text from each member of the composite. The texts appear in outline order. The text is not (currently) editable; to edit, select an individual note.

Composites may have a name. When any item in a composite is selected, the name of the composite is shown above the composite, along with a control widget that allows you to rename the composite.

If a note should not become part of a composite, you can set its attribute **$NeverComposite** to **true.** A note's proptotype, for example, might prevent notes of a particular type frim joining composites. To disable composites throughout a document, set the default value of **$NeverComposite** to **true.**

Composite Actions

Notes may have an $OnJoin action. When a note that was not previously a member of a composite is dragged to touch a note with an $OnJoin action, the $OnJoin action is performed on the dragged note. If a note touches more than one note in the composite, each $OnJoin action is performed in turn. In the $OnJoin action, *this* is bound to the note joining the composite and *that* is bound to the note to which is is now adjacent.

Built-in Composites

The File ▸ Built-In Composites submenu allows you to create composites that are predefined and that may be useful in various settings. **You may adapt these composites as you like, and may also add your own composites to the /Composites container**.

Composites in the /Composites container can be instantiated by the right-clicking in the map view and choosing the composite you want from the **Create Composite** submenu, or by selecting the composite from **Note ▸ Create Composite.**

Items in a composite may have a $Role, which indicates what part they play in a composite. For example, a composite that describes a research paper might include notes with roles like Author, Abstract, and References. If a note has a $Role but not $Subtitle, the Role is displayed where the Subtitle would normally appear.

Some kinds of composites might have multiple notes with the same role. For example, a list with a header would have one note with the role Header, but might have any number of notes with the role ListItem. The boolean attribute $IsMultiple indicates that a role may have multiple notes. If $IsMultiple is true, when a note is moved into the composite touching that note, the moved note inherits the note's role. Thus, an item moved into a composite and touching a ListItem will take on the role of a ListItem.

Composites and Actions

Actions and queries can now use information about composites. All are read-only unless specified.

> **compositeFor(theNode)** returns a list of paths of notes in a composite.
>
> **compositeWithName("theName")** returns the a list of paths of notes in a composite with a given name. If several composites have the same name, one of those composites is returned.

The path list returned by **compositeFor()** or **compositeWithName()** may be used as a designator, or to obtain additional information about the composite.

> **compositeFor(theNode):count** returns the number of notes in a composite.
>
> **compositeFor(theNode):name** returns the composite name. This expression is read.write: compositeFor(this):name="example" renames the composite.
>
> **compositeFor(theNode):role(theRole)** returns a list of paths of notes in a composite, including only the note or notes with the designated role. For example, to set the color of all notes with the role "author": $Color(compositeFor(great books):role(author))="red"
>
> **compositeFor(theNode):roles** returns a set of roles that appear in the composite.
>
> **compositeFor(theNode):kind** returns the name of the composite from which this composite was instantiated. For example, if a composite was created from the built-in list composite, this function will return "list". The query **compositeFor(this):kind=="book"** would locate all notes that participate in composites instantiated from "book" – all books.

The expression **my:** is equivalent to **compositeFor(this)**. For example,

> **my:role(author)**

returns the notes with the "author" role in this note's composite.

Command and Info

The command bar gives you keyboard access to many Tinderbox commands and lots of Tinderbox information.

To open the command bar, select **Help ▸ Commands & Info…** (⌘⇧-U)

Type any word or phrase to see available topics. Up-arrow and down-arrow move along the list. Press [Return] to perform the selected action.

Home and End move to the first and last items, respectively. Esc dismisses the command bar, as does clicking outside the command bar.

Open ….

Lets you open any Tinderbox document you've recently used.

Select …

Lets you select any note in the current document; if possible, Tinderbox will scroll to make the note visible.

Explain …

Provides instant access to some key places in aTbRef

$Attribute …

Defines system attributes. If one note is selected, also shows the value of this attribute for the selected note.

View …

Notable Tinderbox videos

Inspector…

Open the inspector and switch to a named pane

Taggers

Taggers

Taggers help Tinderbox's neural networks to identify topics of special interest to you.

A tagger is defined by making a new note inside /Hints/Taggers. The name of each Tagger note corresponds to a set attribute. Built-in taggers are provided for $NLPlaces, $NLNames, $NLOrganizations, and $NLTags. You can add your own taggers for user set attributes.

Each line of a tagger describes one possible tag.

```
Tag:word 1;word 2;word 3
```

For example, if NLTags contains the line

```
Eastgate:Eastgate;Tinderbox;Storyspace;Mark Bernstein
```

then $NLTags will include the tag "Eastgate" if the text of the note mentions "Tinderbox" or "Storyspace" or "Mark Bernstein". The shorthand

```
Eastgate
```

may be used if only the tag itself is to be recognized; it is equivalent to:

```
Eastgate:Eastgate
```

Taggers are particularly useful for agents. Often, an agent needs to look for several significant or interesting terms in the text:

```
query: $Text.icontains("Caesar") |$Text.icontains("Cato") ...
```

In projects that have lots of text, these agents have a difficult task: each clause might need to search all the text in the document, and each clause needs to build its own regular expression matcher. The tagger is more restricted — it searches only for entire words, not for regular expressions — but it is significantly faster and it runs only when the text of the notes changes. In the example above, we could replace the query with the simpler query

```
query: $NLTags.contains("Rome")
```

and add a single line to /Hints/Taggers/NLTags

```
Rome:Caesar;Cato;Cicero;Aggripa;Brutus;Decius;Catullus
```

Now, the agent needs only to examine set $NLTags, which requires far less work than searching all the text of a large document.

Taggers examine both the name and the text of each note.

Taggers are available in seven major languages.

Sentiment Analysis

Sentiment Analysis

Tinderbox taggers also attempt to analyze the general tenor or sentiment of each note. Sentiment is measured on a scale from +1 to -1. An enthusiastic note like

> “This cheese is tasty, wholesome, and creamy. It is a delight!”

is scored near 1, and a critical note like

> “This cheese is rotten, slimy, and stale. It tastes terrible and should be thrown away.”

is scored near -1.

The average sentiment for the entire note is stored in `$Sentiment`, and the score for each paragraph of the note is stored in `$Sentiments`.

Sentiment analysis is available on macOS 10.15 and later, in seven major languages.

Highlighters

Highlighters

Highlighters let Tinderbox' highlight words and phrases it finds in the text of your notes.

A highlighter is defined by making a new note inside the container `/Hints/Highlighters`. This container holds notes that describe ways to highlight the text of a note.

Any note can opt in to highlighting by setting the **$SyntaxHighlighting** value; for example, setting $SyntaxHighlighting to **markdown** would tell Tinderbox to use the highlight methods found in /Hints/Highlighters/markdown.

The highlighting methods are a list of regular expressions and formats. For example:

```
pattern:parent|grandparent|nextSibling|previousSibling

    color: red
```

would locate any of these words and color them red. The first line of a rule must not be indented; additional lines for each rule must be indented. Empty lines may be added between rules for clarity.

Comments are preceded by // and continue for the rest of the line:

```
pattern:parent

    color: red // perhaps dark red?
```

In addition to **color:**, syntax highlighting can set:

- the **background:** color
- the **size:** in points
- the **line-spacing** as a multiple of conventional line spacing for the paragraph containing the pattern
- the **indent** in points for the paragraph containing the pattern
- the **first-indent** applied to the first line of the paragraph containing the pattern.

The command

```
    bold: yes
```

makes the text bold, and the command

```
    italic: yes
```

makes the text italic.

```
    strike: yes
```

strikes through the text.

```
    underline: yes
```

underlines the text.

```
    wildcards: no
```

instructs the pattern to ignore regular expression metacharacters such as "*" and "+".

```
    case-sensitive: no
```

instructs the pattern to regard upper- and lower-case letters as equivalent.

Tinderbox Fonts

Tinderbox includes a number of high-quality fonts that work beautifully for notes. These fonts can be used for any purpose withing Tinderbox, and include:

Sans-Serif

- Ideal (elegant everywhere)
- Decimal (ultramodern)
- Ringside (compact)
- Tungsten (extremely compact headline font)

Serif

- Mercury (elegant text font)
- Archer (super-legible and very modern)

Handwriting

- Sketchnote (a *good* handwritten font. Don’t use Comic Sans!)

Queries and Actions

Actions, Expressions, and Rules

In many places, Tinderbox allows you to change what information Tinderbox displays or what actions Tinderbox performs on notes that you have created or moved. These symbolic actions and expressions can provide great flexibility and can save you lots of time. For example:

- A container that holds a “reading list” can know that notes added to the list are likely to be books or articles, and can automatically set the size, appearance, and prototype of new notes.
- A container that holds a set of “chapters” for your book can display the current total word-count for all notes inside it.
- An agent that contains “urgent tasks” can automatically change their color in order to draw your attention.

An *expression* is simply something that has a value. For example, the expression `0.05*$Price` might calculate the sales tax on an item based on its price.

An *action* describes a change made on one or more notes. For example, the action `$Color(California)="red"` sets the color of the note named “California” to “red”.

Actions may be performed at a variety of times.

- An **OnAdd action** is performed when a note is added to a container, is discovered by an agent, or is placed atop an adornment, and affects the note being added.
- A **Rule** is performed at frequent intervals, and affects the note that possesses the rule.
- An **Edict** is performed after a document is opened, and at infrequent intervals while the document remains open.

Comments

Comments may be included in actions to explain implementation details and clarify your intention. Avoid excessive comments; it is better to make code clear than to add comments.

Comments begin with the characters //. Comments end with the next occurrence of the characters //, at the next newline characters, or at the end of the action — whichever appears first.

```
$Width = 3 + // 10 + // 2; ➛ 5

$Width =5; // was $Width=7.5 ; ➛ 5

// $Width = 5; ➛ no action
```

Expressions

A Tinderbox expression represents a value — something that could potentially be stored in an attribute. The following are expressions:

- 42
- "Now is the winter of our discontent"
- (7+6.5)/5
- $Price

Note that strings are enclosed in straight quotes – you may use single or double quotation marks as you prefer, but the closing mark must match the mark that opens the string. Number may include decimals. When you want the value of an attribute, place a “$” sign before the attribute name; thus $Price represents the current value of the user attribute $Price for *this* note.

Most actions simply store the result of an expression in an attribute. For example

```
$Price=1.95
```

sets the value of the attribute `Price` in the current note, and

```
$Color="red"
```

turns the current note red.

In addition to their use in actions, expressions have a variety of uses in Tinderbox:

- `$AgentQuery` is an expression; if the `$AgentQuery` is `true` for a given note, then that note will be listed by the agent.
- `$DisplayExpression` is an expression; if `$DisplayExpression` is not empty, then the value of `$DisplayName`, ands the title displayed for the note in maps and other views, is the result of evaluating `$DisplayExpression`.
- The value of `$HoverExpression`, if `$HoverExpression` is not empty, appears while the cursor hovers over a note in map or or outline view.
- The value of `$TableExpression` determines what appears in a note’s summary table.

True and False

When evaluating an agent query or assigning a value to a Boolean attribute, the following values are all treated as equivalent to `false`.

- the keyword `false`
- the number 0
- the empty string ""
- the color #000000
- the date "never"

Any other value is treated as `true`.

Actions

Assigning Values

Most Tinderbox actions simply assign a value to an attribute.

```
$Color="red"
```

Several actions may be combined, separated by a semicolon.

```
$Color="red";$BorderColor="white";
```

To remove the value from the current note, restoring the inherited or default value, simply omit the value the follows the “=” sign.

```
$Color=;
```

The conditional assignment operator |= will assign a value to an attribute only if that attribute’s value is currently false, zero, or empty.

```
$Priority |= "low"
```

The conditional assignment operator &= will assign a value to an attribute only if that attribute’s value is not false, zero, or empty.

+= and -=

These assignment operators are a convenient shorthand for incrementing or decrementing an attribute. For example, the two following statements are equivalent:

```
$MyNumber += 3;
```

```
$MyNumber = $MyNumber + 3;
```

Similarly:

```
$MyNumber -= 3;
```

```
$MyNumber = $MyNumber - 3;
```

These operators may also be used with lists and sets, and += may be used to append strings.

```
$MyString += "!"
```

Note that += and -= are not currently available for dictionary references of the form

```
$MyDictionary[index]+=1;
```

Instead, use the conventional form:

```
$MyDictionary[index]=$MyDictionary[index]+1;
```

Conditional Actions

A conditional action is performed only if a specified condition is satisfied.

```
if($DueDate<date("tomorrow")) {$Color="red";}
```

Conditional actions may also specify what should be done if the condition is not satisfied.

```
if($DueDate<date("tomorrow")) {$Color="red";} else {$Color="green"}
```

while(test){...} performs an action repeatedly until the expression test is false. For example:

```
while($Text) {var:string s=$Text.sentence; $Text=$Text.substr(s.size).trim(); process(s);}
```

calls the function **process()** on each sentence of the text in turn, until the text is empty. In order to reduce the incidence of infinite loops, while loops are limited to 10,000 iterations.

Side Effects

If an action is simply an expression, the expression is evaluated and the result discarded. For example, the action

```
runCommand("open /Applications/iTunes.app")
```

will ask your computer to open iTunes, and the action

```
notify("find concert tickets")
```

will post a notice to your computer’s notification center.

Repeated Actions

A repeated action is performed for every element in a list or set. For example:

```
$Result=0; $MyList.each(x){$Result=Result+x;}
```

would take a list of numbers, add them up, and store the sum in $Result.

Repeated actions are uncommon.

Edicts

An **Edict** is an action, much like a rule, that is performed when a document is opened and at infrequent intervals while the document remains open. Edicts are useful for housekeeping chores which are unlikely to be urgent.

For example, suppose your Tinderbox document contains, among other things, a few thousand Tasks of various sorts. You'd like each task to adjust its appearance depending on a variety of factors: whether the task is important, how soon the task is due, whether the task has already been completed.

You *could* use a rule, inherited from the Task prototype, to perform these chores. Each morning, you'd open the document, Tinderbox would review each task in turn, and then adjust the appearance of each task, updating any tasks which had become overdue since the most recent check. Once done, though, Tinderbox's rule manager would then check each Task again, just in case a task had changed status since the previous check. This does no particular harm, but it does use a little extra processing power and consume some battery charge, while the benefit of checking whether a task has suddenly become overdue in the previous minute is slight.

Instead, we can use an *Edict* to adjust the appearance. Edicts run infrequently, and so they consume less processing power and battery charge. (At present, edicts run at startup and then at intervals of approximately one hour, though these details are subject to change.)

Attributes and Constants

Attribute References

To refer to an attribute by name, precede the name with "$". For example,

```
$Width
```

refers to the width of this note, and the action

```
$Width=7;
```

will set its width to 7.

To refer to the value of an attribute of a different note, follow the attribute name with a designator in parentheses. For example,

```
$Width(parent)
```

refers to the width of the parent of this note,

```
$Width(Burbank)
```

refers to the width of the note named "Burbank", and

```
$Width(/people/Roosevelt)
```

refers to the width of the note named "Roosevelt" that is inside the container named "people".

Tinderbox provides a rich list of designators for use with attribute references.

Constants

Enclose string values in either (straight) single or double quotes:

```
$Name="Theodore Roosevelt"
```

Double quotes are preferred; use single quotes when the quoted string contains double quotes.

```
$Name='She said, "Hello," before I had time to think.'
```

Date constants are best expressed using the date operator:

```
date("today")
```

```
date(12,25,1917)
```

Color constants can be expressed using string equivalents, such as "red", "#FF9900", or "RGB(255,255,255)".

The boolean constants `true` and `false` are not quoted.

false	the boolean value `false`
"false"	a five-letter word

Designators

Designators specify to which note an attribute reference refers. For example, in the expression

```
$Width(parent)
```

the designator "parent" refers to the note that contains this note. If the designator is omitted, the reference refers to `this` note.

- In rules, `this` refers to the note whose rule is running.
- In agent queries, `this` refers to the note being examined by the agent.
- In agent actions, `this` refers to the newly-created alias that satisfies the agent's query.
- In OnAdd actions, `this` refers to the note that is being added.

adornment, or the note whose rule is being run

`this`	The note being examined by an agent, the note being added to an adornment, or the note whose rule is being run

next	The note that follows this note in outline order
previous	The note that precedes this note in outline order
prevSibling	The next older sibling of this note
nextSibling	The next younger sibling of this note
nextItem	The note that follows this note in outline order. See below for the difference between nextItem and next.
previousItem	The note that precedes this note in outline order. See below for the difference between previousItem and previous.
prevSiblingItem	The next older sibling of this note. See below for the difference bwteen previousSibling and prevSiblingItem.
nextSiblingItem	The next younger sibling of this note. See below for the difference between nextSibling and nextSiblingItem.
firstSibling	The first sibling of this note
lastSibling	The last sibling of this note
parent	The parent of this note
grandparent	The parent of the parent of this note
child	The first child of this note
child[n]	The nth child of this note, where n is a number. The first child is child[0]. If n is negative, the nth youngest child of this note: child[-1] is the youngest child.
lastChild	The last child of this note
randomChild	A randomly selected child of this note
current	The note Tinderbox is currently exporting. When not used in an export template, current is equivalent to this. Where a note is exported to its own page, current is the same as this. If the note is included in another note's page, however, current refers to the page being exported.
cover	The first note in the document.
source	The link source (used only in link expressions)
destination	The link destination (used only in link expressions)
agent	Available only in agent queries and agent actions; refers to the agent that is currently examining the note.
adornment	Available only in adornment queries and adornment actions; refers to the smart adornment that is currently examining or performing an action on the note.
original	In aliases, refers to the original note associated with the alias and useful for addressing properties of the original such as `$Xpos` and `$Ypos` that are not inherent to the alias. Otherwise synonymous with this.
that	Within a find() expression, this is bound to the note being tested by find(),and that is bound to the note otherwise bound to this.

A note's $ID value may be used as a designator. For example, $Name(1524673590) locates the $Name of the note whose ID is 1524673590. This usage should be avoided whenever possible, as it is hard to read and prone to confusion, but may be useful in some cases where unique paths are impractical.

Designators may be combined. For example, `$Name(nextSibling(parent))` is the name of the next sibling of this note's container.

next and nextItem

For convenience in export, several designators skip notes that are not exported, or usually considered outside the document outline. These include:

- adornments
- separators
- other notes which do not export

These include **next, previous, nextSibling, previousSibling.**

To designate a note that might be an adornment, separator, or unexported item, use the corresponding *item* designators: **nextItem, previousItem, nextSiblingItem, previousSiblingItem.**

Group Designators

Some operators accept a group designator that designates multiple notes. For example, `any(child,$Checked)` is true if the $Checked attribute of any of this note's children is set to true. Group designators include:

children	the immediate children of a note
descendants	all descendants of a note
siblings	all siblings of a note
ancestors	all ancestors of a note
adornments	all adornments that are children of this note
all	all notes in the document

A list or set of pathnames may be used anywhere a group designator is required.

Group designators can be used to assign a value to many notes at one time.

```
$Color(children)="red";
```

A group designator can also construct a list of the values of an attribute used by a list or set of notes:

```
$MyList=$Color(children);
```

find()

The special designator find() searches through the entire Tinderbox document to locate notes that satisfy an expression. For example:

```
$Color(find($Status="Urgent"))="red";
```

will locate every note whose $Status is "Urgent" and turn is red.

The find() designator acts in many ways like an agent. In general, prefer agents to using find(), because the agent's results can be reused by other agents.

Text Patterns

The operator .contains() tests whether a text pattern or regular expression appears in a note's attribute. For example,

```
$Text.contains("frog")
```

is true if the text of the note contains the string "frog". The .contains() operator is case-sensitive, but

```
$Text.icontains("frog")
```

will match both "frog" and "Frog".

In older documents, you may encounter the obsolete syntax Text(pattern). This is equivalent to $Text.contains(pattern) — note that the '$' is omitted in the older syntax. This deprecated syntax continues to function but is harder to read than the modern .contains().

Regular Expressions

A number of special characters represent "wild cards" and other classes of text patterns. Complete information on Tinderbox's regular expression engine may be found at:

http://www.boost.org/libs/regex/doc/

Wildcard Characters

The period character, ".", matches any single character.

The plus sign, "+", matches one or more occurrences of the expression that precedes it. The pattern

```
!+
```

will match one or more exclamation points, and the pattern

```
....+
```

will match any string with at least four characters.

An asterisk matches zero or more occurrences of whatever precedes it;

```
10*
```

matches 1, 10, or 1000.

The question mark "?" matches zero or one occurrence of whatever precedes it.

You can also specify the minimum and maximum number of repetitions:

```
Xa{2,4}Y
```

will match XaaY, XaaaY, or XaaaaY, but won't match XaaaaaaY.

Character Ranges

A set of characters to be matched may be enclosed in square brackets. For example,

```
[0123456789]
```

will match any digit. Ranges of consecutive characters can be written more concisely:

```
[0-9]
```

will match any digit, and

```
[A-Z][a-z]*
```

will match any capitalized word. Beginning a set with the character "^" matches everything except the set;

```
[A-Z][^0-9]
```

will match any capital letter provided it's not followed by a digit.

Several special sequences represent common sets of characters:

\w - any word character (including underscore)

\W - any non-word character

\< - the start of a word

\> - the end of a word

\s - any white space character

\d - any digit

\l - any lowercase letter

\u - any uppercase letter

Anchors

The special character "^" matches the beginning of the text or attribute being searched. When searching the text of a note, ^ matches the beginning of any paragraph in the note.

The special character “$” matches the end of the text or attribute being searched. When searching the text of a note, $ matches the end of the paragraph in the note.

The \ [backslash] Character

The backslash character "\" removes the special meaning from the character that follows it. Use "\\" to search for the backslash character itself.

Parentheses

Grouping expressions in parenthesis determines the scope of wildcards. For example,

```
Name=(\u\l+)+
```

Would match “Rochester” and “SmallTalk”.

In addition, when Tinderbox sees a parenthetical expression, it remembers the substring(s) that matched it and can use those substrings in actions. For example, the agent

Query: `$Text.contains("^Color: (\w+)\b$")`

Action: `$Color=$1`

scans the document for any notes that contain paragraphs like this:

```
Color: red
```

If it finds any matching notes, the agent extracts the word that follows the string “Color: ” and changes the note’s color to match. Here, $1 stands for “whatever matched the first set of parenthesis”, $2 for the second set, and so forth. $0 stands for the entire matched string. The expression %matches represents a list that contains all available regular expression matches. This is equivalent to the list [$0;$1;$2….$N], where N is the count of available regular expression matches.

Back References

When agents search for regular expressions, the agent saves substrings that match parenthesized sub-expressions. These substrings or back references can be used in actions, and are especially useful for automatically extracting information from notes.

For example, an agent might search for notes that contain fields like this:

From: Henry Higgins

by searching for the pattern

```
From: (.+)$
```

An action can then refer to the name extracted following "From: " as $1

```
$Author=$1;
```

The author will now be set to Henry Higgins. Subsequent subexpressions may be referred to as $2, $3, and so on.

Local Variables

In addition to the note’s values, you can declare temporary variables for use only within the current action using the var statement:

```
var area; area=$Width*$Height; $MyNumber=area;
```

Multiple local variables may be declared in one statement, separated by commas.

```
var x,y; …
```

A local variable may be assigned an initial value as follows:

```
var taxRate(0.0625); $Total=$Price*(1+taxRate);
```

A local variable may equivalently be initialized by assigning a value:

```
var taxRate = 0.0625;
```

Vars may have optional types.

```
var:number x(5);
```

Available types include “number”, “set”, “list”, “date”, “color”, “date”, “interval”, and “dictionary”. Providing an explicit type helps Tinderbox provide the answer you want. For example:

```
var x:number(5); x=x+5; $MyString=x; ➤ 10
var x:list(5); x=x+5; $MyString=x; ➤ 5;5
var x:string(5); x=x+5; $MyString=x; ➤ 55
```

Functions

Tinderbox allows you to define your own operators.

Operators are typically defined in notes inside the **Library** container in your Built-In Hints container. Notes in this container which have names enclosed in parentheses, such as

```
/Hints/Library/(What's This)
```

are treated as documentation; all other notes are executed at document startup and after they are edited.

Defining functions

Functions are defined by the **function** statement:

```
function fname(args){action}
```

For example, a function might return a number 18% larger than its argument.

```
function addTax(price){return(1.18*price);}
```

This defines a new function that we can use in any action or expression.

```
$MyNumber = addTax(500);
```

This would set MyNumber to 1.18*500 = 590.

A function may have no arguments:

```
function reset(){$Color=;$BorderColor=;}
```

If a function has arguments, the caller must supply values for each argument. Extra arguments are ignored.

Function declarations may optionally specify the type of their arguments. For example,

`function append(x:list) { return x+1;}`

declares that values passed to the function **append()** will be treated as lists.

If a function is redefined, the most recent definition replaces any existing definitions.

Functions should not use names reserved for built-in operators. Names are case-sensitive. Function names should begin with a letter, and may contain letters, digits, or the underscore character. Functions should not share names with attributes.

Returning values

The return statement returns a value from a function. For example:

```
function answer(){return 42;}
```

Always returns 42.

```
function fib(n){
 if (n<2) {return (n);}
 return fib(n-1)+fib(n-2) ;
}
```

…returns the Nth Fibonacci number.

The return value may be enclosed in parentheses and should be terminated by a semicolon.

String Operators

+ (concatenation)

The + operator joins two strings.

"We "+"few"	"We few"

* (repetition)

The * operator repeats a string multiple times.

"!" * 5	"!!!!!"

.at("keyword")

Extracts a value from a list of keyword-value pairs in which the keyword and value are separated by colons. For example, suppose

`$MyString = "USA: North America; Canada: North America; France: Europe; Japan: Asia; default: unknown"`

Then

`$MyString.at("France") ==> "Europe" $MyString.at("Canada") => "North America"`

The special keyword "default" applies to any keyword not contained in the list. For convenience, when several keywords have the same value they may

be separated by pipe characters “|”:

"USA | Canada: North America; ..."

The **at("keyword")** operator or **Lookup Table** can often replace complex conditional logic. For example, if $Continents(/config) contains the string

"USA: North America; Canada: North America; France: Europe; Japan: Asia; default: unknown"

The we can simply write:

$MyContinent = $Continents(/config).at($Country)

rather than the long and error-prone

if ($Country=="USA"){$MyContinent="North America"} else {if ($Country=="Canada") .....

See **.lookup()** for details on lookup tables.

.capitalize

The .capitalize() operator transforms a string by capitalizing its initial letter.

"ostrich".capitalize()	"Ostrich"

See .lowercase() and .uppercase().

.beginsWith

The .beginsWith operator returns true if a string begins with its argument.

"ostrich".beginsWith("o")	true

If you wish to search for a regular expression patterns, see .contains().

See also .endsWith().

.bold

Adds the bold style attribute to the string. This has no effect unless the string is placed in $Text, perhaps via **.replace()**.

.contains

[String]contains("pattern") is true if the string contains the quote-enclosed regular expression "pattern", which may be a regular expression or a string literal. Matches are case-sensitive. The pattern should always be enclosed in quotes. If $Text contains the string “Asia”, then:

$Text.contains("Asia")	true, since the text contains "Asia"
$Text.contains("s.a")	true, since the text contains an "s" which is followed by a character followed by an "a".

If a match is found, the actual matching text is bound to $0.

If the regular expression contains parenthesized sub-expressions, the entire matches string is bound to $0 and the matching sub-expressions are bound in order to $1…$99. The expression %matches represents a list that contains all available regular expression matches. This is equivalent to the list [$0;$1;$2….$N], where N is the count of available regular expression matches.

.containsAnyOf

[String].containsAnyOf(*set*) - true if the string contains at least one pattern from a semi-colon-delimited set, which may contain regular expressions or a literals. Matches are case-sensitive. The most common use of this expression is:

```
[String].containsAnyOf(wordsRelatedTo("word"))
```

.countOccurrencesOf

[String].countOccurrencesOf("pattern") - return the number of times the string “pattern” occurs in the string. Note that pattern is a string, not a regular expression.

For example:

```
$MyString,countOccurrencesOf("*")
```

… returns the number of asterisks in $MyString.

.deleteCharacters

[String].deleteCharacters(characterSet)

Returns a copy of the string from which any characters in the character set have been removed. For example

```
[String].deleteCharacters("1234567890")
```

…would remove any digits from the string.

.empty

String.empty is true if the string is has no characters.

"".empty	true
"element".empty	false

.endsWith

The .endsWith operator returns true if a string ends with its argument.

"ostrich".endsWith("rich")	true
"ostrich".endsWith("RICH")	false

If you wish to search for a regular expression patterns, see .contains().

See also .beginsWith().

.extract

[String].extract(*pattern*) searches the string for a regular expression, and returns the sub-expression of the first match. If the pattern has no subexpressions, the entire match is returned. For example,

```
$Text.extract(#[A-Z][a-z]+)/
```

would extract "#Tinderbox" from the following text

```
This was exported by #Tinderbox on 25 March 2023.
```

.extractAll

[String].extract(*pattern*) searches the string for a regular expressiona list of all matches. For example,

```
$Text.extract(#[A-Z][a-z]+)
```

would extract "#Tinderbox;#GoodNews" from the following text

```
This was exported by #Tinderbox on 25 March 2023. #GoodNews!
```

.find

The .find operator returns the position of the first occurrence of a string. If the string is not found, .find returns -1.

"ostrich".find("rich")	3
"ostrich".find("RICH")	-1

.textColor(color)

Adds a text foreground color style attribute to the string. This has no effect unless the string is placed in $Text.

For example

```
$Text="example".textColor("#FF0000")
```

sets the the note's text to a red word.

The color may be a symbolic color like "red" or "warm gray", or a hex color like "$ff6633".

See also .bold and .plain.

.following

The .following(*pattern*) operator extracts information from one or more paragraphs of a string. If *pattern* appears in a line, then whatever follows the pattern is the result. If pattern appears in more than one line, the return value is a list of results.

For example, if the text of a note is

```
From: Mark

To: Clotilde

Subject: Mignardise
```

Then $Text.following("To:") would return "Clotilde". $Text.following(":") would return "Mark;Clotilde:Mignardise".

In addition, .following binds any text that precedes the matched pattern to $0. If the pattern appears several times, the first prefix is bound to $0, the second to $1, and so forth. In the example above, the expression .following(":") would bind "From" to $0, "To: to $1, and "Subject" to "Mignardise".

Note that the *pattern* argument is interpreted as a sequence of characters, not at a regular expression. If you need the flexibility of regular expressions, use .contains()

The .following() operator can make it very easy to extract information from common forms and simple internet formats.

.highlights

The expression **$Text.highlights** returns a list of strings that have been highlighted in the text using Format ▸ Highlight. The result may be limited to strings highlighted in a specific color by requesting

```
$Text.highlights(color)
```

where *color* may be red, green, blue, yellow, or magenta.

.icontains

The case-insenstive variant of .contains(). If the note's text contains the word "Asia", then

$Text.icontains("Asia")	true, since the text contains "Asia"
$Text.icontains("ASIA")	false

.icontainsAnyOf

String.icontainsAnyOf(*set*) - true if the string contains at least one pattern from a semi-colon-delimited set, which may contain regular expressions or a literals. Matches are case-insensitive. The most common use of this expression is:

```
[String].icontainsAnyOf(wordsRelatedTo("word"))
```

.italic

Adds the italic style attribute to the string. This has no effect unless the string is placed in $Text, perhaps via **.replace()**.

.jsonEncode

Returns the string in JSON-encoded form. Characters that have special meaning in JSON are escaped with backslashes.

.lookup

Treats the string value as a lookup table and returns the value corresponding to the argument. For example:

`"Cat:0; Dog: 1; Chicken 3: default: 42".lookup("Dog")`

would return “1”.

A lookup table is a list of values and results. The value and result are separated by a colon:

```
Cat:0
```

For convenience, several values may share a common result; separate them by a pipe (|) character

```
Cat | Lynx | Lion : 0
```

A range of values may replace the value itself:

```
Cat-Dog: 0
```

Here, all values greater than or equal to “Cat” but less than “Dog” will return the result “0”.

.lookup is a synonym for **.at**, but always uses a lookup table where **.at()** with a numeric argument returns the Nth item of its list.

.lowercase

Returns a copy of the string, transforming all capital letters to lower case.

See also .capitalize and .uppercase.

.next

.next generates convenient note names and other strings in a sequence. For example:

`"footnote".next` ➛ "footnote 1"

`"footnote 1".next` ➛ "footnote 2"

Specifically, .next searches a string for its last run of digits. If no digits are found, .next returns the string followed by " 1". Otherwise, the number is increments and placed in the same position in the string.

`"Agent 007 (active)".next` ➛ "Agent 008 (active)"

.nounList

The .nounList operator returns a list of each noun in the string, exception those recognized as pronouns and proper names. This operator is currently available in English and several other languages. It is available on macOS 10.14 and later.

See also .wordList.

.paragraph(N)

Returns the nth paragraph in the string. The first paragraph is .paragraph(0). Only available in macOS 10.13 and later.

.paragraphCount

Returns the number of paragraphs in a string. Only available on macOS 10.13 and later.

.paragraphs(N)

Returns first N paragraphs of the string.

,paragraphList

.paragraphList

The .paragraphList operator returns a list of paragraphs contained in a string. It is available on macOS 10.14 and later.

.plain

Removes bold, italic, and strikethrough style attribute to the string. This has no effect unless the string is placed in $Text, perhaps via **.replace()**.

.replace(pattern,replacement[,arguments])

If the regular expression "pattern" is found in the string, the matching portion of the string is replaced by the replacement string. Additional arguments allow replacement of multiple sub-expressions assuming extra back-references are created by "pattern".

.reverse

Reverses the sequence of characters in the string.

`"ostrich".reverse()` "hcirtso"

The .reverse operator is most useful for gaining access to a part of a string relative to the end of the string, when the convenient operator provides access from the start. For example, $Text.paragraphs(1) returns the first paragraph of the text, and

```
$Text.reverse.paragraphs(1).reverse
```

returns the final paragraph.

.sentence

The operator **.sentence(n)** extracts the nth sentence from a string. For example, if $Text contains the passage

> Mr. Smith went to Washington. He shook hands. He kissed babies.

then $Text.sentence(0) returns "Mr. Smith went to Washington." and $Text.sentence(2) returns "He kissed babies."

If the argument is omitted, the initial sentence is returned.

The definition of “sentence” is heuristic, and varies depending on the locale. In the example above, for example, Tinderbox recognizes that the period following "Mr." ends an abbreviation, not a sentence.

.show

[String].show lets actions display a transient message. This can be especially useful when constructing complex actions. When called, a small placard slides up from the bottom of the main window, in which the message is displayed. The placard remains in view for five seconds, and is then automatically hidden. If a second message is shown while the placard is visible, the placard remains visible for an additional five seconds, and the new message replaces the old message.

Optional arguments specify the background and text colors for the display. For example:

"Error: overdue tasks".show("bright red", "black")

.size

Returns the number of characters in a string.

.split("pattern")

Splits a string into a list, dividing the string at designated "pattern"; the matched "pattern" characters are discarded.

`"lions and tigers and bears".split(" and ")` "lions;tigers;bears"

.strike

Adds the strikethrought style attribute to the string. This has no effect unless the string is placed in $Text, perhaps via **.replace()**.

.substr(start,length)

Extracts substrings form the string, starting at character number start (zero-based) and including up to length characters. A negative start value (1-based) may be used to cite a start counted from the end of the string.

> `"Hello, world".substr(0,5)`: Hello

If the second argument is absent, the result runs from start through the end of the string.

> `"Hello, world".substr(7)`: world

If start is negative, the substring begins -start characters from the end of the string.

> `"Hello, world".substr(-5)`: world

If the length is negative, it is treated as relative to the end of the string.

> `"Hello, world".substr(0,-2)`: Hello, wor

If the requested substring exceeds the length of the string, Tinderbox trims the request to the actual length of the string. If the requested substring lies entirely outside the range of the string, Tinderbox return the empty string "".

.trim

.trim()

The .trim operator removes unwanted characters from the start and end of a string. With no arguments, .trim removes whitespace and end of line characters.

`"Hello world ".trim` ➛ `"Hello world"`

If used with the argument “punctuation”, .trim also removes punctuation marks.

`"[tab] Hello world?! ".trim(punctuation)` ➛ `"Hello world"`

If used with the argument “quotes”, .trim also removes quotation marks " and ' that occur at the start and end of a string.

.toNumber

Converts a string of digits to the corresponding number. If the string is empty, toNumber() returns zero. If the string cannot be parsed as a number, the result is undefined.

.tr(characters,replacements)

Replaces every occurrence of a character with a replacement character.

"Hello, world".tr(w,W) : Hello, World

Multiple characters may be replaced at once:

"Hello, world".tr("aeiou","AEIOU"): HEllO, wOrld

If no corresponding character appears in the replacement string, the matched character is deleted.

"Hello, world".tr(aeiou): Hll, wrld

.uppercase

Returns a copy of the string, transforming all lowercase letters to upper case.

See .capitalize() and .lowercase().

.wordCount

The .wordCount operator returns the number of words in a string.

The string is first scanned for its dominant language, and the conventions of that language are used for counting words. If the language is unknown or if it cannot be determined, Tinderbox uses the conventions for English.

.wordList

The .wordList operator returns a list of each word in the string. If assigned to a set, the set will contain a list of each unique word in the string. This operator is currently available in English and several other languages. It is available on macOS 10.14 and later. See also .nounList.

.words(N)

Returns the first N words of the string, if the string has at least N words; otherwise, returns the entire string. If N is negative, returns the last N words of the string.

"Now is the winter of...".words(2) ➛ "Now is"

escapeHTML(string)

HTML and XML cannot contain certain reserved characters. This function converts any such characters found in the string to the corresponding XML entities. For example, '<' is replaced by '<' and '&' is replaced by '&'.

format(value, format-string)

Format(value,format_string) converts attributes to strings and lets you choose exactly how you prefer the data to appear. format() is useful for export and for preparing data for other programs and Web services.

For numbers, the first argument is the number of decimal places to which the number should be rounded. An optional third argument allows you to specify a total width for the formatted string.

For example, if $MyNum is 3.1415927, then

format($MyNum,2,7) is " 3.14"

format($MyNum,2) is 3.14

format($MyNum,0) is 3

An option fourth argument specifies the padding character to be used if needed. for example

format($MyNum,2,7,0) is "0003.14"

If value is a date, the format string is the same as the format used by Tinderbox's date export codes.

L: local time, in long format, using the system format settings (example: Tuesday, April 29, 2003.)
l: local time, in short format, using the system format settings (example: 4-29-03)
d: day of the month (example: 29)
D: formats date 01-31, with leading zero
m: number of month (example: 4)
M: abbreviation of month (example: Apr)
MM: name of month (example:April) w: abbreviation of weekday (example: Tue)
M0 (em-zero): formats month numerically 01-12, with leading 0
W: name of weekday (example: Tuesday)
y: year (example: 2003)
t: time, in local format (example: 2:32 pm)
h: hour of the day on a 24-hour clock (example: 13:39)
H: hour of the day on a 12-hour clock (example: 1:39)
mm: minute of the hour (example:05 for five minutes after the hour)
s: second
p: AM or PM
*: date/time in RFC 822 format (example: Thu, 18 Feb 2004 19:12:00 0500)
=: date in ISO 8601 format (example: 2004-02-18)
U: date as Unix epoch (seconds since 1 January 1970)

For example:

```
format($Created,"L")
```

formats the note's creation date a "long local date" such as "Sunday, March 23, 2007 1:26pm".
If value is a list or set, the format string is the delimiter that is to be used to separate discrete list elements

```
format($DisplayedAttributes,",")
```

converts displayed attributes to a comma-separated list.
Optionally, you may supply four quoted string arguments besides value to format the set as an HTML list:

```
format(value,list-prefix,item-prefix,item-suffix,list-suffix)
```

For example

```
format($MyList,"
    ","
  • ","
    ","
")
```

If value is a number, then the arguments are numeric and interpreted as follows:
`format(value,precision, width)`
In quoted string arguments here (as elsewhere), \" is converted to a quotation mark, \n to a carriage return, and \t to a tab.

idEncode(string)

Converts a string to form suitable for use as an HTML identifier. Punctuation and other characters that are neither letters or digits are replaced by underscores.

urlEncode(string)

The function
`urlEncode(value)`
returns a string in which characters (such as spaces) that may not occur in an internet URL are replaced by their encoded equivalents.

wordsRelatedTo()

The function

```
wordsRelatedTo("word")
```

returns a list of words that are related to a specific word. The relationship is not necessarily synonymous, and is based loosely on the likelihood of the two words appearing in close proximity.
The function is available only in macOS 10.14 and later, and only in a few supported languages. These currently include English, Spanish, Portuguese, French, and Simplified Chinese.
An optional second argument requests a specific number of related words:

```
wordsRelatedTo("constitution",5)
```

A common idiom is

```
$Text.contains(wordsRelatedTo("sprocket"))
```

which would search for sprocket, flange and pulley.

String Processing

Tinderbox's new string processing operators are intended to help extract information from structured and semi-structured text. This text may be hand-typed, for copied from sources like email. Often, it may be imported from other programs or downloaded from web services to a Tinderbox attribute. We want to take this information and extract information we need.

Broadly speaking, our approach is to begin at the start of the string and proceed, step by step, following a recipe. For example, we might say:

1. Read until you find a line that begins with "To:", "From:", or "Subject".
2. If you find a "To:", get everything up to the next space character and save it in $Email.
3. If you find a "From", get everything up to the next space character and save it in $EmailFrom.
4. If you find a "Subject:", get the rest of the line and use that for the $Name of this note.
5. Once you've found the subject, delete all the headers you've processed and leave the rest of the text.
6. If you never find a "Subject:", don't delete anything.

All string processing operators accept a string, called the *stream*, of text being processed. They return the unprocessed remainder of the stream, which may be passed to another operators. For example:

```
$MyString.skip(23).captureNumber("MyNumber")
```

Takes the value of MyString, skips exactly 23 characters, and extracts a number to be stored in $MyNumber.

Failure

As we process a string, we may find that the string is not what we expected. The .fail operator raises a flag to say, "this process did not work." The .failed operator is true if the current processing tasks has failed.

The Expect Operators

The expect family of operators looks at the current place in the text stream to ensure that the text is what you expect it to be.

```
[String].expect("abc")
```

Tests that the next characters in the stream are "abc", and advances the stream beyond "abc". Otherwise, fails.

```
[String].expectWhitespace
```

Tests that the next characters in the stream are whitespace, such as spaces, tabs, and carriage returns. Advances the stream to the first non-whitespace character. Fails if the next character is not whitespace.

```
[String].expectNumber
```

Tests that the next characters in the stream can be interpreted as a number. Numbers include "0", "5.7", and "-13". Skips over the number. Fails if the next character is not part of a number.

```
[String].expectWord
```

Tests that the next character is not whitespace, punctuation, or a digit. Skips to the next whitespace, punctuation, or digit. Fails otherwise.

The Capture Operators

The capture family of operators extract a chunk of information from the stream and store them in an attribute.

```
[String].captureLine(attribute)
```

Captures text from the current position, up to the next line ending or the end of the string. The text is stored in the designated attribute, and the stream advances to the character following the line break. Fails if the stream is empty.

```
[String].captureNumber(attribute)
```

Captures a number from the current position, up to the next character that cannot be considered part of a number. Advances the stream to that character. Fails if the stream cannot be interpreted as a number.

```
[String].captureWord(attribute)
```

Captures text from the current position up to the next character that is a digit, whitespace, or punctuation. Fails if the stream is empty or begins with a digit, whitespace, or punctuation.

```
[String].captureToken(attribute)
```

Captures text from the current position up to the next character that whitespace. Fails if the stream is empty or begins with a whitespace.

```
[String].captureRest(attribute)
```

Captures all remaining text in the stream.

```
[String].captureTo(delimiter,attribute)
```

Captures all text up to the one of the characters in the delimiter string. If no delimiter is found, captures all remaining text in the stream. Fails if the stream is empty.

The Skip Operators

The skip operators advance the current position of the stream.

[String].skip(n)

Skips exactly n characters. Fails if the stream is exhausted.

[String].skipTo("target")

Advances the stream to the first occurrence of the target string, skipping over the target string as well. Fails if the target is not found.

[String].skipWhitespace

Advances the stream to the first character that is not whitespace. Fails if the stream is exhausted.

[String].skipLine

Advances the stream to the next carriage resturn. Fails if the stream is exhausted.

[String].skipToNumber

Advances the stream to the next number. Fails if the stream is exhausted.

.extract

The operator **.extract(*pattern[, ignore_case]*)** returns the first matched subexpression of a regular expression. If the regular expression has no subexpressions, the entire match is returned.

For example, $Email=$Text.extract("To:(.*)") looks through this note's Text for the letters "To" followed by a colon. If found, it gathers everything that follows the colon, up to the next carriage return, and stores the result in $Email.

If the optional second argument is true, the regular expression search is case-insensitive.

.extractAll

The new operator **.extractAll(*regex[, case_insensitive]*)** returns a list of all matches for a pattern found in a string. For example:

```
$Text.extractAll("#[A-Za-z]+")
```

would return any hashtags such as "#Tinderbox" found in the text.

If the optional second argument is true, the regular expression search is case-insensitive.

.eachLine

[String].eachLine(x[:condition]){action}

Iterates through each line of a string, where a line is one or more characters ending in a carriage return, line feed, unicode paragraph separator, or the end of the string. Each line, in turn, is bound to a temporary variable, and the action is then performed.

If the optional condition is specified, only lines that satisfy the condition are passed to the action.

For example

```
$MyNumber=0; $Text.eachLine(x){$MyNumber=$MyNumber+1;}
```

will set $MyNumber to the number of lines in the $Text of this note.

```
$MyNumber=0; $Text.eachLine(x:x.contains("@")){$MyNumber=$MyNumber+1;}
```

will set $MyNumber to the number of lines in the $Text that contain the "@" symbol.

try{}

This operator saves the value of an attribute while one or more actions are performed. If any of those actions invokes the action fail(), then the saved attribute is restored to its previous value. For example, consider an imported note whose text contains formatted data — perhaps JSON or XML.

```
$Text.try{.....try to parse text.... ; $Text="";}
```

Would attempt to parse this data. If successful, the text would be discarded. If the data cannot be parsed, the text would be restored to its former state.

```
$Text.try{...}.thenTry{...}.thenTry{...}
```

If the first try{} clause fails, successive thenTry{} clauses are attempted until one succeeds. If the first try{} clause succeeds, the successive thenTry{} clauses are ignored.

JSON

Tinderbox can now process JSON stored in any attribute, including $Text.

[$String].captureJSON;

Attempts to parse the string as JSON. Fails if unsuccessful. The parsed JSON is saved as the current JSON item; only one JSON item may be current at any time. Returns the rest if the string following the JSON.

[$String].json[key]

[$String].json[n]

If there is no current JSON object, attempts to parse the string as JSON and fails if unsuccessful. If there is a current JSON object, that object will be reused.

If the top-level element is an object, [$String].json[key] return a dictionary for that object. If the top-level element is an array, [$String].json[n] returns in

nth element of that array.

For example, if the text is `{ "title":"Becket", "price": 9.95 }`

Then `$Text.json['title']` is "Becket".

Nested objects may be identified using a sequence of keys.

For example, consider:

```
{
 "person": { "firstName": "Thomas",  lastName: "Roe"},    "coordinates" : [-90,41]
 }
```

We may obtain Thomas's latitude as

```
$Text.json["coordinates.0"]
```

And his family name as

```
$Text.json["person.lastName"]
```

[$String].json.each {....}

If the top-level element is an array, rebinds the JSON object in turn to each array element. After calling the action block for each element, the JSON object is restored.

For example, if the text is

`[ ["price":1], ["price",2]]`

Then

```
$Text.json.each{$MyNumber += json["price"]}
```

would add 3 to $MyNumber.

An optional path argument supplied a path to the array to be iterated. For example, if that $Text is

```
{
 "person": { "firstName": "Thomas",  lastName: "Roe"},    "coordinates" : [-90,41]
 }
```

then `$Text.json.each(coordinates){...}` would iterate through the array of coordinates.

If the string argument is omitted, `json.each(){}` reuses the current JSON object. This can be much faster than repreatedly reparsing a complex json package.

XML

Tinderbox can now process XML stored in any attribute, including $Text.

In the following examples, we will consider the following sample XML:

```
<shelf>
 <book price="9.95">War and Peace</book>
 <book price="4.95">No et Moi</book>
 <audio price="14.95">Born To Run</book>
</shelf>
```

[$String].captureXML;

Attempts to parse the string as XML. Fails if unsuccessful. The XML is saved as the current XML item; only one XML item may be current at any time. Returns the empty string, since an XML object may contain nothing outside of the XML.

[$String].xml(path)

If there is no current XML object, attempts to parse the string as XML and fails if unsuccessful. If there is a current XML object, it will be reused.

Returns a specific piece of data from the XML object, determined by the path. **Path** should be an expression or a quoted string. **path** is a subset of the Xpath standard, as follows:

/shelf/book: returns a list of children if the **shelf** object that are books. "War and Peace;No et Moi"

/shelf/book[2]: returns the child of the shelf object that represents the second book.

/shelf/book@price: returns the price attribute of each book.

Note: to accord with standard XML and XPath usage, the first child of an XML node is child [1], not [0].

Note: XML attributes are not related to Tinderbox attributes.

[$String].xml.each('/shelf/book') {....}

Locates the XML object for each book in the shelf. Invokes the action block with the xml item bound in turn to each book element. On completion, restores the XML object to its previous state.

Inside the action clause, you can refer to the value of the iterated item as **xml[]**, and the attribute of that item as **xml[@attribname]**.

Numeric Operators

Numeric Functions

The familiar mathematical operators + - * / operate on numeric expressions in the expected way.

Note that + and - have different meanings when applied to sets, and that + has yet another meaning (concatenation) when applied to strings.

Mathematical Operators

In all the following, the N argument is a number.

`abs(N)` computes the absolute value of is argument.

`atan(N)` computes arctangent, and returns its result in radians.

`ceil(N)` returns the first integer that is larger than N. This may also be used as a dot operator: $ceil

`degrees(N)` converts radians to degrees.

`exp(N)` returns the exponential of N, e^n.

`floor(N)` returns the first integer that is smaller than N. This may also be used as a dot operator: $MyNumber.floor

`log(N)` computes the natural logarithm of its argument, which should be a number, a numeric attribute, or an expression that can be interpreted as a number.

`mod(N,dividend)` computes the remainder of A divided by B.

`pow(N,K)` raises N to the power of K.

`radians(degrees)` converts its argument, in degrees, to radians.

`rand()` returns a pseudo-random number between 0 and 1.

`rand(N)` , where N is an integer, returns a pseudo-random integer between 0 and N-1.

`round(N)` rounds the value of its argument to the nearest integer. This may also be used as a dot operator: $MyNumber.round

`sin(N)`, `cos(N)`, and `tan(N)` are the familiar trigonometric functions, taking arguments in radians.

`sqrt(N)` computes the square root of its argument.

sum(), sum_if, avg and avg_if()

In the following operators, the parameter *expression* may be an attribute reference, a literal number, or an expression.

count_if(group,expression) returns the number of items in a group that satisfy an expression. For example,

```
sum(child,$WordCount>100)
```

counts the number of a note's children thaty have more than 100 words in their text.

sum(group,expression) computes the sum of a series of expressions. For example,

```
sum(child,$WordCount)
```

computes the total word count of all the children of a note. 'group' may be any group designator {children,descendants,sibling,ancestor,all}. In addition, 'group' may take a single argument that designates a particular note other than this; for example

```
sum(children(/agents/books),$TextLength)
```

sum_if(group,condition,expression) computes the sum of the values for *expression* for each item in the group, ignoring any notes for which if condition is not true. For example:

```
sum_if(children, $Checked, $Price)
```

totals the prices of all the note's children which have been checked.

avg(group,expression) and **avg_if(group,condition,expression)** compute the mean of the values for *expression* for each item in the group. For example:

```
avg_if(group,condition,expression)
```

computes the sum or the mean of a value computed in each note for a group that meets the designated condition.

```
sum_if(child,$Checked,$Qty*$Price)
```

```
avg_if(child,$Checked,$Qty*$Price)
```

compute the total price and the average price of checked items that are children of this note.

Converting Numbers to Strings

[Number].format(decimal_places[, width]) converts a number to a string. decimal_places specified the number of digits to be displayed after the decimal point, and may be zero to display an integer. If the optional width is specified, additional spaces will be added to the left of the number to pad the result to a minimum width. For example:

`$Price.format(2)`

converts the number $Price to a string, rounding to two decimal points.

An optional fourth argument allows you to specify the padding character to be used to fill the specified width. For example, if $MyNumber is 7, then

`$MyNumber.format(0,3,0)`

will return the string "007".

If the argument to **format** is "l", the number is formatted using local numeric formatting conventions. For example, commas are inserted before thousands and millions in the US.

If the argument to **format** is "$", the number is formatted using local currency formatting conventions. For example, the number 1.5 would be formatted as $1.50" in the US, but "£1.50" in the UK.

If the argument to **format** is “$0”, the number is formatted using local currency and rounded to the nearest whole unit. For example, 1000.4 would be formatted as “$1,000” in the US, as the $ and $0 formats also use the locale's thousands seaprator.

If the argument to **.format** is "X", the number is formatted as a Roman numeral.

If the argument to **.format** is "o", the number is formatted as an ordinal, such as “1st” or “7th”.

Color Operators

Color Components

Color.red

Color.green

Color.blue

Three read/write properties allow the individual RGB color channel values to be read or set. The component value is a number (0-255). Values may set using a number 0-255 or a hex number string "#00" - "#ff".

```
$MyNumber=$Color.red
```

```
$Color.red = 255
```

Color.brightness - Read/write property allows a color's brightness level to be read or set using a value in the 0-100 range representing a percentage.

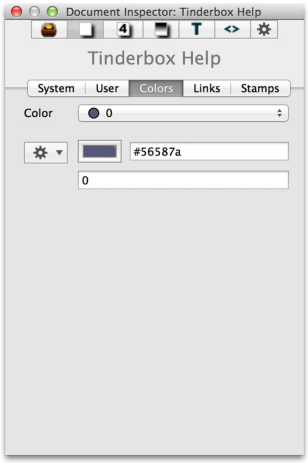
Color.saturation - Read/write property allows a color's saturation level to be read or set using a value in the 0-100 range representing a percentage.

Color.hue - Read/write property allow a color's hue to be read or set using a value in the 0-360 range representing a a circular degree.

Converting Colors to Strings

Color.format() - converts a color attribute to a string representation.

Converting Strings to Colors



Tinderbox provides several flexible notations for interpreting colors.

Named colors are the preferred way to set and change colors in Tinderbox. Named colors may be redefined, and new named colors added or deleted, in the Colors pane of the Document Inspector.

Hex colors, customarily used on the Web, begin with a # sign followed by six hexadecimal digits: #rrggbb, corresponding to the proportion of red, green, and blue in the color. #FFFFFF is white, #800000 is dark red, and #000000 is black.

RGB colors of the form `RGB(red,green,blue)` use decimal colors from 0 to 255 to describe the proportion of red, green, and blue in the color.

HSV colors of the form `HSV(hue,saturation,value)` describe the hue, saturation, and value of the color. Hue is a color wheel angle, in degrees, running from 0 to 360. 0 corresponds to red, 120 to blue, and 240 to green. Saturation and value run from 0 to 100. The value is the equivalent to brightness in `HSB(hue, saturation, brightness)`.

Manipulating Colors

rgb(red,green,blue) - creates a color. Its arguments are integers ranging from 0 to 255. Arguments may also be expressions or attribute references.

```
$Color=rgb($MyRed,255,255)
```

Date Operators

Date Properties

Read/write properties allow various aspects of data/time data to be read or set.

Date.day (1–31)

Date.month (1–12)

Date.year (value of year)

Date.hour (0–23)

Date.minute (0–59)

Date.second (0–59)

Date.weekday (1–7) Monday = 1, Sunday = 7. Read-only

Date.week (1–52) the number of the week within the current year. Read-only.

If the Date is "never", all functions return 0 (zero). For example:

```
$Width = $DueDate.month
```

```
$DueDate.day=1;
```

Date Operators

Date**.format("format-string")** converts a date to a string, using the quoted date format string. For example, `$DueDate.format("l")` will format the value

of $DueDate using the system's short date format.

L: local time, in long format, using the system format settings (example: Tuesday, April 29, 2003.)

l: local time, in short format, using the system format settings (example: 4-29-03)

d: day of the month (example: 29)

D: formats date 01-31, with leading zero

m: number of month (example: 4)

M: abbreviation of month (example: Apr)

MM: name of month (example:April) w: abbreviation of weekday (example: Tue)

M0 (em-zero): formats month numerically 01-12, with leading 0

W: name of weekday (example: Tuesday)

y: year (example: 2003)

t: time, in local format (example: 2:32 pm)

h: hour of the day on a 24-hour clock (example: 13:39)

H: hour of the day on a 12-hour clock (example: 1:39)

mm: minute of the hour (example:05 for five minutes after the hour)

s: second

p: AM or PM

*****: date/time in RFC 822 format (example: Thu, 18 Feb 2004 19:12:00 0500)

=: date in ISO 8601 format (example: 2004-02-18)

U: date as Unix epoch (seconds since 1 January 1970)

date(year,month,day,hour,min) constructs a date from individual numeric elements. This is useful, for example, if you need to assemble a date from separate attributes. year is the 4-digit year, month is a number from 1–12. The time arguments are optional, and are specified in a 24-hour clock.

`date(2004,7,23,16,45)` is 23 July 2004 4:45pm

date("string") constructs a date from string which is a string literal or string expression. Usually, this is not necessary as Tinderbox will coerce the string to a date type automatically. In some contexts, though, it may be more convenient or more clear to make the conversion explicit. (See format() to convert dates into strings)

Date/time comparison operators

days(date1,date2) returns the number of days that elapsed between date1 and date2, as a number. If date2 is the earlier date, the result is negative. The internal comparison includes the time portion of the date. thus if 2 dates are 1 day apart but the times are less than 24 hours aport, the result is 0 and not 1 as might be expected.

weeks(date1,date2) returns the number of weeks that elapsed between date1 and date2, as a number. If date2 is the earlier date, the result is negative.

hours(date1,date2) returns the number of hours that elapsed between date1 and date2, as a number. If date2 is the earlier date, the result is negative.

minutes(date1,date2) returns the number of minutes that elapsed between date1 and date2, as a number. If date2 is the earlier date, the result is negative.

seconds(date1,date2) returns the number of seconds that elapsed between date1 and date2, as a number. If date2 is the earlier date, the result is negative.

months(date1,date2) returns the number of moths that elapsed between date1 and date2, as a number. If date2 is the earlier date, the result is negative.

years(date1,date2) returns the number of minutes that elapsed between date1 and date2, as a number. If date2 is the earlier date, the result is negative.

time(date1,hours,minutes) returns a date that is the same calendar day as **date1** but has the time set to **hours** and **minutes**.

time(date1,"hh:mm") returns a date that is the same calendar day as **date1** but has the time set to the time hh:mm.

Locales

When converting between strings and dates, Tinderbox uses a variety of formats including the date format used in your locale as specified in System Preferences:Language and Region.

When practical, you should adapt these preferences to your needs. In particular, if you frequently use historical dates, you should avoid formats with two-digit years.

The **locale()** action allows you to check your current locale setting and to change it. For example, if you are a Swedish historian working in Australia you might want the Tinderbox notes for your next article to use Swedish date formats with your task list might use Australian settings.

Interval Operators

Interval Operators

Two intervals may be combined with the + operator, and the difference between two intervals may be obtained using the - operator. Intervals may be multiplied or divided by numeric values as well. If $MyInterval is five minutes, `4*$MyInterval` is twenty minutes.

The operators **.day**, **.hour**, **.minute**, and **.second** convert the interval to familiar units. For example, if $MyInterval is 12:00:00 -- twelve hours -- then $MyInterval.day will be 0,5, $MyInterval.hour will be 12, $MyInterval.minute will be 12×60=720.

interval(s) converts a string **s** to an interval. For example, **interval(20:15)** returns an interval of twenty minutes and fifteen seconds.

interval(start,end) returns the interval between two dates, **start** and **end**.

.format(*formatString*) converts intervals to strings according to the supplied format string. Possible values include

I the **local** positional format, such as 1:05

L the local long format, such as “1 minute, 5 seconds”

List and Set Operators

A *list* is a list of strings or tags separated by semicolons.

```
Arwen; Eowyn; Galadriel
```

A *set* is a *list* where duplicate values are not allowed. A given element can appear in a set at most once; duplicates are automatically removed.

When you wish to make it clear to Tinderbox that something is a list or set, enclose it in square brackets. In particular, this allows you to make lists of lists:

```
[ [Red Sox; Yankees; Orioles] ; [Cubs ; Mets; Braves] ]
```

The order of elements in a *set* is not significant, and Tinderbox controls the order of elements in a set. You determine the order of elements in a list.

Two *lists* are equal if they contain the same elements in the same order. Two *sets* are equal if they contain the same elements.

Adding and Removing Items

The operators + and - add and remove items from lists and sets.

```
$MySet = $MySet + "important";
```

adds the value “important” to $MySet. Since sets cannot contain duplicates, this will have no effect if $MySet already contains "important".

```
$MyList = $MyList + "important";
```

adds "important" to the end of $MyList. Since lists can contain duplicate entries, this always changes $MyList.

You can add more than one tag at a time:

```
$MySet = $MySet+"important;overdue";
```

You can retrieve or set a specific item of a list:

```
$MyString=$MyList[2];
```

```
$MyList[2]="rock";
```

You can also add a list or set to another list or set:

```
$MySet = $MySet + $UrgentTags
```

Similarly, subtraction removes tags from a set or list:

```
$MySet = $MySet - "important"
```

removes "important" from $MySet. If $MySet does not contain "important", this action will have no effect.

The **.extend** operator adds a list without flattening it.

```
[1;2].extend)[3;4;5] ➛ [1; 2; [3;4;5] ]
```

```
[1;2]+[3;4;5] ➛ [1; 2; 3;4;5 ]
```

The Range Operator

The range operator ... constructs a list of numbers from a specified starting point to a specified end point.

```
1...3 ➛ 1;2;3
```

```
3...1 ➛ 3;2;1
```

The range operator can be useful for performing a task a specific number of times.

```
1...10.each(x){ var path="/container/item "+x;create(path);}
```

The range operator binds more tightly than arithmetic operators.

```
1...3 * 2 ➛ (1..3) *2 ➛ 2;4;6
```

List multiplication

Lists of numbers may be multiplied by a number. For example:

```
$MyList="1;2;3";
```

```
$MyList=$MyList*2; // 2;4;6
```

List/Set.contains("item") - true if any element of the List or Set matches the quoted string "item" exactly. Matches are case-sensitive. Match values must be enclosed in quotes. The returned value is the zero-based offset of the match +1.

```
"Arwen;Eowyn;Galadriel".contains("Eowyn") 2
```

"Arwen;Eowyn;Galadriel".contains("Frodo") 0

List/Set.asString() – returns a string representation of a list or set. This can be useful, for example, if you want to use **.contains** to search for a regular expression or a specific substring, rather than the presence or absence of a member. "Eowyn;Frodo".asString.contains("w") is true, though "Eowyn;Frodo".contains("w") is false.

list.at(n) – returns the nth item in the list. If n is zero, returns the first item. If n is negative, returns the nth item from the end of the list. If the list does not contain n items, returns the empty string.

list[n] – returns the nth item in the list. If n is zero, returns the first item. If n is negative, returns the nth item from the end of the list. If the list does not contain n items, returns the empty string.

list.at("keyword") – extracts a value from a list, string, or set of keyword-value pairs in which the keyword and value are separated by colons. (This feature is deprecated as of Tinderbox 9.6; prefer dictionaries instead.)

For example, suppose

```
$MyString = "USA: North America; Canada: North America; France: Europe; Japan: Asia; default: unknown"
```

Then

```
$MyString.at("France") ==> "Europe"
$MyString.at("Canada") => "North America"
```

The special keyword "default" applies to any keyword not contained in the list. For convenience, when several keywords have the same value they may be separated by pipe characters "|":

```
"USA | Canada: North America; ..."
```

List/Set.each(n){...actions... } – performs an action for each element of the list or set. The local variable "n" is bound to each element in turn. For example:

```
$Result=0; $MyList.each(x){ $Result=$Result+x;}
```

adds each element to $Result

List/set.any and **List/set.every** let you test each item in a list or set.

$MyList.every(x,x>5)

is true if every element in $MyList is greater than 5. The local variable "x" is bound to each element of the list or set in turn.

$MyList.any(x,x>5)

is true if any element in $MyList is greater than 5. The comparison may also be applied to lists of strings:

"apple; pear; plum".every(x, x>"aardvark")

Is true because every element follows "aardvark" in alphabetical order.

If the target list or set is empty, .any() always returns false, and .every always returns true.

[List].collect(label, expression) binds label in turn to each element in the list. The expression is evaluated for each item, and a list of the results is returned.

```
$MyList="1;2;3;4;5";
MyList.collect(x,x*x) ➛ 1;4;9;16;25
```

[List].collect_if(label, condition, expression) binds label in turn to each element in the list. The condition is evaluated, and if it is true then the expression is evaluated . A list of the results is returned.

```
$MyList="1;2;3;4;5";
MyList.collect_if(x,x<3,x) ➛ 1;2
```

List/Set.count counts the number of element in a list or set

List/Set.count_if(var, test) counts the number of elements in a list or set that satisfy the test condition. For example, $MyList.count_if(x, x>100) will return the number of elements of MyList that are greater than 100.

List/Set.sum_if(var, test) computes the sum of the elements in a list or set that satisfy the test condition. For example, $MyList.sum_if(x, x>100) will return the sum of any elements of MyList that are greater than 100.

List/Set.sum_if(var, test,expression) allows you to transform the element before it is added. For example, $MyList.sum_if(x, x>100,sqrt(x)) will return the sum of the square roots of elements of MyList that are greater than 100.

List/Set.empty - true if the List or Set has no elements.

List.first returns the first element of a list.

List.first(n) returns the first **n** elements of a list.

List.slice(n) returns the portion of the list commencing with the nth element. If n is negative, we count from the end of the list.

List.slice(n,count) returns the portion of the list commencing with the nth element and continuing for **count** elements. If n is negative, we count from the end of the list.

List/Set.format("delimiter") - converts a List or Set to String-type data, concatenating each value with the specified "delimiter" characters. thus if "delimiter" is ", " the source data becomes as comma+space delimited list as a single String.

List/Set.format("list-prefix", "item-prefix", "item-suffix", "list-suffix") - converts a List or Set to a String. The list begins with the "list-prefix" and ends with "list-suffix"; each item is preceded by "item-prefix" and followed by"item-suffix".

List/Set.icontains("item") - true if any element of the List or Set matches the quoted string "item" exactly; differences between upper and lower-case letters are ignored. Regular expressions are not supported (unlike with pure string operations). Matches are case-insensitive. Match values must be enclosed in quotes. The returned value is the zero-based offset of the match +1. The latter is to ensure that matches are always 1 or more and thus also

coerce to 'true' for the true/false element of testing for match.

List/Set.intersect(list) - returns the intersection of two sets or lists — a set or list of the values shared by each.

List.last returns the last element of a list.

List.last(n) returns the last **n** elements of a list.

List/Set.replace("item","replace") - replaces the designated item with a replacement value. Matches are only the whole list values; partial value matches are not supported. Match values must be enclosed in quotes.

List/Set.remove(element) – creates a new list by removing all appearances of an element, or a list of elements, from a list or set. If $MyList is 1;2;3;4 `$MyList.remove(3)` returns "1;2;4".

List/Set.select() - selects the notes designated in a list of paths. "this".select() selects this note. "".select() clears the selection.

List/Set.size - the number of values in the the list. Read-only property.

List/Set.sum - adds up lists of numbers. If $MyList is 1;2;3;4, then `$MyList.sum` is 1+2+3+4=10. Read-only property.

List/Set.avg - calculates the mean value of a list of numbers. If $MyList is 1;2;3;4, then `$MyList.avg` is 2.5. Read-only property.

Sorting

Lists (but not sets) may be sorted. The following sort operations apply only to lists:

List.sort - case-sensitive sort

List.isort - case-insensitive sort().

List.nsort([expression]) - nsort() behaves exactly like sort(), but assumes that the list elements, are numbers.

List.sort([expression]) - if List is a list of note names or $Paths, returns a list sorted by the corresponding expression (or attribute value). If expression is a Number-type, a numeric sort is used; if Date-type, a date sort is used. Otherwise, the list is sorted lexically.

List.isort([expression]) - as above but case-insensitive.

List.nsort([expression]) - numeric sort. An nsort() behaves exactly like sort(), but assumes that the list elements, or the result of the optional expression, are numbers. Thus an nsort() always sorts numerically.

List.reverse - reverses the sequence of items in the list.

For example, suppose $Favorites is a list of favorite notes.

```
$Favorites.sort($Price)
```

returns a list of notes sorted by price with the lowest price first, and

```
$Favorites.sort($Price).reverse
```

is sorted by price with the highest price first.

List.unique() - returns a new list that contains each unique item found in the source list. Duplicate entries are not copied. Note that all items in a Set are, by definition, unique.

List Operations

Several functions are useful for extracting notable elements from sets and lists. In the following, the list argument implies set or list data.

count(list) returns the number of items in list.

max(list) returns the largest item in list

min(list) returns the smallest item in list

Equivalently, you may write:

```
$MyList.max
```

```
$MyList.min
```

```
$MyList.count
```

If **max(list)** or **min(list)** is evaluated in a numerical context, numerical comparison is used. Otherwise, Tinderbox uses lexical comparison.

The mathematical functions **sum()**, **sum_if()** and **avg_if()** take a group scope (i.e a list of $Paths) so can be used with lists—see Numeric Functions further above.

Building Lists

collect(group,expression) builds a list by visiting each note described by group and adding the value of the designated attribute to the list. For example,

```
collect(children,$Name)
```

constructs a list with the name of each child of the note. group may be any group designator: {children,descendants,sibling,ancestor,all}. In addition, group may take a path modifier; for example

```
collect(children(/agents/books),$Name)
```

expression can be any expression, but is typically an attribute.For example,

```
collect(children,$Name)
```

```
collect(children,$Width*$Height)
```

collect_if(group,condition,expression) constructs a list by collecting all the notes corresponding to group, testing each note to see if it meets condition, and adding the expression to the set for each such note. For example

```
collect_if(children,$Status="Important",$Name)
```

will construct a set of the names of all of this note's important children.

Should a unique list of values be required, pass the output to a Set to remove duplicate values.

find(query) is equivalent to

```
collect_if( all, query, $Path)
```

Note that collect_if is related to agents; many tasks you might perform with collect_if can be done as well, or better, with an agent. The results of an agent can be reused by other agents, while find() and collect_if() need to start from scratch each time they are run.

values("attribute_name") returns a set of unique values stored in that attribute throughout the document. If the attribute is a set or a list, values() returns a set of unique elements used throughout the document; if the attribute is a any other type, values() returns each of the distinct values in use.

values(group_designator, "attribute_name") returns a set of unique values stored in that attribute in a specific part of the document. For example, values(children,"Subtitle") returns a list of the subtitles of the children of *this* note.

Logical Operators

Boolean operators & (and), | (or), and ! (not) operate as expected. Boolean assignment (and queries) coerce non-boolean attributes to either to or false.

number	zero is false and any non-zero is true.
string	"" and "false" are false, other values are true
set	empty sets are false, all others are true
date	"never" is false, all others are true
color	"black" (or RGB(0,0,0) or "#000" or "‹000000") is false, others are true

Logical equality tests

Use == to equality and != to test inequality. The older form of using a single equals sign (=) for an equality test is now deprecated.

A direct reference to an attribute of any data type can be be used as Boolean test for a (non-zero value):

`$MyString` is the same as testing `$MyString!=""`

`!$MyString` is the same as testing `$MyString==""`

`$MyNumber` is the same as testing `$MyNumber!=0`

`!$MyNumber` is the same as testing `$MyNumber==0`

`$MyDate` is the same as testing `$MyDate!=date("never")`

`!$MyDate` is the same as testing `$MyDate==date("never")`

Action, File, List, Set and URL data types conform to the String type test above as their data is just a particular form of string.

Logical Group Operators

Logical group operators examine a group of notes and determine whether every note in the group meets a criterion, or if any note does.

every(group,expression)

any(group,expression)

The designator group describes the notes to be examined. group may be any group designator {child,descendant,sibling,ancestor,all}. In addition, group may take a single argument that designates a particular group other than this; for example

```
collect(children(/agents/books),$Name)
```

The expression may be any valid expression, but is most often a reference to an attribute.

```
any(children,$overdue)
```

```
every(children,$status="important")
```

Dictionary Operators

A dictionary pairs **keys** with **values**. Dictionaries are faster to construct, and large dictionaries are *far* faster to check, than lookup tables.

A dictionary collects pairs of strings separated by a colon. The first string is the *key,* and the second string is the *value.*

```
$MyDictionary="cat:animal; dog:animal; rock: mineral";
```

The key “cat” has the value “animal”, while the key “rock” has the value “mineral”.

In some cases, it might not be clear whether a string should be treated as a dictionary; the enclose the dictionary in curly braces {...} `$MyDictionary={"cat":"animal"; "dog":"animal"; "rock": "mineral";}`

The **dictionary()** operator converts a string to a dictionary.

```
$MyDictionary=dictionary("cat:animal; dog; animal ; rock; mineral")
```

The value associated with any key may be retrieved from the dictionary:

```
$MyDictionary[key]
```

If this key was not present in the dictionary, the empty string is returned.

Values may be assigned to specific dictionary keys.

```
$MyDictionary[apple]="pie";
```

Adds the key "apple" to $MyDictionary with the value "pie".

Dictionaries may be merged by adding them.

```
$MyDictionary=$MyDictionary+"{apple:plant"};
```

adds they key “apple” and associates it with the value “plant”. If the key was already found in the dictionary, its value is replaced by the new value. If the key was not found in the dictionary, both the new key and the new value are added.

Operators

.at() returns the value corresponding to a dictionary key.

```
$MyDictionary.at("dog")
```

If the dictionary has no key "dog", **.at()** returns the empty string.

You may also lookup values by enclosing them in square brackets:

```
$MyDictionary["dog"]
```

.size returns the number of keys in the dictionary.

.empty is true if the dictionary has no keys, and is false otherwise.

contains(*key*) is true if the dictionary contains the designated key.

.icontains(*Key*) is true if the dictionary contains the designated key, using case-insensitive comparision.

.keys returns the set of all keys in the dictionary.

[Dictionary].add(key, value)

Sets the value of a given key. Equivalent to Dictionary[key]=value;

[Dictionary].extend({key:value})

Sets the value of a given key if it has not been set. If the key is already defined, appends the new value to a list of values. Use this operator to manipulate values that are lists or dictionaries.

Link Operators

Link Operators

links[(item)].direction.linkType.attribute builds a list from a collection of links.

item selects which note's links should be collected. It may be a single item, a list of items, or a group.If item is omitted, "this" is implied; if item is omitted the enclosing parentheses may be omitted as well. item may be a designator (e.g. "parent") or a note path (/config/details).

The argument direction is either "inbound" or "outbound".

linkType collects only links of a specified link type. linkType is a regular expression: wild-card characters are permitted and have special meanings. If linkType contains white space or periods, enclose it in quotes:

```
links.outbound."responds to".$Name
```

If 'linkType' is left empty, links of all types are collected with the exception of prototype links. Prototype links are always omitted. As a regular expression, linkType may match more than one link type:

```
links.outbound."agree|clarify".$Name
```

attribute is the name of the attribute whose values are to be collected in the result.

For example

```
$MySet=links(/config).outbound.supports.$Name
```

constructs a list of all the Names of notes that are linked to the top-level note named config via links of type "supports".

```
$MySet=links.inbound..$Name
```

collects a set of all the names of notes that are linked to this note.

Finding Notes With Links

`linkedTo(target [,linkType])` `linkedFrom(target [,linkType])`

Return true if the this note has a link to or from the designated target. If the optional linkType is specified, the link must share the linkType; otherwise, any link will return true.

`originalLinkedTo(target [,linkType])` `originalLinkedFrom(target [,linkType])`

Return true if the the original correwsponding to this note has a link to or from the designated target. If the optional linkType is specified, the link must

share the linkType; otherwise, any link will return true. These queries are especially useful in agents, where one often is interested in the links of the original note rather than any links to the note owned by the agent.

`linkPath(pathName) linkPath(pathName [, start [,end] ])`

These functions return a list of notes that are on a designated path. The pathName "*" designates any path, regardless of path name. If *start* list provided, that path starts at the designated note and ends on reaching the end note or when all links on the path reachable from start are exhausted. If only the pathName is provided, all notes on the path are listed.

If several paths exist, Tinderbox will return a path that is not longer than any other path.

Making Links

Five actions create or remove links between notes.

linkTo(target [,linkType])

unlinkTo(target [,linkType])

linkFrom(target [,linkType])

unlinkFrom(target [,linkType])

The first argument to each of these actions, is the name or path of a note, or a list of paths. The optional second argument is the link's type; if no argument is supplied, untitled links are used. Unlike links(), above, the linkType argument is not a regex and is a string literal.

linkTo and linkFrom will not create a new link if the link already exists. unlinkTo and unlinkFrom will do nothing if the described link does not exist. Thus, these actions may be used in rules and agents:

`Rule: if($Checked) {linkFrom(/agenda/today/tasks,"urgent!")}`

This rule will insure that an "urgent!" link runs from the note "tasks" in today's agenda; if the link already exists, the action has no effect.

If linkType is not supplied, unlink actions will remove links of all link types.

In agent actions, this is the alias of the note being added to the agent. Often, you want to add links to the original note, not the alias. The operators

linkToOriginal(target [,linkType])

unlinkToOriginal(target [,linkType])

linkFromOriginal(target [,linkType])

unlinkFromOriginal(target [,linkType])

operate identically, but if either the source or the destination are aliases, the link is created or removed from the original note.

createLink(source,destination[,linkType])

creates a link between any two notes. If a link of the designated type exists, no additional link is created.

eachLink(x){...}

eachLink(x,whichNote){...}

Examines each link, either inbound or outbound to this note, with the exception of prototype links. The local variable x is bound to a dictionary of link properties, including:

- source
- destination
- sourceIDString
- destIDString
- type
- class
- title
- target
- url
- comment
- anchor
- bold
- dashed
- dotted
- visible

The property **isFirst** is true for the first link in the enumeration and false otherwise. The property **isLast** is true for the final link in the enumeration and false otherwise. Changes to this variable values are not (yet) recorded as changes to the link.

The optional second argument designates which note's links are to be examined. If absent, eachLink examines the links to and from **this** note.

Inside the eachLink body, changing the values of most keys will change the link. Read-only keys are: anchor, target, title, and class.

Graph operations

Several operators return information about the ways that links connect notes.

neighbors(which,n)

Returns a list notes that can be reached by following exactly *n* links from the designated note or notes. Only the shortest path between two notes is considered. For example,

`neighbors(this,3)`

returns the set of notes that are connected by exactly three links to **this** note.

neighbors(which,n,linkType)

Returns the set of notes that are connected by exactly n links, considering only links of the specified type. Unnamed links are of type "*untitled."

neighborsWithin(which,n)

neighborsWithin(which,n,linkType)

These functions return the set of notes that are connected by n links or less to the designated note or notes. For example,

```
neighborsWithin(this,3)
```

Includes all notes connected to this note by no more than three links. The starting note or notes are considered to be connected to themselves.

Several operators return information about the ways that links connect notes.

neighbors2(which,n)

neighbors2(which,n,linkType)

neighbors2Within(which,n)

neighbors2Within(which,n,linkType)

These operators act as the corresponding operators above, but treat links as bidirectional. This, if A is linked to B, both A and B are considered to be connected to each other,

Attribute Operators

attribute()

The **attribute()** operator returns a dictionary of information about an attribute. For example,

`attribute("Width")`

returns information about the attribute `Width`. Use the **.at()** operator or [] to retrieve specific kinds of information:

```
attribute("Width").at("default")

attribute("Width")[suggested]

attribute("Width").at("category")

attribute("Width").at("type")

attribute("Width").at("description")
```

The description of the attribute as well as its default and suggested values may be set by actions:

```
attribute(attributeName)[suggested]="value 1; value 2";  attribute(attributeName)[default]="value 1";
```

Type

It is occasionally useful to be able to test the type of an attribute.

The operator type(attributeName) returns a string representing the type of the designated attribute. The attribute may simply be the attribute name (without quotes or a preceding "$") or an expression that, when evaluated, yields an attribute name.

```
type("Width") ➛ "number"

$MyString="Modified"; type($MyString) ➛ "date"
```

Note that type($MyString) returns the type of the attribute who name is stored in $MyString, while type(MyString) returns the type of the attribute MyString.

If no such attribute exists in the document, the operator returns the empty string.

Note Operators

inside

```
inside(targetNote)
```

return `true` if this note, or an alias of that note, is the child of targetNote. Also returns `true` if the targetNote is an adornment and this note overlaps that adornment in map view.

descendedFrom

```
descendedFrom(targetNote)
```

return `true` if this note, or one of its aliases, is a descendant of the targetNote.

Eval(), Action(), and Update

The eval() function evaluates an string and returns the result. For example,

```
eval("$Price*1.05")
```

returns the value of price augmented by 5%.

The action() function performs an action stored in a string.

```
action("$Total=$Price*1.05")
```

changes the value of $Total.

action() and eval() are most frequently used in export templates, but may occasionally prove useful when a note must assemble a rule on the fly.

update(notes) asks Tinderbox to update one or more notes by evaluating their rule and edict. **Notes** may be an individual or group designator, or a list of paths. If the note has been evaluated recently, Tinderbox will not evaluate it again. update() returns a list of updated notes.

create operators

create(path) or **create(container_path, name)** creates a new note at the designated path, and returns the path to the note that was created. If the note exists, no note is created. Any intermediate containers will be created as needed; for example, **create(/examples/basic**) will create the top-level container “examples” if that container does not already exist, and then will create the note “basic” inside that container.

createAgent(path) or **createAgent(container_path,name)** creates a new agent at the designated path. If a note exists at that path, no new agent is created

createAdornment(path) or **createAdornment(container_path,name)** creates a new adornment at the designated path. If a note exists at that path, no new adornment is created.

distance()

distance(note1,note2) computes the distance between two notes in the Tinderbox map view.

distanceTo()

distanceTo computes the approximate distance in kilometers between two notes for which $Latitude and $Longitude are known.

For example, if **this** note has the latitude and longitude of Boston and **Paris** has the latitude a longitude of Paris, then `distanceTo("Paris")` is about 5582.

document

document

The operator `document` return a dictionary of useful information about the document that contains **this** note.

`document[path]` returns the file system path to the document.

`document[url]` returns the file URL of the document.

`document[name]` returns the name of the document.

`document[user-attributes]` returns a list of user attributes available in this document.

`document[id]` returns a string that may be used to identify this document.

fetch

fetch(url,headers,attributeName,actions[,method])

fetch() retrieves information from a remote server, stores that information in an attribute or local variable, and then performs addition

- **url** is the url to be fetched.
- **headers** is a dictionary of HTTP(S) headers. This may be an empty dictionary "{}" if you don’t need custom headers.
- **attributeName** is the name of an attribute or local variable. The data from the remote server will be fetched asynchronously, and stored in this attribute or local variable. Note that this parameter is not evaluated.
- **actions** are one or more Tinderbox actions to be performed after the data is fetched, which may be some time after the the fetch operator returns. It may be an empty string.
- **method** is the HTTP method to be used in the request. If omitted, GET is assumed.

Note that the server might take several seconds or more before supplying a response. **fetch()** returns immediately without waiting for the server, and arranges for the **command** to be evaluated when the data arrive. It may be necessary for you to take steps to avoid repeating the same **fetch()** before the previous command has completed.

inheritsFrom

inheritsFrom()

The function **inheritsFrom()**, checks whether a note uses a specific note as a prototype, either directly or through other prototypes. For example, suppose Flower has the prototype Plant, and Rose uses the prototype Flower. Then

```
inheritsFrom(/Plant)
```

is `true` for both Flower and Plant.

```
inheritsFrom(/Flower)
```

is `true` for Rose, but `false` for Plant. You can also write

```
inheritsFrom(/Flower, prototype)
```

which returns `true` if the note which inherits from the note prototype.

hasLocalValue

hasLocalValue()

The operator **hasLocalValue()** lets you determine whether a note has a specific value for an attribute, or whether that value is inherited from a prototype or a default.

```
hasLocalValue("attributeName" [,target] )
```

Note that the attribute name should be enclosed in quotes and should not be prefixed with a $ sign.

```
hasLocalValue("MyString")
```

The arguments are evaluated, so

```
hasLocalValue($MyString)
```

Returns information about the attribute whose name is currently saved as the value of $MyString.

You will rarely if ever need to know whether a value is set locally or inherited. Wanting this information is often a sign that your overall design is incorrect!

isDuplicateName

isDuplicateName()

The operator **isDuplicateName()** is true if another note in the document has the same name as this note.

notify, speak

Notify

notify(headline[,subtitle][,date]) adds a notification to your computer's notification center. Headline and an optional subtitle are displayed in the headline. If the optional date is supplied, the notification will be held until after that date and time; otherwise, the notification will appear immediately.

Speak

.speak() will speak the value of any string value. For example:

```
$Text.speak()
```

will speak the note's text. An optional argument specifies the voice to use.

```
"Now is the winter of our discontent".speak("Tessa")
```

will pronounce the string using the South African voice Tessa.

locale()

locale()

The function locale() lets you change the locale used to translate dates. For example, Americans write 12/1/2006 to denote December 1, while in England the same date is written 1/12/2006.

The action

```
locale("en_GB")
```

sets the current locale to "British English". Locale codes begin with a two-letter language code, followed by and underscore and a two-letter region code. These are ISO standards ISO-639 and ISO-3166 respectively. Code combinations are available for any language supported by Mac OS X.

To return to the user's preferred locale, use

```
locale();
```

You may also save the old locale in an attribute for subsequent use. For example:

```
$OldLocale=locale("en_GB");
```

```
… do various things ...
```

```
local($OldLocale);
```

Note that changing the locale() can be fairly time-consuming, as lots of machinery must be torn down and rebuilt for each change.

require()

require("Hints")

The operator **require("Hints")** adds the /Hints container and its subsidiary containers. It is equivalent to selecting Built In Hints from the File menu. If any of the containers already exists, it is left unchanged.

The argument is currently ignored, but in the future values other than "Hints" may install other facilities.

runCommand()

runCommand

The **runCommand()** function asks the operating system to start a new process and results the result of that process.

```
$Text = runCommand(command_line, input)
```

runCommand() executes the specified command line in your default, passing it the the value input as its standard input. The standard output of the

command, if any, is the value of the completed action. For example:

```
$Text = runCommand(ls ~/Documents)
```

replaces the note's text with a list of the files and folders in the user's Documents folder, and

```
$Text |= runCommand("curl "+$URL)
```

will – if the note has no text – ask curl to fetch whatever text is found at the url stored in the note's $URL.

It is not necessary to use an attribute to hold the output from runCommand, allowing the operator to be used 'bare' in action code. If $CommandValue holds a valid command line string, this can be used in a rule or action:

```
runCommand($CommandValue)
```

select()

select([list]) - selects the notes designated in a list of paths. select() clears the selection.

stamp()

The stamp() operator applies a named stamp to one or more notes.

```
stamp(designator,stampName)
```

applies the stamp to the designated note or notes; for example

```
stamp(children,markAsComplete)
```

will mark each child of this note as complete. If the designator is omitted

```
stamp(markAsComplete)
```

the stamp is applied to this note.

If the stamp exists, the oeprator return true; otherwise, the operator returns false.

version()

version(["arg"])

The action operator **version()** returns information about the version of Tinderbox. With no argument, is return a complete version string such as "8.7.2b468". Optional arguments can be **major**, **minor**, **fix**, and **build**, returning a specific part of the version. For example, **version(major)** returns 8 in Tinderbox 8.8.0

Fetching Information From The Web

Tinderbox offers several ways to fetch information from a Web page.

- You can create a note whose text is the contents of a Web page you designate. Tinderbox can automatically fetch the contents of that Web page, so that the note is always updated with the current contents of that Web location.
- You can have notes that open automatically in a Web browser.

Attributes for Working With the Web

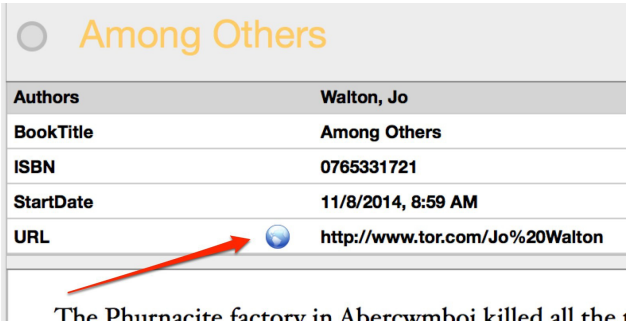
URL

$URL lets you associate a URL with a note. Other attributes use this URL to fetch data.
Examples:

```
http://www.eastgate.com
```

```
mailto:info@eastgate.com
```

If URL is a displayed attribute, a globe icon appears next to the URL. Press the globe to view the URL in your preferred Web browser.



AutoFetch

If $AutoFetch is true, the note will attempt to download text from the URL specified in $URL. If no URL is specified, or if the URL cannot be reached, or if Tinderbox cannot open an internet connection, AutoFetch has no effect.

AutoFetchCommand

The attribute $AutoFetchCommand extends AutoFetch's capabilities.
AutoFetchCommand is an action that runs when the Tinderbox is ready to fetch information from the Web – when the file is opened, and periodically when the file is open. Typically, AutoFetchCommand will invoke an outside program, perhaps fetching some information from the user's hard disk or network. For example:

```
$Text=runCommand("ls ~Documents")
```

will replace the text of the note with a list of all the files currently in the user's Documents folder.

```
$Delivered=runCommand ("myDatabaseQuery "+$TrackingID)
```

will run the shell script myDatabaseQuery, passing it the value of the note's TrackingID attribute as an argument.

ReadOnly

If $ReadOnly is true, you cannot modify the text in this note—you can only view it. It makes sense to set this true for notes that fetch their contents from the Web.

LastFetched

$LastFetched records the date and time when a note was post recently updated from the Web.

Import

Tinderbox provides many options for importing text from various sources.

Copy and Paste

To move moderate amounts of text from another application to Tinderbox, it may be easiest to copy and paste the text. This approach has the added benefit of giving you a chance to review your notes: arrange them, adjust their appearance, consolidate related notes, and prune notes that are out-of-date or irrelevant.

Drag Data Into Tinderbox Views

Drag and drop can be the quickest way to add an entire text file to your Tinderbox document.

You can drag and drop:

- text: drag a text selection into Tinderbox and—
    - drop it into a text window to insert the dragged text at that point in the text
    - drop it into a view window to create a new note.
- a text file: drag a text file from the Finder. Drop it into a view window to create a new note containing the dragged text.
- Finder clipping files
- DEVONthink Office Pro items
    - drop into a text window to insert the text or image clipping at that point in the text
    - drop into a view window to create a new note containing the text or image
- Bookmarks from web browsers, Bookends, or Yojimbo create Tinderbox notes; the URL of the note reflects the URL of the bookmark
- News items and subscriptions from news readers such as NetNewsWire
- OPML files
- Taskpaper files
- Microsoft Word® .doc files and Word XML .docx files
- pdf files
- tab-indented outlines
- vCards, dropped into Tinderbox from the Address Book or other vCard-enabled programs

Spreadsheets and Comma-Separated Value Files

Selecting a table in a spreadsheet and pasting it into Tinderbox will create a useful set of notes.

- The first row is treated as a set of headings, which map to attributes. New user attributes will be created for attributes that do not already exist.
- If a column is named Name, the name of the note is taken from that column. Otherwise, Tinderbox assumes that the first column contains the names of the notes.
- A new container will be created for the table's rows.
- Each row of the table becomes a note. The table's fields become displayed attributes, and these attributes are populated from the table

Note that Tinderbox is rather conservative in judging whether or not pasted text is actually a table. In particular, missing values in the rightmost column can lead Tinderbox to conclude that your data is not, in fact, formatted as a table. Filling in the empty cells will help Tinderbox import the information as you intend.

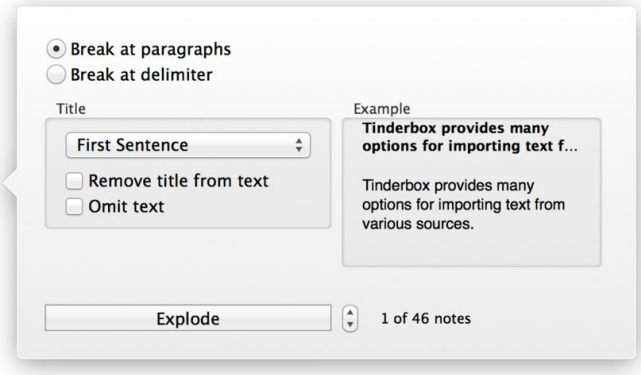
Explode

The Explode command divides a long text note into several smaller notes, splitting the text at locations you have chosen.

To explode a note:

- Select the note you wish to explode.
- Choose Explode... from the Note menu
- Decide whether to end each note after a certain number of characters or after a delimiter such as a period, then click the radio button next to your choice
- Change the number of characters or the delimiter if necessary. To remove the delimiter from the new exploded notes, check the Delete delimiter box.
- Click Explode.

The new notes will appear as children of the original note, in a new container called Exploded Text.



Delimiters you can use to separate the source content include:

- \r (new exploded notes will begin with each Return)
- \n (new exploded notes will begin with each new line)
- \t (new exploded notes will begin with each tab)
- , (new exploded notes will begin with each comma - useful for CSV data)
- any custom string, e.g. #### or XYXYXY.

Explode can also optionally delete the delimiter character(s), making it easier to use custom string that would look odd if left in the output.
It is also possible to control what part of the exploded text is used as a new note's title. Options are:

- first sentence
- first two sentences
- first paragraph

Watching Shared Notes

Tinderbox provides several options for automatically importing and updating notes created in other note-taking tools. This is especially useful for automatically gathering notes created on your iPhone or iPad.

The Notes Application

File ▸ Watch ▸ Folder From Notes… will create a top-level container that holds notes imported from a designated folder in the Notes application.

Tinderbox can now automatically import selected notes from the **Notes** application, which is installed in all macOS and iOS devices.

To connect Tinderbox to a Notes folder, create a top level container to hold the imported notes and set its $NotesFolder attribute to the name of the Notes folder you want to warch. Tinderbox will now automatically watch that container whenever your file is reopened, and periodically thereafter, importing new and modified notes to the corresponding Tinderbox container.

The imported notes inherit from a built-in prototype named “Imported From Notes,” allowing you to set common displayed attributes or visual appearance.

Changes to the imported notes will not be propagated to the application or to other devices.

The import process is comparatively time-consuming; it may be preferable to limit import to folders with no more than a few dozen notes.

Evernote

File ▸ Watch ▸ Notebook From Evernote… will create a top-level container that holds notes imported from a designated Evernote notebook.

Tinderbox can also automatically import selected notes from **Evernote**. Evernote import works much like import from Notes.

To connect Tinderbox to an Evernote notebook, create a top level container to hold the imported notes and set its **$EvernoteNotebook** attribute to the name of the notebook you want to watch. Tinderbox will now automatically import or update that container whenever your file is reopened, and periodically thereafter.

The imported notes inherit from a built-in prototype named “Imported From Evernote,” allowing you to set common displayed attributes or visual appearance. The $URL of each imported note provides access to the corresponding data in Evernote.

Changes to the imported notes will not be propagated to the application or to other devices.

The import process is comparatively time-consuming; it may be preferable to limit import to notebooks with no more than a few dozen notes.

Files and Folders

File ▸ Watch ▸ Folder from Finder… will create a top-level container that holds notes imported from a designated folder. These containers will watch the corresponding application periodically and will automatically add new notes and update notes which have changed. Changes made in Tinderbox are *not* forwarded to the application.

The watched file may be local to your Macintosh or may be shared via iCloud or Dropbox.

Tinderbox can now automatically import selected notes from the any Finder folder application, including folders in Dropbox or other remote servers.

To connect Tinderbox to a folder, create a top level container to hold the imported notes and set its $WatchFolder attribute to the name of the Notes folder you want to import. Tinderbox will now automatically import or update that container whenever your file is reopened, and periodically thereafter.

The imported notes inherit from a built-in prototype named “Imported From Finder,” allowing you to set common displayed attributes or visual appearance.

Changes to the imported notes will not be propagated to the application or to other devices.

Tot

File ▸ Watch ▸ Tot will create a top-level container that holds notes imported Tot™, a note-taking application from The IconFactory. Inside this container, Tinderbox imports notes corresponding to individual Tot panels. The list of panels to be watched is controlled by the value container’s $Tot attribute. The imported notes inherit from a built-in prototype named “Imported From Tot,” allowing you to set common displayed attributes or visual appearance.Changes made in Tinderbox are *not* forwarded to the application.

DEVONthink Pro

Tinderbox can now automatically import selected groups of records from **DEVONthink Pro**.

To connect Tinderbox to a DEVONthink group, create a top level container to hold the imported notes and set its **$DEVONthinkGroup** attribute to the unique DEVONthink ID of the group you want to watch, or the the DEVONthink URL of the group you want to watch. Tinderbox will now automatically import or update that container whenever your file is reopened, and periodically thereafter.

Groups inside groups are not currently imported.

The imported notes inherit from a built-in prototype named “Imported From DEVONthink,” allowing you to set common displayed attributes or visual appearance.

Changes to the imported notes will not be propagated to the application or to other devices.

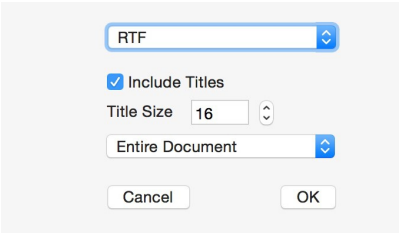
Export

Simple Export

Tinderbox provides a variety of simple formats that let you export your Tinderbox notes in one easy step. Tinderbox also provides a very flexible and powerful template-based export facility that lets you define precisely how you'd like your work to be exported; though Tinderbox calls this "HTML Export," it is equally suitable for working with XML, JSON, or even LaTeX.

The simple export formats appear in the Export submenu of the File menu.

- **Export as Outline** exports the title of each note to a single text file. Titles are indented with tab characters, providing a simple overview of your document.
- **Export as Text** exports the text, and optionally the title, of each note in your document to a single text file, RTF file, or Microsoft Word® .doc file. You may also choose to export to OPML, which many outliners can read, or to a Scrivener file.



- **Export as Attribute Browser,** available only when an attribute browser is the current view, exports a textual summary of your attribute browser results as an RTF file.

Text Substitution

When exporting text or RTF, Tinderbox interprets several useful markup elements.

^include(which)^ is replaced by the styled text of the designated note.

^value(expression)^ is replaced by the value of the expression. For example, ^value($Width)^ will be replaced with the value of the width of the exported note.

^if(expression)^ ... ^endIf^ and **&^if(expression)^ ... ^else^ ... ^endIf^** permit conditional export; the material within the curly braces will be exported only if the expression is `true`.

Exporting to HTML

Tinderbox provides exceptionally powerful and flexible export to HTML, XML, and many other formats. Every Web site and application has unique needs and requirements; Tinderbox makes it easy to pour information from your Tinderbox notes into files that meet your exact needs.

Indeed, Tinderbox templates can export data to all sorts of other programs and formats. Since HTML and XML are likely the most common formats in use today, our discussion will focus on these.

How Export Works

Tinderbox uses *templates* to understand how your notes should be formatted as Web pages.

The template is simply a note with text that looks like the information you want to export, with *placeholders* to indicate the location of information Tinderbox is to take from your document. The template is a form the Tinderbox fills out, in which placeholders to represent blanks and boxes that into which Tinderbox inserts information from your notes.

Many different notes can use the same template. Exporting them will create many Web pages, with different content—different titles, text, links, and so on—but in the same format on each Web page. Other notes may use a different format, and so may require a different template.

Templates

A template note is simply an a note in which special codes, or *placeholders* have been placed in the text to indicate where Tinderbox should insert information from each note. For example, `^title^` is replaced by the title of the note, and `^text^` is replaced by the text of the note.

Tinderbox defines a host of placeholders for various purposes, but the majority of these are seldom required very rarely. The principal placeholders are:

- ^title^
- ^text^
- ^value^
- ^include^
- ^children^

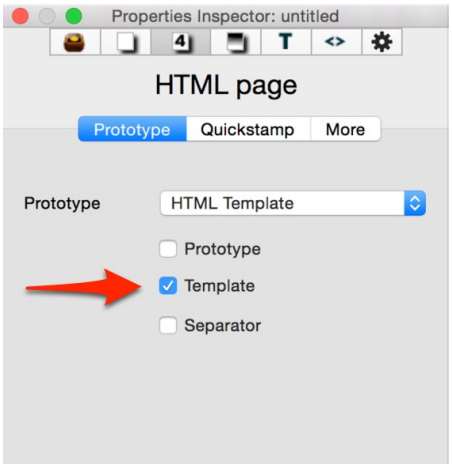
These will suffice for many users.

When you export a document, Tinderbox takes each note, puts information in the appropriate places in the appropriate template, and thus creates a new file for that note.

Making A Template Note

To tell Tinderbox that a note should serve as a template:

1. Select that note
2. Open the Properties Inspector and switch to the Prototype pane
3. Check the Template checkbox



A variety of built-in templates may be added to your document automatically; choose those you want from **File ▸ Built-In Templates**.

Exporting Information

What will transfer to the export

Text. The text in your notes translates to text on Web pages. Characters that must be translated or encoded in HTML, such as © and typographic quotes, will be translated appropriately.

Images. Images will be placed in the translated Web pages where they appear in the Tinderbox notes. Tinderbox will automatically translate the image files into fo either JPEG or PNG file.

Text links. Tinderbox links from portions of text will translate into links from that text on the web page to other exported notes or external web links.

What will transfer in part

Text appearance. Tinderbox's translation attempts to translate text styles, such as italics, boldface, and relative size, to similar text styles on most Web browsers.

The translation also attempts to recognize such meaningful items as headings and lists, and create appropriate HTML to display those items. Tinderbox has a simple lists feature that treats a paragraph beginning with an asterisk as an unordered list; use two or more asterisks to embed lists within lists.

Or, you can instruct Tinderbox to leave your text exactly as written—you will want to do this if you have incorporated Web formatting codes into your text already; do this using ^text(plain)^.

Lists. Tinderbox has a simple lists feature that treats a paragraph beginning with an asterisk or bullet (• ⌘⌥-8) as an unordered list. Use two or more asterisks to embed lists within lists.

Ordered lists may be created by beginning new paragraphs with the '#' symbol.

Basic links. If you choose, Tinderbox can add links to each Web page corresponding to the basic links from that note.

Hierarchy. If you choose, by adding appropriate export codes Tinderbox can create additional links from text on each Web page to provide access to the notes that are nearby in the Tinderbox hierarchy (ancestors, siblings, children, etc.).

Overlapping text links. In HTML, text links may not overlap. Text in a note that is the source of more than one link is linked on the Web to a special Web page listing the links that can be followed.

Additional Export Elements

The most common elements in export templates include

- ^title^
- ^text^
- ^value^
- ^include^
- ^children^

Other elements are occasionally useful.

^action(...)^ performs a Tinderbox action or a series of actions separated by semicolons. This element exports nothing, but can be useful for setting the locale or performing other actions before or during export.

^exists(...)^ returns **true** if the designated note exists, and false otherwise.

^firstSibling^, ^nextSibling^, ^previousSibling^, ^lastSibling^ return the relative URL of a sibling of this note. These can be useful for creating navigational links: **Next**.

^indent(n,c)^ exports *n* instances of the string c. If *c* is omitted, exports *n* tab characters.

^lowerCase(...)^ and **^upperCase(...)^** convert arbitrary strings to lowercase or uppercase letters.

^opmlencode(...)^ encodes any string, escaping characters that may not appear in OPML.

^randomChildOf^ exports one randomly-chosen child of a this note or, if an argument is supplied, one randomly-chosen child of the designated note.

^url^ exports the relative URL of the designated note, relative to **current,** the page being exported.

^urlencode(...)^ encodes any string, escaping characters that may not appear in URLs.

^version^ exports the current Tinderbox version.

Export Templates

Tinderbox placeholders begin with a caret (^). On typical English keyboards, carets are entered by typing Shift+6.

```
^title^
```

```
^value($Color)^
```

The placeholder continues from the opening caret and ends with white space, an optional closing caret, or a closing parenthesis. Although the closing caret is optional, it is best to include one unless familiar with exporting - the closing caret saves Tinderbox having to make assumptions about where a particular code ends.

> A placeholder ends with a closing parenthesis, white space, or a caret; the closing caret is only required if the placeholder directly abuts a printing character.

Most placeholders refer to the note that is currently being exported, but some placeholders can extract information from different notes.

^title^ : the name of this note

^title(this)^ : equivalent to ^title^

^title(parent)^ : the title of this note's container

^title(/tasks/132)^ : the title of the note "132" inside the container "tasks"

Placeholders that refer to other notes can refer to them in several ways:

- The name of the note
- The path to the note — especially useful if several notes might share the same name— such as /Configuration/PriceList
- A designator, such as "parent" or "nextSibling"

Paths

Export templates allow you to identify notes by relative or absolute paths. For example:

```
/news is the top-level note named "news"
```

```
/news/local is the note named "local", inside the top-level note "news"
```

The file name "." stands for this note, and ".." stands for the parent of this. Thus,

```
../Chicago is the note named "Chicago" that is a sibling of this note
```

```
../.. is the grandparent of this note
```

If a path begins with "/", the path begins at the top level of the document. If it does not begin with "/", it begins inside this.

Designators

The most common and useful designators include:

this - the note being exported immediately. If a note includes other notes, **this** is the included note while **current** (below) is the note that represents that is being exported.

current - the note that represents the page that Tinderbox is currently exporting. If a note includes other notes, **this** is the most immediate note, and **current** is the note that ultimately includes it.

next - the note that follows this note in outline order

previous - the note that precedes this note in outline order

prevSibling - the next older sibling of the this note

nextSibling - the next younger sibling of the this note

parent - the parent of this note

child - the first child of this note

lastChild - the last child of this note

Placeholders

To export the title of a note, we simply include the placeholder

```
^title^
```

```
^title(object)^
```

For example, we might want to show the title and the parent's title:

```
<h2> ^title(parent)^ : ^title(this)^ </h2>
```

Similarly, we can use ^text^ to convert a note's text into HTML and insert it into our page.

```
^text^
```

```
^text(note)^
```

```
^text(note,max_words)^
```

While the note's text and title are most likely to be of interest, we can export any attribute.

```
^value($attribute)^
```

gets the value of any attribute you specify, for this note.

```
^value($attribute(parent))^
```

gets the value of any attribute of this note's parent.
^value^ also evaluates expressions, permitting more complex forms:

```
Total Price: ^value($Price*$Quantity)^
```

Placeholders In Text

While export template codes are most often found in export templates, you may also type them directly into your note's text. For example, if the text reads

```
This note, ^value($Name)^, contains ^value($WordCount)^ words
```

the exported HTML for a particular note might read:

```
<p>This note, Objections To Operation TORCH, contains 731 words.</p>
```

We can also include information from other notes within the text.

```
^include(object)^
```

will format another note, using its own export template, and insert the results into our page. Alternatively, we might specify a special template

```
^include(object,summaryTemplate)^
```

Or, we can simply include some other note's text:

```
Please keep in mind the following warnings:

^text(/warnings/fragile)^

^text(/warnings/inflammable)^

^text(warnings/heady)^
```

^include^ and the related code ^children^ can help you to assemble complex pages and weblogs, pulling information from dozens or hundreds of notes into a single layout. These techniques are discussed in a chapter on Complex Pages.

Macros

Macros provide a convenient shorthand, helping you save typing and making your work easier to keep up-to-date.

```
^do(disclaimer, show, contact, phone)^
```

might expand into a lengthy, customized passage that might need to appear in many notes, and that would be tedious to retype.

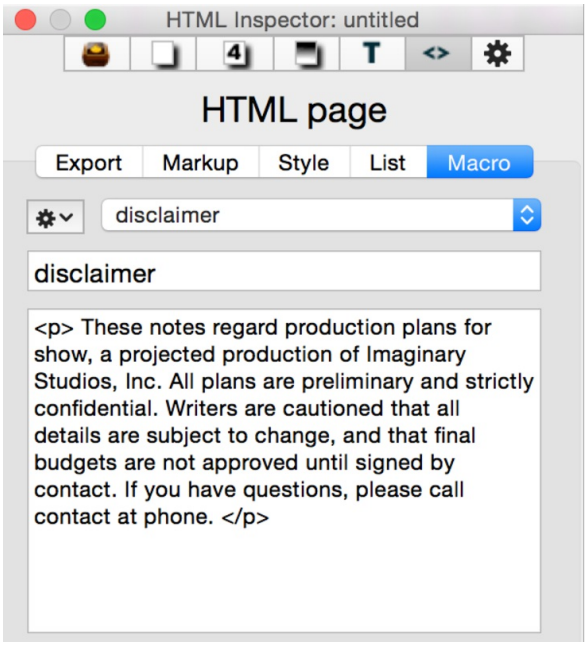
```
<h3>DISCLAIMER</h3>

<p>These notes regard production plans for show, a projected production of Imaginary Studios, Inc. All plans are preliminary and strictly confidential. Writers are cautioned that all details are subject to change, and that final budgets are not approved until signed by contact. If you have questions, please call contact at phone. </p>
```

Note that, if the legal department wants to change the wording of the disclaimer, we need only change the macro. If we had typed the disclaimer in several places, we would need to search and correct each occurrence. A macro is a text pattern that, when invoked, expands into a larger element.

```
^do(...)^
```

To define a macro, use the Macro pane of the Export Inspector (⌘-2).



For example:

> Name: dot
>
> Value: `<img src="anImage.gif" height="16" width="16">`

Defines a macro that will replace `^do(dot)^` with the specified 16-pixel image.

The ^do^ command may take additional arguments that are passed to the macro. Wherever the macro contains the substring `$1`, that substring will be replaced by the first additional argument. The substring `$2` will be replaced by the second additional argument.

If a paragraph consists entirely of a macro, paragraph markup is not applied. This is useful, for instance, as it makes it easy to use a macro to insert an element like an image (<img>) which requires no enclosing paragraph.

Group Designators

Most HTML markup elements concern an individual note, either this or a designated note. Some element consolidate information from a group of notes.

For example

> `^every(children,$Urgent)^`

is true if each child of this note is has a true value for its Urgent attribute.

Group designators include

- `children` - all immediate children of this note
- `descendants` - all notes descended from this note
- `sibling` - all siblings of this note
- `ancestor` - all notes from which this note is descended
- `all` - all notes in the document

Export Operators And Groups

`^every(group,attribute)^`

`^every(group,attribute,value)^`

`true` if every member of the group (such as every child) is true (if two arguments are supplied) or if every member of the group has a specific value for that argument (if three arguments are supplied). Always `true` if there are no examples of that group

`^any(group,attribute)^`

`^any(group,attribute,value)^`

`true` if any member of the group is true (two arguments) or if any member of the group has an attribute with the specific value (three arguments). Always `false` if there are no members of the group.

`^count(group)^`

counts the number of elements in the group

`^min(group,attribute[,precision])^`

the smallest value in the group.

`^max(group,attribute[,precision])^`

the largest element in the group

`^sum(group,attribute[,precision])^`

computes the sum of all elements in the group.

`^mean(group,attribute[,precision])^`

computes the arithmetic average of the group

Several group codes allow an optional argument, precision; if supplied, it indicates the number of decimal places desired in the result.

Note that most group codes can also be expressed using ^value^. For instance:

```
^value(any(children,$IsOverdue))^
```

Markdown

Some writers prefer to use styled text, others prefer 'Markdown'. Tinderbox works the way you prefer.

To use Markdown for a note, use the built-in prototype Markdown.

A note's $PreviewCommand determines how Markdown notes are processed. $PreviewCommand may be

- Markdown (to use classic Markdown)
- CommonMark (to use CommonMark)
- a path to any executable Markdown engine, such as pandoc, that you have installed on your computer.

If a note's $PreviewCommand is CommonMark, Tinderbox previews the text using the CommonMark Markdown standard (cheat sheet and documentation). CommonMark is significantly faster for large documents than the original Markdown script. CommonMark is now adopted by the built-in Markdown prototype in new documents, though the older Markdown.

To preview the formatted text, switch to the Preview pane by:

- Selecting Preview from the Text Pane Selector at the top of the text pane, or
- Choosing Preview In Text Pane (⌘⌥-E) from the Windows menu.

The appearance of formatted previews may be customized through a note in the built-in hints, /Hints/Preview/style. This note contains styles that are applied by the built-in Markdown template. This template can itself be modified by editing /Hints/Preview/template.

Marked2

Tinderbox automatically communicates with the streaming preview of the application Marked2 (https://marked2app.com). When you select or edit a Markdown note in Tinderbox, it will automatically be sent to the Marked2 streaming preview window.

AppleScript

Tinderbox now offers limited AppleScript support, making it easier to automate workflows with other applications.

Some sample expressions that Tinderbox supports include:

```
set myNote to make new note in document "Workspace.tbx"
```

```
set myAgent to make new agent in myNote
```

…creates a new top-level note, and creates an agent in that note.

```
set name of myNote to "inbox"
```

…sets the name of the note created in the preceding line.

```
set value of (attribute of myNote named "Width") to 5
```

```
get notes in myNote
```

…returns a list of all the notes inside myNote.

```
get agents in myNote
```

```
get adornments in myNote
```

…returns a list of the agents and adornments inside myNote.

```
set value of (attribute of myNote named "Width") to 5
```

…sets the value of an attribute, here $Width

```
get value of (attribute of lastChild of MyNote named "Width")
```

…fetches the value of the designated attribute. Note the parentheses in this and the preceding example. They're ugly, but AppleScript seems to compel this.

```
delete value of (attribute of myNote named "Width")
```

…removes any local value assigned to that attribute, restoring the inherited or default value.

You can get or set the default value of an attribute:

```
get defaultValue of (attribute of myNote named "Width")
```

```
set defaultValue of (attribute of myNote named "Width") to "5"
```

Or:

```
name of note 3 of document "Workspace.tbx"
```

Returns the name of the third top-level note in the specified document.

```
links of note 3 of document "Workspace.tbx"
```

returns a list of outbound links (excepting Web links) from the designated note.

```
find note in [note or document] with path "/path/to/note"
```

Returns a reference to the designated note. If the target is a document, the path should be an absolute path. If the target is a note, the path can be an absolute path or a relative path with respect to that note. For example: find note in myNote with path "parent" would return a reference to myNote's container.

```
move myNote to theContainer
```

Moves a note to the specified container. `delete myNote`

Deletes a note.

```
selected note of document "Workspace.tbx"
```

Returns the selected note. If several notes are selected, returns one of those notes, typically the first selected note. If no notes are selected, returns *missing value.*

```
selection of document "Workspace.tbx"
```

Returns a list of a selected notes

```
set selection of document "Workspace.tbx" to myNote
```

Selects a note. (At present, selecting more than one note is not supported.)

The expression

```
act on theNote with "action"
```

performs an action on the designated note. An action is typically one or more assignment or conditional statements, such as `Color="red"`. Act on does not return a value. The expression

```
evaluate theNote with "expression"
```

returns the result of evaluating an expression. In an action, = means "assign"; in an expression, = means "comparison" (although the unambiguous operator == is preferred.)

The expression

```
refresh theNote
```

informs Tinderbox that a note has changed, and requests updates in the user interface.

Link types may also be created and modified.

```
make new linkType with properties {name: "example" }

set a to linkType named "agree"

set the dotted of a to true
```

Note that scripts can do very bad things to a document; keep great backups.

Feathering Your Nest

Tinderbox is designed to adapt to your needs and preferences. This section points out a few of the many ways that you can customize Tinderbox.

Favorites

File > Open Favorites provides a menu of your favorite Tinderbox documents that will always be readily accessible. (File>Open Recent provides a menu of the Tinderbox documents you've opened most recently.)

A Tinderbox document is a favorite if its file, or an alias of its file, is stored in the Favorites Folder:

```
~/Library/Application Support/Tinderbox/favorites
```

You can view the support folder by choosing Reveal Support Folder in Finder from the Help menu.

User Badges

Tinderbox comes with a collection of built-in badges for a variety of uses. In addition, you can easily add your own badges.

Badges are .png image files stored in the Tinderbox support folder. A size of 32px x 32px is recommended; larger images will be scaled automatically.

To add a new collection of badges, simply add a folder of badges to the support folder. To add an individual badge, simple add it to the badge folder; it will automatically be added to the User badge collection.

The badge folder is located in your Library folder:

```
~Library/Application Support/Tinderbox/badges/
```

You can view the support folder by choosing **Reveal Support Folder in Finder** from the Help menu.

Character Badges

Badges may also be individual characters, such as emoji: 🅰🅱. If the value of $Badge does not correspond to a known badge and is precisely one character long, that character is drawn in place of the badge.

Badge Attributes

Badges are nominally 32×32 pixels in size. The attribute **$BadgeSize**, if not zero, requests that the badge be drawn at a larger or smaller scale. Not all views permit oversize badges.

The attribute **$BadgeMonochrome** is useful for badges that are normally black, allowing better display against dark backgrounds.

User Fills

Tinderbox comes with several pattern files or *fills* that can add texture to notes. You can add your own fills, too.

Fills are simply image files stored in the Tinderbox support folder

```
~Library/Application Support/Tinderbox/fill
```

You can view the support folder by choosing **Reveal Support Folder in Finder** from the Help menu.

A Distinct Look For Each Project

Many people use several different Tinderbox projects daily. Perhaps one tracks business expenses, another contains notes on a long-term research project, and a third lists books you'd like to read and films you'd like to watch. By using smaller, focused documents, you can more easily adapt each Tinderbox to the specific needs of its core tasks. In addition, smaller documents load more quickly.

When you use different documents frequently, it's handy to make sure you can tell at a glance which project you're using. Some good ways to do this include:

- Change the default value of $MapBackgroundColor
- Change the default value of $MapBackgroundFill
- Change the default value of $TextBackgroundColor
- Use a distinctive and attractive $NameFont

Color Schemes

The default Tinderbox color scheme is simple and straightforward, but many alternative schemes are available.

To change the color scheme, simply choose **Document Settings** from the **Edit** menu and select the Colors pane. Each scheme in the list offers a description and custom color swatches; most also include custom font settings and other appearance modifications.

Release Notes

9.6.0

HIGHLIGHTS

Hyperbolic View

The hyperbolic view has been revised to improve its layout and to offer you more control over what the view will include.

A list of link types lets you select links to include in the *spanning tree* that determines the contents of the view, and also which link types to show as crosslinks. Dragging a link type higher in the list increases its priority; high-priority links will be included in the spanning tree in preference to lower-priority links.

When creating a link, the disclosure control at the left edge of the control panel now allows you to expand the control panel, making more of the link list visible at one time.

A new force-directed layout technique improves the legibility of the view even more.

Drag a link from any note to the background to create new links. This technique can be excellent for brainstorming.

Links Between Documents

Links Between Documents

Tinderbox now lets you make a link between documents.

To make a link between two documents, open both documents. Drag a link from the source to the source window's main parking space. Then, select the window for the destination document, and drag a link from that window's main parking space to the destination.

When Tinderbox needs to refer to the destination, it will now reopen the destination document automatically if that document is not already open.

How Tinderbox Resolves Links Between Documents

Each Tinderbox document has a unique ID. Tinderbox keeps an internal list of documents that it has seen, indexed by this unique ID. As a result, Tinderbox can find documents even if they have been renamed or moved — provided that they have been opened in Tinderbox since they have moved.

A document receives a new ID if (a) it has been opened as Untitled, e.g. from the Favorites Folder, or (b) it has been **Saved As…** a new document. Note that duplicating a file in Finder does not changed the ID.

Change to Parking Spaces

When a link is dragged to any *regular* (left-hand) parking space, *all other regular parking spaces* also receive that note.

The behavior of *text-pane* parking spaces has not changed.

Deleting Links Between Documents

Links between documents may be deleted in the usual way from the Browse Links popover.

If the note that is the destination of the link is deleted, Tinderbox is free to delete the link the next time the source of the document is saved. If the destination document itself cannot be found, Tinderbox may retain the link, since that document might be restored from a backup.

Posters

This release now supports **posters**, an exciting new extension to map view. See the separate demonstration Tinderbox for details on posters.

NOTABLE

Actions

changed()

A new action, **changed([target]),** notifies a note that it needs to redraw itself.

Fetch

A new operator lets you fetch data from a server with greater flexibility than AutoFetch affords.

fetch(url,headers,attributeName,command[,method])

url is the url to be fetched.

headers is a dictionary of HTTP(S) headers. This may be an empty dictionary "{}" if you don't need custom headers.

attributeName is the name of an attribute or local variable. The data from the remote server will be fetched asynchronously, and stored in this attribute or local variable. Note that this parameter is not evaluated.

command is a Tinderbox action to be performed after the data is fetched, which may be some time after the the fetch operator returns. It may be an empty string.

method is the https method to be used. If omitted, GET is assumed.

Note that the server might take several seconds or more before supplying a response. **fetch()** returns immediately without waiting for the server, and arranges for the **command** to be evaluated after the data arrive. It may be necessary for you to take steps to avoid repeating the same **fetch()** before th previous command has completed.

jsonEach

In [String].json.each(){...}. an optional path argument supplies a path to the array to be iterated. For example, if $Text is

```
{
"person": { "firstName": "Thomas", lastName: "Roe"}, "coordinates" : [-90,41]
}
```

then **$Text.json.each(coordinates){...}** would iterate through the array of coordinates.

If json.each(){} is the first item in an expression, it reuses the current JSON object. This can be much faster than repreatedly reparsing a complex json package.

The older syntax $Text.json[coordinates].each(x){....} is no longer supported.

Select

The operators **select** and **.select** select designated notes.

```
select("/A;/B");
"/A;/B".select();
```

To deselect all notes, use **select()** with no argument.

The operator **require("Hints")** adds the /Hints container and its subsidiary containers. It is equivalent to selecting **Built In Hints** from the **File** menu. If any of the containers already exists, it is left unchanged.

Comments begin with // and continue to the end of the line. Formerly, comments terminated with a second //, but URLs are not unheard of in comments and this caused confusion.

The expression **%matches** represents a list that contains all available regular expression matches. This is equivalent to the list $0;$1;$2….$N, where N is the count of available regular expression matches.

Attribute Browser

In the attribute browser, selected notes may now be copied — whether the selection is an individual note or more than one note.

Built-In Prototypes and Hints

The Built-In Prototypes and Hints containers now respect $CleanupAction.

$Participants has been added to the displayed attributes of the built-in prototype Event.

Built-in Hints overrode changes to highlighters and taggers that had been installed. This should no longer occur.

Displayed Attributes

Multi-line attributes are now permitted for actions, sets and lists as well as strings.

Edicts

Tinderbox once more performs the edict of every note — not only the selected notes — when a document is loaded.

Email

A new attribute type, **email**, is now definite. $Email now has this type.

Attributes with type **email** appear in the Displayed Attributes table with a button that allows you to compose a custom email to the recipient(s). The body of the email is taken from the text of the message. If the note has an $EmailTemplate, the body of the email is composed by exporting the note using this template. **$EmailSubject**, a new string attribute, allows you to set the Subject: of emails composed from the DisplayedAttributes table.

When using the email button of the Displayed Attributes table, an email is composed to be sent to the recipients contained in $Email. The body of the email is taken from the $Text of the message. Alternatively, if $EmailTemplate specifies a path to a template note, that template is applied to **this** note and the result of that export is used as the message body.

Export

For convenience in export, several designators skip notes that are not exported, or usually considered outside the document outline. These include:

- adornments
- separators
- other notes which do not export

These include next, previous, nextSibling, previousSibling.

To designate a note that might be an adornment, separator, or unexported item, use the corresponding item designators: nextItem, previousItem, nextSiblingItem, previousSiblingItem.

Get Info

Get Info:Maps gain an **Open In Maps** button that opens the Maps application.

Get Info: Book once again displays covers for many (though, of course, not all) books.

Hyperbolic View

Hyperbolic view now defaults to setting *untitled links for tree-building.

Hyperbolic View no longer offers a crosslinks checkbox, as we now have fine-grain control of spanning tree construction and cross links.

Hyperbolic view omits “Delete” from the contextual menu of the focus note, since deleting the focus would leave nothing in the view.

In hyperbolic view, creating a link with a new type now refreshes the list of link types.

Hyperbolic view now displays notes within 8 links of the focus note. This keeps the display within reasonable bounds and avoids numerical instability as we approach the edge of the display.

The “spacing” control in Hyperbolic View has been replaced with a refresh button, which resets the display and restarts force-directed layout.

When force-directed layout is running, moving the center of the display will suspend force-directed layout for a second. This avoid problems with misplacement of very distant notes, because a small misplacement of a distant note can have very large effects.

Panning the hyperbolic view is now restricted so that part of the graph always remains in view. Previously, it was possible to move the entire graph to the far distance, leaving no hint of where to find it.

Import

DEVONthink watch folders should again work as expected.

When dragging an item from DEVONthink, we now set $URL to the item URL if it has once. For example, if the item is a web page, $URL is set to the URL of the page. If the item doesn’t have a URL, we continue to set $URL to the DEVONthink internal URL to allow prompt reference to the item.

Imported calendar events are assigned the prototype Event if that prototype exists, but only if the newly-created note wasn't already assigned a prototype by the OnAdd action.

Ventura introduced a new drag flavor for calendar items. Tinderbox didn’t get the memo. (Was there a memo?)

Inspector

The primary role of the Text pane of the Name Inspector is to change the default text style of new notes. In addition, using this pane to change the text color, line spacing, or paragraph spacing immediately changes the text of currently-selected notes.

Links

When creating links, the swap button now records any pending changes to the destination name.

The Links Inspector now reports the number of links that use the currently-selected link type.

When a partial link is placed into the normal (left-hand) link parking space of any window, the normal (left-hand) parking spaces of all currently-open windows also received that link. The text parking space (right-hand) is not affected.

Outline View

When double-clicking a note in the Links pane to select it, Tinderbox will expand any collapsed ancestors of that note in outline view.

Map View

When dragging multiple notes into an *empty* container in outline or other views, the notes’ relative positions are now retained. Notes will continue to be repositioned when dragged into containers that already contain notes, to avoid collisions.

Sorting

A new sort transform, **library sorting**, permits notes to be sorted as they might be filed in a library. By default, library sorting ignore initial words “a”, “an”, and “the”. The note **/Hints/library sorting/** , if present and not empty, contains a list of words that will be ignored for library sorting. You can customize this for different languages and filing practices. Library Sorting also respects some locale-specific rules for handling diphthongs and diacritics, and is always case insensitive.

Stamps

In building the Stamps menu, separator notes now become separator menu items (if not named) or disabled menu items (if the separator has a name).

Tabs

The tab contextual menu command for Hyperbolic view now sets the focus to the selected note.

Text

If **Format ▸ Font ▸ Code Font** is applied to a passage that already uses the code font, the font is changed to the note’s text font.

Format ▸ Font ▸ Code gains a shortcut: ^⌥⌘-C

In the Links pane, right-click on any item in the Inbound or Outbound Links lists in the Links pane for a contextual menu, allowing you to select that note.

Format ▸ Style ▸ Standard Font and **Format ▸ Style ▸ StandardSize** can now be applied to a multiple selection.

Format ▸ Style ▸ Reset Margins is available from the view pane.

In the Links pane, the listx of link destinations now have a contextual menu that lets you select an item. This selection is undoable, allowing you to return to the note you were previously viewing.

DETAILS

Actions

Tinderbox no longer forbids expressions that interrogate properties of the root note — the parent of top-level notes. For example, a top-level note can now get the value of $MapBackgroundColor(parent) if it wants to know the background color of the top-level map.

action()

Revised parsing of **action( )**, which sometimes saw syntax errors in valid expressions.

attribute()

The attribute() operator can now read and set the **lines** property of string attributes.

The key “description” may now be used to set the description of an attribute. For example, **attribute('Price')['description']="The price of the item."**

comments

A comment followed by a var statement caused some statements following the var statement to be treated as comments.

eachLink

The eachLink(){} action crashed if the action deleted the targeted link. It is not safe to delete links inside the eachLink(){} operator.

The eachLink(){} operator now examines each link in the sequence used by **Browse Links…**, examining text links that appear earlier in the text before those that appear later. Previously, eachLink examined links on order of creation.

When modifying the url of web links in **eachLink(x){…},** Tinderbox mistook the colon in the scheme delimiter as a signal of a dictionary key.

The **eachLink(x){…}** operator exposes additional properties of the link, and allows those properties to be changed. New properties include **visible, dashed, dotted, bold, broad, linear**,, **sourceIDString,** and **destIDString**. In addition, the **url** property may be modified if the link is already a web link.

Assignment

Assignment to a list of notes now recognizes bracket-enclosed lists correctly:

```
$Status([this;parent])="urgent";
```

Designators

In actions, offset references in which the inner values is an expression are now supported. For example, you can now write

```
$Name(nextSibling($MyDictionary[‘magicKey’]))
```

Dictionaries and Lists

The new {...} and [...] syntax for dictionaries and lists allows complex structures, such as lists of lists and lists of dictionaries.

Indexed references to a list of lists, e.g.

```
var:list myMatrix = "[0;1;2];[3;4;5][6;7;8][9;10;11]";

$MyList=[[0;1;2];[3;4;5][6;7;8][9;10;11]]
```

failed because the parser stripped brackets incorrectly. The parser has been revised to handle nested lists more consistently.

collect

collect(*where,what*) and its relatives failed when *where* was a **find(…)** expression

date()

Date conversion of integers has been modified to better handle prehistoric dates. When integers are converted to dates, the conversion is as follows:

- -35000 to +2500: refers to a year between 35,000 BCE and 2500 CE* otherwise: refers to the number of seconds before or after midnight UTC on 1 Jan 1970

.format

.format( ) now uses “duck typing” to assess the type of the object being formatted if the type cannot be determined. For example, if $MyList is [[1;2] ; [3;4]], then **$MyList[0].format(',')** inspects the result, "[1;2]", decides that it looks like a list, and formats it as a list.

inside()

The operators **inside(*path*)** and **descendedFrom(*path*)** did not recognize designators

Attributes

A new attribute, **HoverBackgroundColor**, controls the color of the hover view.

$Separator is renamed **$IsSeparator**. The old and now-deprecated attribute continues to function as a secondary name of the new attribute.

links

links.[inbound/outbound].path.attributeName expressions failed if the attribute was a URL attribute, because the system was misinterpreting the scheme (such as https:) as a dictionary key.

unlinkTo/unlinkFrom

The unlinkTo and unlinkFrom operators no longer delete prototype links.

AppleScript

The command

```
make new linkType with properties {name:"tester6", ...}
```

Now recognizes the property “broad” for broad links.

Displayed Attributes

In the displayed attributes table, the action of the values pulldown menu has been improved when editing sets and lists. Now, any pending edits are recorded, and then the chosen value is added to the set or list. Previously, pending edits were discarded.

When dragging a URL into a Tinderbox container, the value of $DisplayedAttributes is now set to “URL” only if the OnAdd action has not already assigned a value.

The Displayed Attributes table should no longer terminate the editing session of a value when a rule or agent action changes that value.

After selecting from the pulldown value menu in the displayed attributes table, the currently-selected attribute remains selected for list and set attributes, since you might well want to add or remove several elements. For other attributes, the next row of the displayed attributes table is selected.

Export

When a document is loaded, Tinderbox checks all notes that are used as (a) an HTMLExportTemplate, (b) an EmailTemplate, or (c) a PosterTemplate. If a note is in use as a template, it is marked so $IsTemplate is true. Notes that are not currently in use as templates are *not* marked with $IsTemplate as false; a note my advertise its willingness to serve as a template even if it is not in use.

The disused attribute $TextExportTemplate is now deprecated.

Infrastructure

Prevented a crash observed when Tinderbox to attempt to use the index of a note that doesn’t exist.

Internally checking whether a note has text is now closer to being thread-safe.

TbxDocument’s **close** method could get into trouble if the agent queue was very busy — especially if pending actions would add more work (such as screen updates) to the agent queue. We now clear pending work, since this will either not need to be performed (for screen updates) or will be performed on reopen (for agent updates) before putting the action recycle operation onto the queue. This should reduce problems on close.

Revised handling of the internal indexing primitives. These are especially tricky because they must run in the background, and must be careful to sequence themselves relative to each other and to changes performed by the user and by actions. This issue is believed to have been responsible for crashes sometimes encountered when bringing a Tinderbox window to the foreground after using another application.

Inspector

Corrected the width of the Badge control of the appearance inspector.

%matches is now treated by TbxCodeField as a keyword.

The Hover inspector pane has been modified to accommodate HoverBackgroundColor.

Syntax highlighting now recognizes keywords such as **function** when they appear at the start of a note.

Syntax highlighting of quoted strings now includes the closing quote.

When a new link type is created in the Document Inspector’s Links pane, it takes its color from the current color scheme. Previously, the new link type used black links.

The Inspector’s word count failed to update in large documents, and failed to adapt when changing documents. We now throttle updates to this instead, because counting hundreds of thousands of words can take time. The reported value may lag the actual value by several seconds.

Map View

Corrected the text area computation for notes that use the shape “left tag” to have the same size, and therefore the same line breaks, as the shape “right tag.”

Text thumbnails are now drawn in **rounded** notes as well as **rectangular** notes.

An outmoded animation in the **Cut** handler for map view sometimes left “ghost” views, as animations for removing and selecting the view could wind up interacting.

In map view, the inbound and outbound stub counts are once again positioned correctly.

Outline View

In outline view, the icon symbol no longer shows an outbound link if the note has only Web links.

Paste

When a note is pasted into a document, its $Created and $Modified dates retain their values. Previously, both $Created and $Modified were set to the date on which the note was pasted.

Print

Printing the text pane when multiple notes are selected is temporarily disabled.

Roadmap

Less space is reserved for link comments in the torn-off Roadmap window. When the selected link comment is empty or brief, or when no link is selected, even less space is reserved.

Text

Addressed an error in accounting for text link changes when editing a multiple selection in which text links appear in the second or subsequent notes.

The minimum height of newly-created text windows is now increased in proportion to the number of displayed attributes.

Adjusted the placement of the error icon in the text pane. It now aligns with the top of the title.

Suggested links should now exclude notes in the Hints container.

Pasting from Microsoft Word® into the text pane set $URL to an applewebdata: URL. Because these URLs aren't useful, we no longer record them.

Treemap View

Treemaps now respond immediately to selection changes.

Ziplinks

When Tinderbox converts a smart link in the text to a Tinderbox web link, it now assumes that the appropriate scheme is https if no scheme was specified.

9.5.2

NOTABLE

Actions

The dot operators **.round()**, **.floor()**, and **.ceil()** are now recognized.

The operator **rand(N),** where N is an integer, returns a random integer between 0 and N-1. As before, **rand( )** returns a random number between 0 and 1.

[List/Set].randomItem() returns a randomly-selected item from a list or set.

A new operator, [String].wordCount, returns the number of words in a string. The string is first scanned to determine its dominant language, and the word count is based on the conventions of that language. If the language cannot be determined or if it is not known to macOS, the conventions of English are used.

Colors

Added a new color scheme, London.

Displayed Attributes

String attributes may be set to display multiple lines in the Displayed Attributes Table. In the Document Inspector, open either the System or User attributes pane and select the string attribute of interest. A slider allows you to display from 1 to 7 lines.

Inspector

Tinderbox code fields, used to edit actions and queries, now record changes automatically and no longer require you to press [Return]. Code fields accept [Return] and [Tab] with their customary meanings.

Flags

Flags now permit several new expressions, useful for progress bars.

bar(value[,min,max])

Draws a horizontal progress bar, using the note's $Color and $AccentColor values. If the minimum and maximum values are not specified, they are assumed to be 0 and 100.

vbar(value[,min,max])

Draws the bar vertically.

An optional target line may be drawn: for example **bar(33,0,100,65)** draws the target line at 65%.

The first flag in $Flags is now drawn in outline view, in the place formerly reserved for the color swatch.

Import

File with extension .md, .mmd, and .markdown that are dragged into a Tinderbox view are now converted to styled text.

When dragging a Markdown file into Tinderbox, hold down the Option key ⌥ to import it as plain text.

Watching Tot works again, after an error in interpreting markdown caused trouble.

Preview

When viewing the Preview of a note, Tinderbox was (understandably) confused if the export path of two or more notes were identical. For example, consider two different notes each export their contents to the file

    /path/to/Note.html

Of course, in a normal export, one exported note will overwrite the file exported by the other. Confusingly, however, this meant that if you selected a note while Preview was active, the selection would automatically switch to the first identically-named sibling. Instead, Tinderbox now allows you to select the intended note.

Tinderbox sometimes truncated previews because it incorrectly estimated the length of UTF8-encoded strings passed to the preview pane.

Views

A new item in the item contextual menu lets you **Copy Note Path**, placing the path of the selected note on the clipboard.

User Interface

Two new operators, **show(msg)** and **[String].show**, let actions display a transient message. This can be especially useful when constructing complex actions. When **show** is called, a small placard slide up from the bottom of the main window, in which the message is displayed. The placard remains in view for five seconds, and is then automatically hidden. If a second message is shown while the placard is visible, the placard remains visible for an additional five seconds, and the new message replaces the old message.

The **show()** and **.show()** commands now accepts optional colors, **.show([background [, text color]]).** This can be useful, for example, in unit testing, where failures are shown in red while notification of success is in green.

If **show** or **.show** are called while a message is already being displayed, the new message will be shown after the previous message is hidden.

Details

Actions

Resolved a problem in parsing comments that included quotation marks, apostrophes, semicolons, or braces.

Resolved a hang that could arise when accessing the $Path of a note in different queues.

collect(…) and related operators can again use find( ) as a designator.

The **values(attributeName)** operator failed if **attributeName** was an expression that requires evaluation, such as a local variable. If **attributeName** is not an attribute, Tinderbox now attempts to evaluate it. Note also that **values($MyString)** returns a list of values found for $MyString, *not* a list of the values of the attribute whose name is stored in $MyString. This is unique to **values()**, and arises because writing values($MyString) instead of values("MyString") was a very common mistake.

.captureWord and .captureToken crashed if the captured element terminated at the end of the string.

The countIf() operator sometimes reported a parsing error when no error was in fact observed.

The result of [String]**.size** is now the number of unicode characters in the string. Previously, it was the size of the UTF-8 string in bytes, which differed in strings containing multibyte characters.

Avoided a crash when the currently-selected note has its link deleted by an action. The link animator must be notified in advance, as it is left holding a reference to animate the deleted link.

The operator sum_if() and its kin now bind **that** as expected. For example,

    sum_if(children,$Price>$Threshold(that),$Price)

Adds up the prices of each child of this note, provided that its price exceeds a threshold stored in the parent note.

Assigning the $SiblingOrder of a note sometimes left the note with a sibling order one greater than the intended position.

The **.extend()** operator now accepts quoted strings. The following expressions are equivalent:

    $MyDictionary.extend({1:able}}

    $MyDictionary.extend("{1:able}"}

Attribute Browser

Corrected the layout of the Attribute Browser's column picker.

A potential hang occurred in attribute browser when the attribute browser's query was evaluated during an agent update.

In Attribute Browser, group headers were not sufficiently distinguished from normal rows. This was a temporary fix for a longstanding cosmetic glitch which cause the highlighting of group rows to fail intermittently and unpredictably. The underlying cause is that the NSTableView assumes that it is free to set the size and font of the textView field. We not longer use the textView field at all, and so the NSTableView no longer interferes.

Brainstorming

Parsing tags and prototypes for brainstorming is smarter about a variety of special cases. Notably, if the first character after a # is a digit, we assume it's an expression like "Activity #3" and do not create a prototype. Also, email addresses in titles (mark@example.com) are not treated as introducing a place.

Built-In Hints

Adding Built-In Hints, either directly or by using the Preview pane, deleted existing stamps.

Explode

Explode was confused if the explode action moved notes to a new container. We now defer performing the action explode has created all its notes, and then perform the action on each note in turn.

Export

When referring to a template by name rather than full path, Tinderbox now uses the first template note of that name, searching in outline order. Previously, Tinderbox selected any note of that name, without checking that the note was intended as a template. This caused unexpected conflicts when template names like “Preview” were used elsewhere in a large document.

Inspector

Word counts now take into account the dominant language of the note, and are smarter about recognizing what is and is now a word.

Recursive word counts — used to measure the word count of a note and its descendants — are now significantly faster.

When you are using the search field in the System Attribute Inspector, switching to another application and then returning to Tinderbox no longer resets the selected attribute. Instead, Tinderbox selects the attribute for which you were searching.

The Agents and Rules pane of the Tinderbox Inspector now displays the name of the note that is being tagged.

Text

When the parking space control was used with the keyboard, it could display two superimposed copies of the link confirmation popover. This made keyboard cancellation problematic.

The key attribute picker, the key attribute table, and Get Info: Attributes now display deprecated attributes as struck through.

Map View

The link info widget is again drawn correctly; a workaround for a system font metric issue is no longer needed.

Improved the dark-mode drawing of the composite name widget, and of monochrome badges against dark backgrounds.

Prototypes

When a note’s prototype is set or changed, the note copies the prototype’s text if the note has none, or if the text it inherited from its former prototype has not been changed. If the prototype has text links, the note also acquires text links.

Stamps

Resolved a potential crash because the stamps menu could be rebuilt while stamps were being used by actions.

Text

Tinderbox text highlighting after Find now respects case-sensitive search.

Corrected a crash when using only a prototype or a place in a ziplink. E.g. [[@smith]] or [[#Person]].

Pasting text from Microsoft Word® no longer places a useless applewebdata:// URL in the $URL attribute.

If a calendar event dragged or pasted into a Tinderbox view has invitees, the names of the invited people are listed in a new Event attribute, $Participants.

9.5.0

HIGHLIGHTS

Brainstorming

Tinderbox introduces new notation for fast and fluid brainstorming. When naming a note, you may now use the “#” symbol to set the note’s prototype. For example, creating “Tinderbox#software” makes a note named “Tinderbox” with the prototype “software”. If the prototype does not alreadt exist, it will also be created automatically.

When naming a note, you may use the “@” symbol to associate the note with a location. For example, “Brie@cheese store” will create a note named “Brie” which is linked to the note “cheese store” by a link with type “place”. If “cheese store” exists and if it has an address, its geographic information will be copied to the new note. If “cheese store” does not exist, it will be created.

The same notation may be used in ziplinks. For example, [[Tinderbox#software@Eastgate]] will link to the note named Tinderbox, creating it if it does not yet exist, and will also link Tinderbox to the place “Eastgate”, creating that place if it does not yet exist.

Export and Preview

Preview is now much simpler to use. If you don’t already have a template, Tinderbox will create a simple template in /Hints/Templates/preview. You can edit this template if you want to customize Preview, but you don’t need to. Similarly, if you have not chosen an export folder, Tinderbox makes a temporary folder for the preview, and cleans up after you’re finished.

Images are now automatically exported when a note with images is previewed.

Images

Tinderbox is now much more efficient in storing images in text, as well as image adornments. You may use images more freely than in the past.

Map View

Geographic adornments are back. If an adornment has an $Address or if it has $Latitude/$Longitude information, Tinderbox displays a map on the face of the adornment. The extent of the map displayed is determined roughly by the attribute $Range, in kilometers; if not specified, the $Range is approximately 1km.

A new shape, “rounded”, provides a rectangle with gently rounded corners.

Text

If several notes are selected at once, you may now view and edit all the selected notes in the text pane. Editing is permitted inside any note, but not in selections that contain more than one note. Nonetheless, the text format may be modified across note boundaries, facilitating changes in font or type size across an extended document.

NOTABLE

Actions

The .speak() operator now works properly when called on a literal string: "hello".speak()

A new operator, while(test){...} performs an action repeatedly until the expression test is false. For example:

```
while($Text) {var:string s=$Text.sentence; $Text=$Text.substr(size).trim(); process(s);}
```

will call the function `process()` in turn on each sentence of the text, until the text is empty. In order to reduce the incidence of infinite loops, while loops are limited to 10,000 iterations.

Lists may now be written by enclosing them in square brackets. Lists may be nested; for example, the list

```
[ 1; [2;3]; 4]
```

contains a 3 elements — 1, the list [2;3], and 4.

Long Tinderbox precedent holds that list addition adds each element of the two lists. For example

```
$MyList = [1] + [2;3] ➛ [1;2;3]
```

To add a sublist to a list, use the operator .extend().

```
$MyList = [1].extend( [2;3]) ➛ [1;[2;3]]
```

Dictionaries may now be written by enclosing them in braces. Dictionaries may be nested; for example, the dictionary

```
{Tinderbox: 1;Storyspace: {Editor: 2; Reader:3} }
```

Contains two elements. The first has a key of “Tinderbox” and a value of 1. The second has a key of “Storyspace” and the value of the dictionary, {Editor: 2; Reader:3}.

The dictionary .add() and .extend() operators now take a single argument — a dictionary of elements which will replace or extend the current elements.

The operator .unique() now preserves the order of elements in a list. "C:A;A;B" now return “C;A;B”.

$SiblingOrder is now editable. If set to any number 1 or less, the note will become the eldest sibling. If set to any number $ChildCount or greater, the note will become the youngest sibling.

The format code “X” converts numbers to Roman numerals. For example, 3.format("X") is III.

The format code “o” converts numbers to their localized ordinal. For example, 3.format("o") is “3rd” in English and “3e” in French.

The operator `isDuplicateName()` regarded all aliases as having duplicate names. This was not helpful.

Function declarations may optionally specify the type of their arguments. For example,

```
function append(x:list) { return x+1;}
```

declares that values passed to the function `append()` will be treated as lists.

A new operator, `.asString()`, converts sets and lists to a string representation. The much-used operator .contains() behaves differently for strings (where it searches for a regular expression match) and for lists and sets (where it tests for set membership). Occasionally, it’s desirable to perform a regular expression test on a list or set — for example, to ask if any of the members of $MyList begin with the letter “a”. In the past, this was accomplished by using the .format() operator; .asString makes your intent clearer.

A new operator .sentence(n) extracts the nth sentence from a string. For example, if $Text contains the passage

> Mr. Smith went to Washington. He shook hands. He kissed babies.

then $Text.sentence(0) returns "Mr. Smith went to Washington." and $Text.sentence(2) returns "He kissed babies." If the argument is omitted, the initial sentence is returned. The definition of “sentence” is heuristic, and varies depending on the locale. In the example above, for example, Tinderbox recognizes that the period following "Mr." ends an abbreviation, not a sentence.

The action create() once more evaluates its first argument, permitting you to compute an expression. Note that this means that paths should be quoted: create("/Containers/People/Mark"), as otherwise parts of paths may be evaluated as expressions. This change appears necessary to support create($MyString), which seems useful and necessary.

The new expression .count_if(var, test) counts the members of a list that satisfy the expression. For example, if $MyList is “1;2;3;4;5”, $MyList.count_if(x, x>3) returns the number of members of MyList that are greater than 3.

The new expression .sum_if(var, test) counts the members of a list that satisfy the expression. For example, if $MyList is “1;2;3;4;5”, $MyList.sum_if(x, x>3) returns the sum of all the members of MyList that are greater than 3. An optional third argument allows you to transform the members before they are added to the sum; $MyList.sum_if(x, x>3, x*x) returns the sum of the squared of each member that is greater than 3.

In eachLink() loops, the following properties of the link are now editable: type, source, and destination.

A new stream parsing action, .skipLine(), skips to the next carriage return or to the end of the stream. .skipLine fails if the string is exhausted.

.extract(regex[,case_insensitive]) and .extractAll(regex[,case_insensitive]) now accept an optional second argument. If that argument is true, the regular expression search is case-insensitive.

The operator `interval(s)` now converts a string to an interval. For example, interval("30:00") is an interval of thirty minutes. The two-argument version,

interval(start,end), continues to return the interval between two dates.

Adding or subtracting an interval from **never** now returns **never**.

An optional second argument for eachLink() allows you to designate the note whose links are to be examined. For example, eachLink(x, parent){…} performs an action on each of the links to and from the parent of this note. If the designator is this, it may be omitted: eachLink(x){…} performs an action on each link to and from this note.

In code fields, eachLink(){..} is highlighted correctly, as is each(){…}.

A new operator, createAdornment(path), creates an adornment at the designated path. If a note already exists with that path, no new adornment is created.

Browse Links

Revamped table layout for improved legibility.

Changing the link type in Browse Links… now fires the OnLink action.

Document Settings

The General Pane of Document Settings now offers a checkbox to disable special treatment of # and @ in note names.

Export

File ▸ Export Selected Note will now export the selected note, as well as images embedded in that note. If the destination directory does not yet exist inside the export folder, it will be created. For example, if the exported note is /people/Lincoln, the folder “people” will be created inside the export folder if that folder does not already exist, and Lincoln.html will be created inside this folder.

If several notes are selected, File ▸ Export Selected Notes will export each selected note.

The export sheet offers export to “Word®” rather than “doc”

Find

Find Bar: the Find Bar gains a prototype menu that lets you restrict a search to specific prototypes.

Gallery

Saved tabs in the Gallery may now be reordered by drag and drop.

Get Info

The count of notes found by an agent is now updated promptly in the Agent pane of Get Info. Formerly, the note count was updated immediately while the agent update was performed on a separate, slower thread. We now update the field after the agent has finished its work.

The Agent pane of the Get Info popover window now performs syntax coloring in the query and action fields.

The layout of the Get Info: Attributes pane is improved.

Images

The maximum image width for images in text is now an attribute, $ImageSizeLimit. It defaults to 1600 px.

Inspector

When Quickstamp is editing a Boolean attribute, it now displays a checkbox in place of the value textfield. This should make editing boolean attributes a bit easier.

Map View

In map view, [Return] creates a new note to the right of the selected note, if space is available. ^[Return] creates a note to the left of the selected note, and ^⌥[Return] creates a note below the selected note. If the initially-chosen location is occupied, a different location will be chosen.

Duplicating a note now positions the duplicate more predictably, while avoiding existing notes.

Outline View

View ▸ Expand All gains a keyboard shortcut of ^⌘-9.

View ▸ Collapse All gains a keyboard shortcut of ^⌘-0.

Text

The Text Pane Selector is now visible unless you hide it; previously, it was hidden until you showed it.

The User Attribute Inspector formerly reordered the selected note’s displayed attributes if a new user attribute was added to the displayed attributes list. The order is now preserved.

Edit ▸ Deselect All now operates in the text pane.

Watched Folders

When watching a folder in Finder, notes that correspond to deleted (or moved) files are now shown with their names struck through by automatically

setting $NameStrike to true.

Watched folders of all kinds are checked more frequently.

When a file in a watched folder is checked, its $LastFetched is now updated even if the note is unchanged. The note's $NotesModified attribute continues to reflect the modification date of the file.

DETAILS

Tinderbox now requires macOS 11 (Big Sur) or later.

Actions

Assigning a floating point number to an interval is now supported, and interpreted as assigning an interval of the stated number of seconds.

When setting properties of an attribute

```
attribute(attribute_name)[facet]="testing";
```

Both the attribute name and the facet to be set may now be enclosed in quotation marks.

A new attribute, IDString, provides a more compact alternative to $ID. For example, a typical value of $ID might be 1666143079. The corresponding IDString is tbx:BjT1Nn. $IDString may be used interchangeably with $ID.

if(){…} and if(){…} else {…} now restore regular expression back-references ($1, $2…) to their state prior to the if() statement.

The .paragraphList() operator was erroneously converting its results to lower case.

When attempting to reason about the type of the result of an expression in ambiguous situations, Tinderbox examines the values passed to binary operators like "+". If a value is interpretable as a number, Tinderbox prefers the numeric interpretation to a string. The rules for "interpretable as a number" were, however, too lax, and expressions like

```
"1883 —" + "1964"
```

were treated as numeric because part of the first string is a number. Now, the string is regarded as a numeric value only if the entire string can be parsed as a number. It is preferable, of course, to use intermediate attributes or typed vars in order to avoid the ambiguity entirely.

Converting a string to a dictionary yielded incorrect results if the value contained a colon:

```
$MyDictionary="l:able:baker"; // ➛ l:able
```

create( ) was too aggressive in evaluating its arguments, which caused trouble when note names contained apparent operators such as [ ].

Agents

Even lowest-priority "Occasional" agents now are run when the document is opened.

Export and Preview

Preview now exports (a) the current page, and (b) any pages to which the current page has text links, if those pages are not currently exported to disk. If the current page is already present on disk, the current version is saved and exported later. This arcane waltz is necessitated by ill-documented security measures which appear to offer little if any benefit to security, but which consumes time and resources that might do someone, somewhere, some good.

Tinderbox relaxes historic constraints on exported file names. The default replacement character in export file names, $HTMLExportFileNameSpacer, is now " " rather than underline. Punctuation other than "/" is now allowed in filenames. The default value of $HTMLFileNameMaxLength is now 100 characters, up from 8 characters in Tinderbox 1 and 24 characters in release 3.0.5.

The following characters are now removed from HTML file names:

```
. ~ /
```

Formerly, all punctuation was removed.

If a previewed note has no tag, Tinderbox adds a preview-only framework to ensure that the preview pane uses the expected character encoding.

In Text Export, resolved interference between a sheet asking for details of the export and a dialogue asking where to save the exported file.

When previewing HTML for a document that has an export folder, the preview file saves a copy of the existing file and replaces it after the preview is loaded.

The progress bar is once again displayed during HTML exports.

In HTML Export, the expression will now attempt to evaluate its second argument if that argument does not designate a template. For example, would export the top-level note named "colophon" using the template named in $MyTemplate.

Get Info

The highlight color for words in context in Get Info: Repetitive is chosen more prudently in dark mode.

Import

Text file import is now more tolerant of text files with unusual text encodings.

File with the extension ".json" may now be dragged into the view pane.

Infrastructure

Addressed a potential deadlock when undoing link deletion.

Memory leaks in ExportPathAttribute, URLAttribute, and HTMLMarkup have been corrected, removing problems for heavy use of the Preview pane in

notes where rules refresh the screen at frequent intervals.

Tinderbox no longer creates thumbnail images of the map text if there's not enough space in the map item for the text to be drawn. This saves some battery power.

Resolved a memory leak in action and expression tokenizers.

Map View

Geographic Adornments are updated more promptly when their location changes.

Fixed a possible crash when double-clicking a note in map view, observed on fast Apple Silicon machines running Ventura.

Using ⌥-Tab to focus on the view pane now triggers then focus animation, as expected.

The Map Background image is no longer upside down.

In map view, zooming out from a container does a better job of restoring the scroll position of the map.

News

At startup, Tinderbox checks with an Eastgate server to see if there is a news item the user has not yet seen. If so, the news is displayed in a separate window. News will only be shown once, but you can review the most recent news item by choosing Tinderbox News… from the Tinderbox menu.

Stamps

The stamp menu, and the stamp list maintained by the Document inspector's Stamp pane, are now updated when changes are made to the stamp notes in the Hints container.

Tabs

When switching a tab to use attribute browser from some other view, that tab did not always adopt the correct container if, in the past, the tab had been used for an attribute browser on a different container.

View ▸ Tab ▸ Close Tab is now available from the Attribute Browser and from the Text Pane.

The View ▸ Tabs ▸ Close Tab menu is once more disabled if there is only one remaining tab.

Improved the text position in the breadcrumb bar

The Chart popover controls have a more sensible number of tic marks.

Text

Format ▸ Text ▸ List… misled Tinderbox into thinking that a single note's text was actually a multi-note text. This should no longer occur.

When opening a document in which the preview pane is the current pane, the preview pane shader formerly claimed that "no note is selected" even when a note was, in fact, selected.

Copying or Cutting text in an alias could cause confusion in the placement of text links.

Spell checking is now enabled by default in the title field of the main window text pane, and some other text fields. It may be toggled by choosing Edit ▸ Spelling ▸ Check Spelling While Typing while editing the title in the text pane selected. This state is saved with the document.

Smart Links are now disabled in the built-in prototypes for Code and Action. This prevents expressions like "$MyString.at(0)" from being treated as URLs in Austria.

When links were added while the link pane was hidden, and then the link pane was displayed before changing the selected note, these newly-added links might not be displayed in the link lists. That should no longer happen.

CompressImages() is now performed after edits are inscribed, rather than doing this f0r each note when the document is saved. The latter is slow on large documents.

9.3.0

DETAILS

Actions and Attributes

The value of $Fill may now be a file path, such as ~/Desktop/pizza.jpg.

$Fill is now a file attribute, making it easier to change fill to an arbitrary image file in the Displayed Attributes table.

Date arithmetic no longer truncates seconds.

The time() operator no longer ignores seconds.

The dictionary describing links that is created by eachLink() includes two additional boolean elements. The key **isFirst** is true for the first link in the enumeration and false otherwise. The key **isLast** is true for the final link in the enumeration and false otherwise.

The error checker regarded actions consisting exclusively of white space as unparesable. These are now treated as empty actions.

An error in the parser caused failures when comments were included inside the body of an expression enclosed in curly braces, such as the body of an .each() loop.

A new designator, **adornments**, returns all the adornments that are contained inside this note. **adornments(..)** lists of all adornments that are siblings of **this** note. **adornments(/)** lists all top-level adornments.

In HTML export, any var created in an clauses remains available to clauses and subsequent actions.

The action parser was confused by strings delimited by single quotes, if those strings contained a double quote.

HTMLQuoteHTML is again active, even when HTMLEntities is false. Previously, unless HTMLEntities was true, HTMLQuoteHTML was ignored.

Tinderbox now stores integers as signed 64-but numbers, rather than signed 32-bit numbers. Note that Tinderbox converts freely between integer and floating point arithmetic, so not all operations on integers are guaranteed to use integer arithmetic.

Tinderbox could crash if the regular expression of a .contains() operator contained optional sub-expressions, if one of the optional back reference was unmatched.

A new attribute **$IsAgent** is true for agents and false for all other notes.

.textColor() now respects the current text font.

The statements

```
var:string x;

x=;
```

Formerly set x to the informative, but unexpected, value of “No such attribute” if x was a string. The local variable is now set to the empty string.

Document Settings

The title of the Document Settings window title now accommodates Unicode characters properly.

Export: Preview Pane

The preview pane intercepts clicks on links that correspond to Tinderbox text links, and selects the note that corresponds to the link destination. Formerly, this navigation required that the destination be selectable in the current view; now, Tinderbox will reanchor the current view or expand collapsed containers to allow the destination to be selectable.

inside() and **descendedFrom()** once again evaluate their argument. In recent releases, these operators have assumed that their argument is a path, and that is indeed the most common pattern. Nevertheless, we may need to evaluate an expression if, for example, we wish to parameterize a number of agents:

```
Query: inside($TheTarget(agent))
```

HTPath now recognizes expressions in which an attribute reference designator is stored as a dictionary value. For example:

```
var:dictionary x=...;

$Name(x[key])="value";
```

Get Info

In Get Info:Re[etition, the text excerpt now highlights the repeated word.

Get Info:repetition did not correctly handle re-sorting the table of repeated words.

Hyperbolic View

Corrected an error in Hyperbolic View which caused some crosslinks to be drawn with their arrow orientation reversed.

Import

Bookends and RIS import: $Authors was previously sorted in alphabetical order, but now respects the order in which the authors are listed in the reference.

When scanning watched folders, Tinderbox now respects documents with different file extensions. For example, "test.txt" and "test.jpg" will be imported as two separate notes.

Avoid a crash when importing a spreadsheet that contains a column labelled “Container”, corresponding to the system attribute of that name.

Infrastructure

Saved search strings could contain control characters forbidden in XML. Tinderbox now checks in order to prevent this.

Addressed a crash when an action attempts to enumerate the siblings of the document’s root note.

The prototypes for Code and Template notes now offer more tab stops.

Inspector

The Sort Inspector no longer grows to unreasonable width if the name of a user attribute is unreasonably long.

In the **Sort** pane of the Action Inspector, entering “none” in the search field, or clearing the search field, and pressing “enter”, will remove any local value for the sort attribute, restoring the inherited or default value if any.

Map View

Positioning of notes created in map view by pressing [Return] is once more correct. The new notes had been placed behind the selected note in recent test releases.

If a note has a HoverImage and a HoverExpression, the hover expression is now displayed even if the hover image doesn't exist.

Outline View

In outline view, ⌘-up-arrow and ⌘-down-arrow incorrectly changed the note’s map position.

Roadmap

Tinderbox no longer reselects the first item in torn-off roadmap views whenever the document has been changed.

Text

When displaying text from multiple notes, Tinderbox omitted the name of the final note if that note had no text.

When following a web link, Tinderbox now respects the fragment portion of the URL and no longer attempts to percent-encode the **#** character.

When creating links to a span of text that is selected in the text pane or a text window, if the mouse is clicked inside a text selection then the entire selection becomes the target of the link. If the mouse is clicked outside a text selection, the target of the link is the character nearest the click. When following a text link with a destination span, then entire span is highlighted, not merely the word closest to the click.

When editing a date value in the the Displayed Attributes Table, Tinderbox switches to long date format to avoid ambiguity. After editing a date, the date is now redrawn immediately in the date format specified by $DisplayedAttributesDateFormat.

9.2.1

DETAILS

The Copy Note URL was constructing URLs incorrectly, using an ampersand where a “+” is required. Tinderbox no longer opens a new tab when it can simply select the indicated note.

Navigation within Tinderbox from the Preview pane had been disabled, for reasons that are now obscure.

Normally, word clouds ignore aliases. If viewing the word cloud of a note and its descendants, however, and if the selected note is an agent, the word cloud now indexes each alias found by the agent.

Substantial improvement in speed of Bookends import, which had been delayed artificially by queuing work to the main queue that could be done in a worker queue.

9.2.0

NOTABLE

Actions

The operator **createAttribute(*name[, type]*)** creates new user attributes. If an attribute of the same name already exists, the operator has no effect and returns **false.** Otherwise, a new user attribute is created. Recognized values for **type** include string, number, boolean, date, color, interval, file, list, set, url, and dictionary; if **type** is omitted, Tinderbox creates a string attribute.

Map Views

Document Settings:Map options to expand notes if their name is too long were inoperative in 9.1.0 on newly-created notes.

DETAILS

Actions

[String].find(pattern) returned incorrect results if the string being searched contained multibyte Unicode characters preceding the occurrence of the pattern.

Syntax coloring: an unterminated quote inside a comment could attempt to treat successive lines as quoted.

Syntax coloring now colors typed declarations such as var:string .

Action syntax coloring failed to recognize comments that were not followed by a new line.

The dictionary returned by **document** now includes a key, **link-types**, that contains a list of link types defined in the document.

Protected **.following** from a possible crash if one of its arguments is missing.

Actions that construct the document dictionary failed if the document was unsaved

Link actions that refer to **source** and **destination** now validate that the link has not been deleted whilst the action was enqueued.

Agents

Addressed a crash when an agent action tried to move a note to a new container while the document was being read.

Copy URL

The command to Copy Note URL placed a “+” in the URL where an ampersand was required.

Gallery

Revised button labels in Gallery popover

Import

The following quotation was misrecognized as a CSV spreadsheet import:

Force, in the hands of another, exercises over the soul the same tyranny that extreme hunger does; for it possesses, and in perpetuo, the power of life and death. Its rule, moreover, is as cold and hard as the rule of inert matter.

The Iliad, McCarthy translation

This happened because the first paragraph might be a seven-column CSV, and the remaining paragraph might then be two columns, followed by five missing columns. Leniency is good, but this is too lenient; we now abandon CVS import of more than two fields are missing.

Infrastructure

Revised **CeresOutlinePreparer prepare:**, which was spinning up numerous worker threads without much gain, and which was causing deadlocks in large documents.

Retagging of notes at startup, and when the original note is selected, should now be more reliable.

A crash could occur when switching view types (e.g. View ▸ Map) while renaming a note.

Addressed a crash that intermittently occurred in Monterey when updating the text pane, because prepareTextPane cannot be called between beginEditing and endEditing

Revised the linkType XML format to allow line breaks in onLink actions. Note that this change is not backward-compatible before b545.

We now acquire a copy of the shared spellChecker at startup. NSSpellChecker.sharedSpellChecker is, apparently, not thread safe at initialization though this fact is not documented; trying to display the list of suggested notes while indexing could deadlock at startup.

On closing a document, we are now more careful not to enqueue requests to mark the document as changed that are received after the document has begun to close. Previously, requests arriving at the wrong moment could be enqueued on a stopped queue, which could prevent the document from closing.

The index that maps names to notes now has its own operating queue, removing an apparent race condition occurring during loading. This is likely unnecessary; I think the race cannot actually cause mischief. Nevertheless, an anxious compiler is an unhappy compiler.

Fixed a possible source of change manager crashes, involving testing whether a notification ought to be posted.

Inspector

The System Attribute pane of the Document Inspector now refers to attribute “Group” rather than attribute “Category”.

Map

Revised map item layout to improve spacing between title and subtitle, especially at small and large magnifications.

Outlines

In outline columns, the vertical position of the checkbox is more closely aligned to the baseline of the text.

Sorting

When a container’s sort method is **Don’t sort**, the **reverse** sort flag is ignored. Previously, unsorted agents swapped their sort order on each pass.

Startup

When loading documents, Tinderbox sometimes finished creating windows before the hypertext had been completely loaded. In consequence, view windows might not present all the subviews that they ought, since the corresponding items had not yet been created at the time the window was drawn. Views are now specifically notified when the view has been loaded.

Text

/

Trying to make a zip link of a name enclosed in parentheses [[(name).. failed to name the new note appropriately, because the parentheses were interpreted as a designator for a note named “”. This now works as expected.

9.1.0

HIGHLIGHTS

String Processing

A powerful new family of string processing operators help located and extract information from structured and semi-structured notes.

XML Processing

Tinderbox can now parse XML stored in Tinderbox notes or attributes, allowing you to extract information you need from a wide range of APIs and Web services.

JSON Processing

Tinderbox can now parse JSON stored in Tinderbox notes or attributes, allowing you to extract information you need from a wide range of APIs and Web services.

Stamps

Stamps may now be stored in the Hints container. They are stored as individual notes in /Hints/Stamps, and their actions are stored in the text of the corresponding note.

If you use File ▸ Built-In Hints to create or update the hints container, stamp notes are automatically created for each of your existing stamps.

Notes in the Stamps container with names in parentheses are treated as commentary rather than stamps. For example, the note (What’s This?) explains the container and contains no stamp.

Notes in the Stamps container with names beginning with a period are not listed in the Stamps menu, but may be used with the stamp(stampName) action.

To create stamps that appear as a submenu, continue to name them menuName:stampName.

Functions

Tinderbox now allows you to define your own functions — procedures you can use from any Tinderbox action. Functions are defined in notes contained in the container /Hints/Library; adding new functionality to a Tinderbox project is now as easy as dropping a note to a new container.

NOTABLE

Actions

Vars can now have optional types.

```
var:number x(5);
```

Available types include “number”, “set”, “list”, “date”, “color”, “date”, “interval”, and “dictionary”. Providing an explicit type helps Tinderbox provide the answer you want. For example:

```
var x:number(5); x=x+5; $MyString=x; ➛ 10
```

```
var x:list(5); x=x+5; $MyString=x; ➛ 5;5
```

```
var x:string(5); x=x+5; $MyString=x; ➛ 55
```

... (Range Operator)

The range operator ... constructs a list of numbers from a specified starting point to a specified end point. Note that the range operator is written as three periods.

```
1...3 ➛ 1;2;3
```

```
3...1 ➛ 3;2;1
```

The range operator can be useful for performing a task a specific number of times.

```
1...10.each(x){ var path="/container/item "+x;create(path);}
```

The range operator binds more tightly than arithmetic operators.

```
1...3 * 2 ➛ (1..3) *2 ➛ 2;4;6
```

+= and -=

These assignment operators are a convenient shorthand for incrementing or decrementing an attribute. For example, the two following statements are equivalent:

```
$MyNumber += 3;
```

```
$MyNumber = $MyNumber + 3;
```

Similarly:

```
$MyNumber -= 3;
```

```
$MyNumber = $MyNumber - 3;
```

These operators may also be used with lists and sets, and += may be used to append strings.

```
$MyString += "!";
```

eachLink(x){...}

Examines each link, either inbound or outbound to this note, with the exception of prototype links. The local variable x is bound to a dictionary of link properties, include source, destination, type, class, title, target, url, comment, and anchor. Changes to this variables values are *not* (yet) recorded as changes to the link.

[String].jsonEncode

This operator encodes Json strings, replacing forbidden characters such as returns with Json escape sequences. This operator was formerly know as .json.

List multiplication

Lists and sets of numbers may be multiplied by a number. Two lists of numbers may be multiplied if they have the same length, in which case their elements are multiplied.

```
1..3 * 1..3 ➛ 1;4;9
```

Link operations

createLink(source,destination[,linkType]) creates a link between any two notes. If a link of the designated type exists, no additional link is created.

Dictionary operations

Two convenience methods make it easier to work with lists and sets in dictionaries.

[Dictionary].add(key, value)

Sets the value of a given key. Equivalent to Dictionary[key]=value;

[Dictionary].extend(key, value)

Sets the value of a given key if it has not been set. If the key is already defined, appends the new value to a list of values.

The new operator .extract(*pattern*) returns the first matched subexpression of a regular expression. If the regular expression has no subexpressions, the entire match is returned.

The new operator .extractAll(*pattern*) returns a list of all matches for a pattern found in a string. For example:

```
$Text.extractAll("#[A-Za-z]+")
```

would return any hashtags such as "#Tinderbox" found in the text.

update(notes) asks Tinderbox to update one or more notes by evaluating their rule and edict. Notes may be an individual or group designator, or a list of paths. If the note has been evaluated recently, Tinderbox will not evaluate it again. update() returns a list of updated notes.

The expression $Text.highlights returns a list of strings that have been highlighted in the text using Format ▸ Highlight. The result may be limited to strings highlighted in a specific color by requesting $Text.highlights(*color*), where “color” may be red, green, blue, yellow, or magenta.

Names of user attributes may begin with a wider range of letters, including letters with diacritical marks.

Export

The Export pane now displays the selected note’s HTMLPreviewCommand, if it has one. A checkbox lets you temporarily disable preview command and Markdown processing, enabling you to see the input to the preview command as well at its output.

Get Info

A new Get Info pane, **paths**, displays all the paths in the document and lists each note on the selected path.

In Get Info: Paths, the count labels have been clarified slightly. The tooltip of table entries shows the full path of the note. Double-clicking a table entry will select the corresponding note and hoist the current view to focus on its parent.

Local Variables

Map View

If a note has a fill, ⌘⇧-drag inside the note to move the fill image.

Tabs

The selected tab is now tinted with the user’s accent color.

DETAILS

Action Notes

Notes in the containers /Hints/Library and /Hints/Stamps are typically action notes and use the built-in prototype Action. These provide automatic syntax coloring and autocompletion of Tinderbox actions.

A new boolean attribute, $IsAction, identifies notes that contain action code. These notes receive autocompletion and syntax color for actions.

A new built-in prototype, Action, sets $IsAction and other settings suitable for editing actions.

Action notes now offer syntax highlighting and autocompletion.

Syntax highlighting did not recognize attribute names that begin with lower-case letters.

Code fields no longer select the entire text after updates or when gaining focus.

Actions

The random() operator has been rewritten for better randomness.

Addressed a parsing error that occurred when a comment immediately followed a comment. Comments in actions were not terminated, as expected, by the Unicode line-separator character \u2028, which is inserted by ctrl-Return.

The action parser behaves better when handling unquoted paths, as in

```
create(/Sources/People/Jefferson);
```

Previously, Tinderbox attempted to parse the path, which could lead to unexpected results. Now, a path with an initial “/” is recognized and treated as if it were a literal. Note that literal paths containing commas and semicolons can confuse the parser; if you must include commas and semicolons in your path names, enclose the path in quotes or store the name in a string attribute.

```
create("/Sources/Martin Luther King, Jr./Speeches");
```

```
create($MyString);
```

[String].jsonEncode This operator encodes Json strings, replacing forbidden characters such as returns with Json escape sequences. This operator was formerly know as .json.

capitalize(string) and [String].capitalize() now return the expected result in languages other than English. Previously, these were limited to Latin alphabets.

$WordCount was often inaccurate in Russian and other languages that use non-Latin characters.

Corrected an odd case where the descendants designator could recurse indefinitely.

The attribute $Authors is now a list rather than a set.

The implementation of `.keys` was incorrect, causing failures when used in a binary expression such as $Text+$MyDictionary.keys.

When creating a new note where the newly-created note has a DisplayExpression, Tinderbox could erroneously replace the name with the display expression after using the mouse to move the insertion point.

Autofetch

If we autofetch a URL that ends in .jpg or .png, Tinderbox recognizes that the data represents an image and treats it accordingly.

Browse Links

A layout problem had crept into Browse Links . It has been sent to its room.

Displayed Attributes

Window ▸ Displayed Attributes ▸ Show/Hide Displayed Attributes now applies to all selected notes.

The Displayed Attributes table now displays ellipsis (...) if the text value extends to multiple lines, even if the visible line is not truncated.

In the Displayed Attributes table, the commands Format ▸ Text ▸ Insert Date/Time/Date and Time now insert the requested text at the insertion point, rather than replacing the current value with the requested text.

Tinderbox again respects $DisplayedAttributesFont. If no value is specified, Tinderbox defaults to $NameFont.

The displayed attributes table attempts more emphatically not to refresh values you are editing.

Document Settings

Document Settings:Map now properly reflects the status of Adjacent Notes For Composites and Opaque Adornments.

Document Settings:Outlines again handles Black and White Outline Titles appropriately.

The Document Settings window title failed properly to handle characters that don’t occur in English.

In Tinderbox Settings:Maps, the checkbox “Adjacent Notes Form Composites” again sets the default value of $NeverComposite as it should.

Explode

Explode… again has the shortcut ⌘⇧-E.

Explode… now removes any leading or trailing whitespace from titles of exploded notes.

Export

Corrected HTML export of relative links.

Tinderbox failed to recognize bold passages for some of its bundled fonts, notably Mercury and Ideal Sans.

RTF Export now removes the right margin from exported text.

When the 'Preview' pane of the text pane is active, Window menu now offers to show the Export Pane, previously the "HTML Pane.”

Export Preview failed to turn off entification when $HTMLEntities was false.

Corrected a possible crash when exporting a single note, arising because a closure was trying to capture a stack-allocated instance of HypertextQueues.

The built-in Composites container is now set to neither export itself nor its children.

The Preview pane is now faster in large documents. Before export, Tinderbox builds a table of file names for each exported note. Siblings with identical names are automatically disambiguated during export. However, this process is inherently quadratic in runtime. In large documents, this is negligible when exporting an entire document but significant when previewing an individual page; we now avoid testing name collisions in preview.

When evaluating ^linkTo()^ from the text of a note included by another note, the link url was evaluated relative to the included note rather than the current page.

Corrected a problem in export of relative links when the relative link is ^included from a note that is not a sibling of the current note.

CommonMark preview now permits embedded HTML.

Fills

When setting a value for $Fill, Tinderbox now accepts a file name with or without an extension. If the user file folder contains the file “TestFill.png”, then setting $Fill to “TestFill” or to “TestFill.png” is equivalent.

The Fill folder and its subfolders may now contain aliases of image files as well as the images themselves.

Hyperbolic View

View ▸ Browse Links sometimes failed to open the Browse Links popover from hyperbolic view.

Import

Tinderbox declined to import text files longer than 50K. That’s too low a limit these days — especially since this might include HTML files with tons of stylesheets and javascript that won’t show up in the import.

Infrastructure

We no longer rescan all notes when loaded, as in large documents this takes a lot of time.

The tagging queue is now lower priority to avoid congestion after opening large documents.

The indexing thread is now shut down more promptly, which should eliminate intermittent crashes after closing a document on unusually-busy machines.

Addressed a potential crash when the fill directory has multiple subfolders.

Corrected a race condition and potential crash when updating the text pane after applying a stamp.

The menu bar and the About… window on Chinese systems said “Tinderbox 8”, not “Tinderbox 9”.

Addressed a crash on reverting a document.

Inspector

In the appearance inspector, the label for the Large Badge checkbox is no longer clipped.

Several anomalies with the Window ▸ Prototype Inspector (⌘-3) command have been corrected.

The link Action field of the link types inspector now performs syntax coloring.

In the stamp inspector, if a stamp is selected when we create a new stamp, the new stamp is inserted immediately below the selected stamp and its initial name is derived from the selected stamp.

The inspector window no longer grows automatically if you have lengthy code into the Stamp code pane. The Stamp Inspector’s vertical height could grow without limit if a very long action were pasted into the Action field. (I was unable to reproduce this until I literally pasted the whole megillah into the action field; this made things clear.). The vertical height is now limited to 750px.

Links

When the link-creation popover appears, the initial keyboard focus is now set to the destination name field if a new destination note is being created. Otherwise, the initial keyboard focus remains on the Create Link button.

The Export to: button in the export inspector now has the correct layout height even if the export folder is not present.

When Tinderbox adopts a smart link in the text as a Tinderbox web link, it now gives sets the path of the new web link to *untitled.

Map View

If a note has a fill, the fill image is scaled to fill the width of the note, and the center of the image coincides with the center of the note. The attribute $FillOffsetY moves the center of the fill image vertically.

In map view, you can zoom into an alias of a container by double-clicking its body or by selecting the container and pressing the down-arrow key.

Pressing return to create a note no longer insists that the new note may not be placed on an adornment.

Outline View

The Home and End keys now function in outlines.

In outline view, it is no longer possible to outdent the parent note of a hoisted outline. If the hoisted parent note is selected, ⇧-Tab is ignored.

Recent Badges

Using the Recent Badges contextual menu to select a badge now places the selected badge at the top of the recent badges menu.

The Recent Badges contextual menu now applies to all selected notes.

Tinderbox could save unreadable files if the recent badges list included a badge name containing and ampersand or left-angle-bracket.

Roadmap

Adjusted the layout of the Roadmap’s list cells, which had unsatisfiable constraints.

The roadmap popover now opens with the initial focus on the outbound links list if there are any outbound links. Otherwise, the inbound links list gains the initial focus.

Improved keyboard navigation in the roadmap, which now responds to [Return] as well as the spacebar to follow links from the inbound or outbound links lists. The first responder in the popup is now the inbound links list.

Improved the appearance of unselected links in the roadmap.

Text

In the Format Menu, Strikethrough is checked when the start of the current selection is struck through, and Underline is checked when the start of the current selection is underlined.

The attribute $TextHighlightBlue was misspelled, and as a result attempting to highlight selected text in blue resulted in gray highlights.

Copying and pasting text with text links could fail, especially in older documents, because a pasteboard descriptor for the ID was written as unsigned but read as signed.

Changing the default text font in Document Settings now updates both the text font and font size in notes using the former default font and font size.

Positive values of $NameLeading now represent a multiple of the natural line height. For example, $NameLeading of 1.5 adds a half-line of extra space between lines of the title, and $NameLeading of 0.9 sets the title 10% tighter than normal.

New attributes $TextColorRed, $TextColorBlue. $TextColorGreen, and $TextColorGray let you control the color applied by Format ▸ Style ▸ Red and related commands. Changing these attributes does not change text colors previously applied, but affects future applications of these styles.

New attributes $TextHighlightRed, $TextHighlightBlue. $TextHighlightGreen, $TextHighlightMagenta, and $TextHighlightYellow let you control the color applied by Format ▸ Style ▸ Highlight. Changing these attributes does not change text highlights previously applied, but affects future applications of these styles.

Format ▸ Style ▸ Standard Font, when selected from the view pane, failed appropriately to adjust text color and background color for dark mode.

Automatic link substitution is controlled by $SmartLinks and should no longer be affected by $SmartQuotes.

9.0.0

HIGHLIGHTS

AI

Tinderbox automatically scans your notes to locate names, places, and organizations. New **Taggers** let your documents define terms and synonyms to help your agents do more with less work.

Sentiment analysis helps assess the tone of comments and reviews.

Highlighters let your document automatically highlight selected words and phrases. Built-in highlighters are provided for editing actions and for Tinderbox taggers, but it is easy to add your own highlighters. Notes opt-in to highlighters, either individually or by inheriting a highliter from their prototype.

COMMAND BAR

The command bar **(Help ▸ Commands & Info…)** provides an abundance of information about your Tinderbox document. Some possibilities:

- Quickly open any Tinderbox document you've used before, by typing “Open ” followed by part of the document name.
- Select any note by typing "Select" and its path.
- View Tinderbox tutorials (e.g. “View video”)
- Open any inspector pane (e.g. Open Border Inspector)
- Get help from aTbRef (e.g. Explain Export)
- Explain Tinderbox actions and their usage (e.g. "collect_if")
- Define Tinderbox attributes (e.g. $Width)
- Check whether you need an upgrade ("Can I update")

ACTIONS

A new type of attribute, Dictionary attributes, are available. Dictionaries associate pairs of *keys* with *values.* The expression

```
$MyDictionary["Detroit"]="Michigan"
```

sets the value of "Detroit", and the expression

```
$MyDictionary["Detroit"]
```

returns the value associated with Detroit. Dictionaries provide a powerful tool for representing and using tabular or associative data.

Many new actions are now available, making Tinderbox even more powerful and expressive. Of particular interest:

- **.following** makes extracting information from forms and emails easier and faster.
- **.trim**, **.deleteCharacters()**and improvements to **.substr()** help you clean your data
- **.wordList, .nounList**, and **.paragraphList** help you locate and extract information you need.
- Powerful new graph operators let your identify the **neighbors** linked to a note, the notes on a named **linkPath**(), and the shortest path between two notes.
- New list operators let you easily examine the **.first** and **.last** elements in a list, transform the elements of a list, or select elements of special interest. A new list operator, **.collect(label, expression)** performs computation on the elements of a list. Each item in the list is bound in turn to label, and then the expression is evaluated. The operator returns a list of all the results. **.collect_if(label, condition, expression)**, selects items from the list. Each item in the list is bound in turn to label. The condition is then evaluated; if true, then the result of evaluating expression is appended to a list of results.

GALLERY

You can now save Tinderbox view tabs in the **gallery,** and can retrieve them whenever you like. This makes it easy to save and document complicated views without keeping them on your tab bar.

View ▸ Tab ▸ Gallery displays a list of current tabs and a list of saved tabs for the current document. A tab can be added to the gallery list, making it available later — even if the original tab has been closed. Any saved tab may be added anew to the tab bar.

HYPERBOLIC VIEW

Hyperbolic view has been revised to provide better layout, especially in large a complex documents.

Hyperbolic View now lets you turn off crosslinks, which frequently obscure complex documents.

MARKDOWN

Tinderbox now provides built-in support for writing and previewing Markdown for writers who prefer Markdown to styled text. Tinderbox previews Markdown notes in the preview pane, and also automatically streams to Marked2.

Text links within the document are now interpreted and followed by the preview pane.

NOTABLE

Actions

The **attribute()** operator lets you examine and change the properties of Tinderbox attributes.

The **document()** operator provides information about the current document.

Comments may be added to explain actions.

The **.next** operator generates convenient note names and other strings in a sequence. For example:

```
"footnote 1".next ➛ "footnote 2"
```

Agents

Tinderbox provides better control of agent priority.

- Highest priority agents run every few seconds.
- Normal priority agents are updated at approximately ten second intervals.
- Low priority agents are updated every minute.
- Lowest priority agents are updated every five minutes.
- Occasional agents are updated every hour.

Appearance

Tinderbox provides beautiful new fonts — Archer and Decimal — and a new color scheme Sunny that shows them off beautifully.

Badges

The item contextual menu contains a new submenu of **Recent Badges** that contains a list of badges that have recently been selected using the Badge Picker or the Appearance Inspector. The most recently-used badge is listed first. Badges that are set by actions, displayed attributes, or the attributes pane of Get Info are not taken into account by this menu.

Export

Exporting large documents is now significantly faster. Exporting large document involved checking millions of names, since each note and each included note may have a template that must be located by name in the document. Caching the name attribute reduces export times noticeably.

Improved performance of HTML Export by removing some redundant calls creating extra copies of note text and ensuring that indexing is up-to-date before export.

My own weblog now exports in 44 seconds rather than 91.

Hyperbolic View

Hyperbolic view now offers a control for spread, which determines the angle between child nodes.

The Highlight button in Hyperbolic view now lets you select a link type to be highlighted.

Inspector

Code fields in the inspector now offer syntax highlighting and autocompletion, making it easier to write actions and to avoid mistakes.

The Inspector may now be resized. Code fields, in particular, may now be expanded at will to accommodate more complex logic when needed.

Text

When making a ziplink, Tinderbox displays a list of possible targets in a popup window. Pressing down-arrow selects the first target, and pressing down-arrow again selects the second target. Return completes the link to the selected target. Option-Return is now accepted as equivalent to option-clicking on the list; the target is copied to the current ziplink but the ziplink remains open, allowing you to add other options. For example, you might type `|Anchor]]` to add a custom anchor for the ziplink.

DETAILS

Tinderbox now requires macOS 10.13 or later.

Actions

Local variables may now be initialized when they are declared:

```
var x=5;
```

The operator **.textColor(*color*),** when applied to styled text, sets the foreground color of the text. For example

```
$Text="example".textColor("#FF0000");
```

Sets the the note's text to a red word.

In dictionary expressions $MyDictionary[x], the expression inside square brackets was not evaluated.

Actions may now modify the default value and suggested values of an attribute.

```
attribute(attributeName)[suggested]="value 1; value 2";
```

```
attribute(attributeName)[default]="value 1";
```

collect() , **sum(),** and **sum_if()** now omit notes for which $Searchable is false.

Added a new action, **.avg**, which returns the mean of a list or set of numbers.

.format("I") and .**format**("L"), when applied to intervals, now format the intervals according to the current locale. The lower-case "I" format uses abbreviated form, while "L" spells out the interval in a phrase customized to local usage.

.contains() and **.containsAny()** now use Apple's regular expression engine rather than the BOOST engine. The change is unlikely to cause problems in working agents, and will provide better results when working with non-Latin characters.

When **.trim** is applied to styled text, it now preserves the text styles of its result.

The twitter() action is no longer supported.

[String].contains() now clears the list of back references from previous processes, so $0 and $1 correspond to its own results, not those from prior expressions.

Elements in lists, sets, and dictionaries may be extracted with the bracket operator

```
$MyList[1]
```

```
$MyDictionary["epic"]
```

The argument inside the brackets may be quoted, but the quotes may be omitted if the argument is a simple string.

```
$MyDictionary[epic]
```

This has the same effect as **.at( )**, but may be more convenient.

After each iteration of an each{} loop, accumulated back-references are cleared. Formerly, back-references from each iteration were retained for the lifetime of the loop, making it very difficult to retrieve the desired reference.

runCommand() no longer converts newline characters to carriage returns, a relic of ancient times.

Set attributes have been reimplemented to improve performance in large sets.

Agents

If a note was previously $Searchable and then $Searchable is set to false, the note will eventually be removed from the agent. Previously, the note might persist indefinitely.

Attribute Browser

When examining either a list or a set, the Attribute Browser examines individual elements of the collections rather than the overall value. Formerly, this behavior applied to sets but not to lists.

Composites

The default value of **$NeverComposite** can now be set in Document Settings: Maps.

In new documents, the default value of $NeverComposite is initially true.

Crosstabs

The summary line at the bottom of Crosstabs view now chooses an appropriate background color in dark mode.

Displayed Attributes

Autocompletion again makes sensible suggestions when editing lists and sets in displayed attributes.

Explode

The default setting, Break At Paragraphs, did not delete the return character from the title of the new notes, causing confusion and trouble.

Export

⌘⌥-E now cycles through all three kind of text pane, not only Edit and Preview.

Explode now has the shortcut ⌘⇧-E

Tinderbox documents failed to save their export folder if the export folder contained non-ASCII characters. Tinderbox should now remember the export folder more reliably; this will also improve previews that depend on stylesheets in the export folder.

$HTMLMarkDown is now deprecated. Use $HTMLMarkdown instead.

Spans of text using the note's value of $CodeFont are now exported with >code> tags. The markup applied is determined by **$HTMLCodeStart** and **$HTMLCodeEnd**. The Style pane of the Export Inspector lets you edit these.

Avoid a possible crash when using the ^setRoot()^ markup element in preview without an export template.

Tinderbox automatically communicates with the *streaming preview* of the application Marked2 (https://marked2app.com). When you select or edit a Markdown note in Tinderbox, it will automatically be sent to the Marked2 streaming preview window.

Marked2 streaming preview evaluates markup elements such as ^value()^ embedded in the markdown text.

^action()^ is now performed on the agent queue to avoid conflict with other actions running concurrently.

RunCommand() and Markdown Preview no longer balk at inputs that exceed 65535 characters (about 10,000 words, or 40 double-spaced pages). Shell

tasks, including classic Markdown preview, deadlocked if the data to be placed on stdin exceeded 65536 UTF8 characters.

A longstanding error in the spelling of HTMLOrderedListItemEnd (formerly HTMLOrderedsListItemEnd) has been corrected.

Revised code for $HTMLLinkExtension, which was causing problems when linking to notes with an extension other than .html . If the destination extension is not .html, $HTMLLinkExtension is now ignored.

Find Bar

The Search field of the find bar is now larger if the window is sufficiently wide.

The Find Bar no longer displays a menu of related words as autocomplete suggestions.

The Find Bar no longer offers automatic suggestions for related words. Suggestions may now be requested by pressing F5.

Tinderbox is less inclined to remove the Find highlighting in the text pane prematurely.

A small icon in the upper right-hand corner of the Find, Get Info and Roadmap popovers indicates that these popovers may be torn off.

Geographical Adornments

Tinderbox was creating dozens, sometimes hundreds of unused CVDisplayLink threads to support objects that might be, but were not in fact, geographical adornments. Geographical adornments have been temporarily sent on vacation and will return, after a rest, with better performance.

Get Info

The contextual menu of the Get Info::Attributes pane now offers options to Copy and to Evaluate the current value.

The Get Info popover's list of panes is wider, improving appearance on Big Sur.

The attributes pane now adjusts properly to your preferred Display Attribute Font Size, and it adjusts immediately if you change the size.

A small icon in the upper right-hand corner of the Find, Get Info and Roadmap popovers indicates that these popovers may be torn off.

The stop list created by Built-In Hints now takes precedence over other stop list notes. If a custom stoplist was previously installed in the support folder, Built-In Hints will copy it to your document. Stoplists may contain comments. Line that begin with # or // will be ignored.

Hyperbolic View

The hyperbolic view now does a better job of handling the singularity when lines fall exactly on the diameter of the Poincarré disk.

Import

DEVONthink import set the $SourceURL incorrectly when processing multiple drags.

CSV import was handling escaped quotation marks "" incorrectly and should now give more consistent results.

When importing dragged text, Tinderbox now attempts to interpret it as TSV if the first tab precedes the first comma, and as CVS if the first comma precedes the first tab.

If a watched folder or DEVONthink group contains an item that is either a comma-separated value or tab-separated value file, Tinderbox will attempt to expand each record into an individual note.

A new attribute, **$RISDictionary**, provides a dictionary of information extracted when a reference is imported.

Ziplink expansion in watched folders sometimes created new notes rather than linking to existing notes, because the imported notes were being named asynchronously.

Inspector

The stamp inspector no longer truncates stamp names prematurely.

In the stamp inspector, when a single note is selected, the Apply button reflects the note's DisplayName rather than its Name, and the name is correctly encoded.

The Tinderbox icon in the inspector now has a transparent background, which looks better in dark mode.

The Stamp Inspector's action pane now offers syntax highlighting.

All code-editing text field now use the default value of $CodeFont for the current document.

When creating a new stamp, the initial stamp name defaults to an unused name rather than simply "stamp name", making it easier to create short-lived, temporary stamps.

The action field in the Link Type pane of the Document Inspector now wraps.

In the user attribute inspector, creating a new attribute empties the description field immediately. Previously, the description was cleared after the new attribute was named.

The **Apply** button in the Stamp pane of the Document Inspector is now enabled only if (a) a stamp is selected and (b) at least one note is selected. If there is nothing to apply, or if there is nothing to which this stamp would apply, the Apply button is disabled.

Action inspector now disables editing actions when the selection is empty.

The inspector is now updated properly when the selection is changed in the view pane.

Code editor fields now provide autocompletion for designators in many contexts that accept a designator.

Link Actions

Link actions are no longer performed when the document is being loaded, but only when the link is first created.

Link actions could still be run when the document was loaded, especially in large documents, thanks to asynchronous link loading. This should now be corrected; loading a document will no longer run its link actions.

Links Pane

The Inbound and Outbound Links lists in the text pane share the available width more effectively between the link type and the note name.

Map View

For curved links leaving and entering the corresponding edges of two notes — for example, leaving the left edge of the source and ending at the left edge of the destination — the link arrow was drawn at the wrong angle if the two edges were aligned.

Improved link layout and link widget placement in some constricted situations in which link source and destination were aligned vertically or horizontally.

When dragging a note outside the map window and then back into the map, we reposition the moved note without creating an unwanted new note.

Translucent fill images are again properly composited in map and outline backgrounds.

The User Fill folder may now contain images of type .jpg, .jpeg, and .png.

The User Fill folder may also contain folders of image files.

In map view, if an interior note is locked, clicking on it will no longer begin a drag.

In map view, when using ⌘-drag to remove a note from a composite, the note is removed from the composite immediately. Previously, it was removed from the composite at mouse-up. The new approach makes the effect of ⌘-drag more clear.

The attribute $ShowTitle is now marked as deprecated.

A new attribute, $HideTitle, allows you to suppress display of a note's title and subtitle in map view, allowing more space for text display.

Improved handling of keyboard focus in maps after Edit ▸ Cut.

Markdown

The default template for Markdown notes is now taken from /Hints/Markdown/template, which is created by **File ▸ Built In Hints**. It automatically includes the styles in /Hints/Markdown/style, which in turn reflect custom styles you may have stored in the Markdown folder of the Application Support Folder. You may customize the default template to add headers, footers, or other elements.

Outline View

In outline view, the left-arrow key collapses the selected note if it is expanded. If not, it selects the parent of the selected note if the parent is visible in the view.

In outline view, **View ▸ Arrange ▸ Move Note Up** (⌘-up-arrow) and **View ▸ Arrange ▸ Move Note Down** (⌘-down-arrow) now scroll if necessary to ensure that the moved note remains visible.

In outline view, if you use the arrow keys, fn-arrow keys, or create a note that is not visible in the window, Tinderbox scrolls the window to make the selected note visible.

Outlines and charts could attempt to update their sorting for various reasons as you were editing the name of a note, which could sometimes sweep that note out of view. This should no longer happen. Sorting should also be carried out more promptly after a note has been edited.

Performance

Tinderbox compresses images stored in note text more aggressively, making files more compact.

Revised TbxMapCommands::DidCreate to avoid redrawing items repeatedly after creating a new note.

Actions performed by scripts are now dispatched to the property queue instead of being performed on the main queue.

The change manager now confirms with its document that views that require changes still exist when it's prepared to notify them. Failure to do this led to infrequent but hard-to-diagnose crashes in the change manager. ChangeDocket is now thread safe.

In large documents, Tinderbox spent a great deal of time looking for Taggers in the Hints folder because it was using the most general form of HTPath::Get, and because it was called immediately after the name index had been invalidated, this fruitless search was unusually slow.

The Tot importer periodically set and restored the selection, which would interrupt editing sessions in the key attributes window. This was not necessary, and should no longer occur.

The TbxTextPaneAdjuster was checking out a list of text links on one thread while TbxTextPaneListener was deleting them on the main thread. We now perform the critical section of TbxTextPaneAdjuster's work in small chunks on the main thread .

After editing text that deleted or replaced text containing a text link, the TextPaneListener deleted the link in the background. This could cause trouble if another task was in the process of obtaining or using a list of links that contained the deleted link, a situation most often arising when activating Tinderbox from the background. We now perform the deletion on the main queue. Additionally, with reference to belt and suspenders, the link manager now serializes access and its use from different threads ought not to cause problems.

In export, when running actions to ensure that notes were up to date, it was possible for some update tasks to remain on the agent queue after the markup elements that invoked them had been deleted. This left a string& pointing to nil, and led to crash with std::bad_alloc. We now capture the string in the closure and walk once more the paths of righteousness.

Previously, we created a new tagger object for each note that is to be scanned. This is inefficient. Now, we reuse one tagger, adapting it for each note.

Uncovered a long-standing error which caused the document to report that it was not being closed after it had been closed.

Roadmap

A small icon in the upper right-hand corner of the Find, Get Info and Roadmap popovers indicates that these popovers may be torn off.

Stamps

Using a stamp to make or delete a link now updates the Links pane promptly.

When applying a stamp to a note that is on exactly one adornment, the designator adornment is bound to the adornment. If the note is on several

adornments, the designator adornment is bound to one of those adornments.

Tagger

The tagger abandoned initial tagging prematurely. It won't do that any more.

The tagger conflicted with autosave, which suspended its queue. Tagging now has its own dedicated queue, and is better behaved.

Text

Smart links are again converted automatically to Tinderbox links.

Tinderbox's custom Font Manager has been revised to provide support both for Tinderbox built-in fonts and for system fonts, both before and after Big Sur.

Images pasted into text have less impact on file size.

The displayed attributes table now uses lining figures, which are more appropriate for tables.

In the displayed attributes table, the values pulldown menu handled "normal" incorrectly. "Normal" now restores the inherited or default value.

When dragging a text selection into a Tinderbox view from Safari, Tinderbox sets the $URL of the new note to the URL of the page from which the text was copied.

Addressed a crash arising in a stamp that replaces text from $Text that contains links automatically created our updated from the smart link scanner, because TbxTextLinkAdjuster was caching stale links.

Corrected a layout problem in setting up the text preview window on Big Sur.

The URL button had vanished from the URL attributes in the Get Info:Attributes pane. It has returned to duty. The file button had also absconded; it has returned, too.

Revised autocompletion in Displayed Attributes Tables and in the Attribute pane of Get Info to handle autocompletions correctly when they incorporate multiple words and non-word symbols.

The new Boolean attribute $SmartLinks determines whether the text pane automatically recognizes URLs. It can be modified using **Edit ▸ Substitutions ▸ SmartLinks.**

The menu now **Edit ▸ Substitutions ▸ Smart Quotes** now toggles the attribute $SmartQuotes, and is equivalent to **Format ▸ Text ▸ Smart Quotes**. Smart Dashes are also controlled by Smart Quotes.

Format ▸ Style ▸ Standard Size was too aggressive in resetting the font size and changed the font traits as well. It now respects the font family, bold and italic traits while changing the size.

Corrected a crash occurring when cutting or deleting text containing text links while the deleting links were being animated.

When the font size of the displayed attributes table was changed, Tinderbox failed to update the attribute column's font in the currently-selected note.

The tooltip for attributes in the displayed attributes table is the short description of the attribute, if available.

$DisplayedAttributes is now a list rather than a set. Sets have been reimplemented to improve performance, but are now more aggressive in asserting their control of the sequence of their elements. It is better to allow duplicate DisplayedAttributes in order to allow better control of the order in which they appear.

Timeline

In timeline view, the background color of the lists of unspecified and out-of-range dates now adapts top dark mode.

Toolbar

Tinderbox remembers whether the toolbar was visible when a document was saved, and if so makes it visible when the document is again loaded.

When using the parking space to link to a note by name, it was possible to link to the root note unexpectedly if the document name matched the name of a note elsewhere in the document. The root note is now excluded from the name index.

Views

Copy View As Image puts PNG as well as PDF data on the pasteboard.

The item contextual menu no longer reflects shortcut keys, which is consistent with current human interface guidelines.

TextClipping files may now be dragged to the view pane.

Addressed a Big Sur crash that could occur when magnifying or shrinking a view.

Tinderbox 8

Changes during Tinderbox v8.x.

Version 8 releases are listed below, in reverse chronological order.

8.9.2

NOTABLE

Agents

Revised agents to use background processes more efficiently. Agents should now be more efficient and reliable.

Map View

Geographic adornments are working once again.

When creating links, the rubber-band line is again drawn across all panes, not only the view pane.

Ziplinks

Format ▸ Text ▸ Ziplinks toggles whether the selected notes allow ziplinks.

DETAILS

Actions

Corrected behavior of .replace(), which could perform its replacement repeatedly if more than one replacement was performed.

jsonEncode() no longer escapes single straight quotation marks.

Agents

Revised AgentState::QueryHasChanged, which deletes an agent's aliases in preparation for regenerating them. Deletions were performed asynchronously on the main thread, which meant they could be delayed. In the worst case, this might cause some havoc at quit time as a pending deletion task might remain in the event queue as we disassemble our hypertext. We now perform the deletion on the AgentState's deletion queue, which lets us wait at quit to ensure pending operations complete in proper sequence.

AgentState::DeleteNode, which is called on the agent thread, now hides the note on the main thread.

Modestly increased the quality-of-service requested for the agent queue in order to avoid starving it of resources on busy machines.

The newly-revised agent system was trying to update the sort order of agents whenever a new alias was added or removed, making an efficient sort O(N^2log(n)). This caused trouble when adding an agent that adds numerous aliases; adding 1000 aliases required about 10 million unnecessary comparisons.

Additional fixes to agent updates to ensure that view additions and deletions occur on the main thread.

Improve efficiency of initial note placement inside agents when CleanupMethod is "none"

Attributes

$Path is now marked at intrinsic in newly-created files. Since $Path is a read-only, computed attribute, it is inherently intrinsic, but marking it as intrinsic is likely a good idea.

Built-In Prototypes

The built-in Reference prototype now includes $URL as a displayed attribute as well as $ReferenceURL. $URL holds a the URL to the references in the reference manager, $ReferenceURL holds the URL of the source if it has one.

Get Info

An Inspector may now be opened or closed from a torn-off Get Info window.

Import

When interpreting ziplinks in watched notes, Tinderbox could sometimes make a new note for a ziplink target that had not yet been imported. Tinderbox now imports all the notes and then expands ziplinks.

Infrastructure

Corrected a potential crash when importing watched notes, because the asynchronous ziplink expander failed to capture the target note.

Addressed a possible startup crash in TbxCrosstabsDelegate in Big Sur.

Backed out a performance enhancement in LinkManager::All(), which obscurely caused memory issues when opening multiple files simultaneously.

Quitting while Tinderbox was updating a watch folder could crash because Tinderbox sometimes deferred ziplink expansion and then restarted work while the document was being deleted.

Addressed a crash after deleting a note when updating the lists of links and suggested links.

Addressed a crash where the TbxZiplinkRenamer could accept new work while the document was being closed.

Maps

Modified Content::HasText to be much faster, and revised TextScribe::Draw to speed up the common cases in which there's no text to draw or not enough room to draw it.

8.9.0

HIGHLIGHTS

This release is built ready for macOS 11 and Apple Silicon.

NOTABLE

Actions

The internal representation of dates in handling actions has been changed in order to reduce or eliminate problems handling dates prior to 100 AD/CE.

The operator **containsAnyOf()** is now case-sensitive. **icontainsAnyOf()** is case-insensitive.

Attributes

$**UpdateTextLinksAfterRename** is now a member of the Textual category.

Attribute Browser

The attribute browser now accepts Magnify and Shrink commands and scales rows appropriately.

Export

The exportedString() operator now binds **current** to **this** note. Formerly, **current** was only bound in export templates.

The first line of an Attribute Browser export now contains the names of the attributes exported in each column.

Import

When receiving a dragged email from Mail.app, Tinderbox now records the email's URL in the new note's URL attribute.

When pasting an outline from OmniOutliner, Tinderbox now understands notes associated with OmniOutliner items and saves them in the corresponding note's text.

When dragging an item from DEVONthink, Tinderbox will populate the new note's $SourceURL with the URL of the dragged resource.

Removed menu option to watch Evernote notes, as Evernote 10 lacks scripting support. Existing installations will continue to work with the legacy

release.

View Tab Information

A new checkbox in the Tab information popover for map, outline and chart views allows you to **Hide Breadcrumbs**.

Link Types

When a link is created with a new link type, the new link type is added to the link types menu.

Text

Option-clicking in the ziplink popup menu autocompletes the chosen ziplink but leaves the ziplink open for further editing. For example, after typing [[tar and then choosing “target” from the popup while pressing the option key, the text will read [[target and wait for you to complete the ziplink.

If a stamp changes the text, Tinderbox will immediately expand any ziplinks found in the revised text.

The button to launch the displayed attributes popover is now labeled with a grid. It was formerly labeled with a “+” indicator, which could be confused with the “+” button used to add tabs to the tab bar.

Tinderbox 8.8 automatically updates text links after a note is renamed, if (a) the text link’s destination is the renamed note, and (b) the text link’s anchor text is the old name of the renamed note. This is often the desired behavior, but may in some instances be unwanted. An option to disable automatic renaming is available in Document Settings: Text. Automatic renaming may also be disabled on individual notes by setting the value of the new attribute **UpdateTextLinksAfterRename.**

DETAILS

AutoFetch failed to perform its task because the new Autofetch logic did not explicitly **resume** its task object.

Addressed a rare crash when redrawing views, caused by an overeager data test in - MapItemView margin.

A misaligned checkbox in Document Settings: Text was observed in Big Sur.

The Document Settings: Text control for displayed attributes date format failed to effect the desired change, thanks to the renaming of $KeyAttributesDateFormat.

In Big Sur, the width of the pane list in the Get Info popover was insufficient.

When dragging links in map view, Tinderbox generated console chatter reflecting that its overlay view was behind the some other views, such as the attribute browser’s view. This was alarmist because the other views are hidden. Tinderbox now checks that the offending view is not hidden before complaining.

The .replace() operator triggered the text link adjuster on any .replace() action; it should adjust text links only if the target of the operation is a note’s $Text.

Averted a crash in .split(pattern) if the pattern is not a valid regular expression.

In a torn-off Get Info window, the map pane failed to scale as the window size increased.

$Text.replace() no longer updates text link positions when called during export. As an action, $Text.replace() typically changes the text and so updating text links is very desirable; during HTML export, however, we’re typically manipulating a note’s text to an unstyled string and don’t want the side-effect.

Took steps to reduce or eliminate problems with link actions when editing link types in the inspector.

In the system attribute inspector, when displaying "intrinsic" Tinderbox uses SecondaryTextColor, not DisabledControlTextColor for better legibility.

Moved TbxLinkRenamer’s work off the main thread, as it can be slow in very large Tinderbox documents.

8.8.0

HIGHLIGHTS

NOTE: *Tinderbox 8.8 requires macOS 10.12 or later.*

Find

After selecting a note in the Find window, if the note cannot be selected and previewed in the current view, Tinderbox now displays a preview of the note’s text.

Drag any note from the Find window into a map or outline view to make an alias of that note at the drop position.

Find now offers suggested related words as possible autocompletions in the Find bar and in torn-off Find windows. These are available only in macOS 10.14 and later, and only in selected languages.

Search and Replace

The search results popover now contains controls to replace all occurrences of a pattern with a designated string. Note that this is not undoable. Note that replace *only* replaces occurrences found in the text — not in the Name or user attribute.

NOTABLE

Actions

.replace now understands back-references. For example, the action:

```
$Text=$Text.replace("From: (.*@).*","——$1——");
```

will replace

```
From: Mark@eastgate.com
```

with

```
—— Mark ——
```

.replace, when applied to $Text, now allows style operators to be applied to the replacement argument. For example,

```
.replace("^From: .*",$0.bold)
```

will embolden all lines beginning with "From:".

The operator **$Text.replace(pattern,replacement)** now updates text link positions.

A new style operate **.strike**, operates like **.bold** and **.italic**, but adds strike-through to the styled text.

The operator **wordsRelatedTo("term")** tries to return a list of up to seven words related to its argument. These are often synonyms and inflected forms, but may also include common adjectives and other related words. Fewer words may be returned, or no words may be returned if the argument is unusual, or if the current language isn't supported.

The operator **.containsAnyOf(*aSet*)** is true if any of the words in a set of words is contained in the target string. For example,

```
$Text.containsAnyOf(wordsRelatedTo("emulate"))
```

Will be true if the text contains the word "aspire." .containsAnyOf is always case insensitive.

The action operator **version()** returns information about the version of Tinderbox. With no argument, is return a complete version string such as "8.7.2b468". Optional arguments can be **major**, **minor**, **fix**, and **build**, returning a specific part of the version. For example, **version(major)** returns 8 in Tinderbox 8.7.2.

Applescript

Attempting to call evaluate() where there receiver was either the document or the root note of the document formerly crashed. Tinderbox now performs the requested evaluation as expected.

The expression

```
get the local attributes of TheNote
```

returns a list of attributes for a note where (a) the note has an immediate value for that attribute, rather than inheriting a value from a prototype or the attribute default, and (b) the attribute is not intrinsic, hidden, or deprecated.

The document property **selected note** now operates as it should.

Applescript: Revised the **make new** command to more cleanly create new agents, adornments and notes, and to return the correct value. Note that the returned designator records is based on the outline position of the newly created note, and subsequent calls that make or delete notes might render it invalid.

Scripts can now access link types.

```
linkType named "agree" in front document
```

Returns a link type by name.

```
linkTypes in front document
```

Returns a list of link types.

```
tell document 1
 make new linkType with properties {name:"name", color:"green",bold:true}
end tell
```

Creates a new link type. If a link type with this name already exists, no new type is created and the properties are applied to the existing link type.

Link types make also be deleted in scripts. The expression:

```
delete value of (attribute of theNote named "Width")
```

removes any local value assigned to that attribute. This is equivalent to the Tinderbox command `$Width=;`

The expression:

```
act on theNote with "action"
```

performs an action on the designated note. An action is typically one or more assignment or conditional statements, such as **Color="red"**. **Act on** does not return a value. The expression

```
evaluate theNote with "expression"
```

returns the result of evaluating an expression. In an action, = means "assign"; in an expression, = means "comparison" (although the unambiguous operator == is preferred.)

The expression

```
refresh theNote
```

informs Tinderbox that changes have been made to a note and the user interface may require updating.

Attributes

The new General attribute **CreatedFrom** is used by ziplinks to store the full path of the note that created this note. $CreatedFrom is not inherited from prototypes.

The set attribute **DisplayedAttributes** now controls the tables of *displayed attributes* that appears in the text pane. The old attribute **KeyAttributes** continues to be available as a synonym, but the new attribute name may prove less confusing. Related changes include DisplayedAttributesFont, DisplayedAttributesFontSize, DisplayedAttributesDateFormat, and HideDisplayedAttributes.

A new intrinsic set attribute, **LocalAttributes**, holds the set of attributes for this note which have a local value — a value that is not inherited to taken from the default. Attributes that are computed (like WordCount) or intrinsic (like Xpos) are never local attributes.

Changing an attribute value in the displayed attributes table or in **Get Info: attributes** now updates **Modified**. Changing an attribute value in a stamp (including Quickstamp) or an action does *not* update **Modified**.

Explode

When running on macOS 10.14 and later, Explode is substantially smarter about recognizing sentences. It now understands, for example, that "Dr. Perkins paid $10.00 to the U.S. Treasury." is one sentence, not five. The new AI-based sentence recognizer is available for several common languages; for other languages, or on older systems, Tinderbox fall back to the previous method.

Get Info

The map pane of Get Info has been rewritten to use Apple's MapKit API.

Hyperbolic View

In the background of Hyperbolic View, only the limit circle is drawn.

Link Parking Space

The **Note ▸ Park Link** command (⇧⌘-L) is now available from map, outline, and other view panes.

Map View

Dotted, dashed, and bold linear links are now drawn in the same style as their curved counterparts.

The **Arrange** submenu is now available from from map item contextual menu.

Outline View

When writing in an outline, the sequence [Return][Tab] now creates a new, untitled note, indents it if possible, and leaves that note open for you to enter its name. Previously, Tinderbox expected you to enter then name and *then* press [Tab] to indent the note.

Renaming notes

When renaming notes, Tinderbox finds and updates text links that link to the renamed note and are anchored to the note's former name. This is often useful when working with [[ziplinks]].

Roadmap

Better keyboard navigation in Roadmap popover. Tab to move between lists; spacebar to follow a link; blind type to select a source or destination.

Text

Format ▸ Style ▸ Highlight ▸ yellow highlight gains a keyboard shortcut (⇧⌘-Y). The highlight commands are now toggles; if the selection is already highlighted, the highlight is removed.

Watch Folders

Watch folders should now interpret [[ziplinks]] in text files, creating new notes as needed and linking them to the watched notes. To disable ziplinks in a note, set $Ziplinks to false. To disable ziplinks throughout a watch folder, set $Ziplinks for the folder to false.

Ziplinks are also interpreted when text is pasted in the view pane.

Ziplinks

If a ziplink creates a note, the note's **CreatedFrom** attribute is set to the path to the note that created that note. This allows OnAdd actions and agents to access properties of the note from which the new note was created.

When starting a ziplink with a backlink, [[< displays the same menu as [[displays.

Ziplinks showed a predilection for creating notes in the upper right-hand corner of adornments in maps with adornments. This should no longer occur.

DETAILS

Attribute Browser

Right-clicking an attribute browser group heading displayed a menu to "Open in new tab…". Because groups cannot be opened in a tab, this accomplished nothing; the menu no longer appears.

The header now displays the count of notes in the scope of this attribute browser. If the same note appears in multiple groups, the note is counted separately in each group.

Attributes

When using **make new attribute**, if a user attribute already exists with the designated name, the existing attribute is modified. If a system attribute already exists with the same name, no changes are made and no attribute is created.

Corrected a problem that could setting an attribute's default value to fail.

Bookends

Notes create by ⌘⌥-drag from Bookends now display the imported citation in the TextFont, not Helvetica.

If an imported reference from Bookends is not valid unicode, Tinderbox now rejects the import rather than risk a corrupt value of $ReferenceRIS.

Browse Links

The **Browse Links… popover** could not be dismissed when invoked from a text window. It can now be dismissed by clicking outside the popover or by pressing Esc.

Built-in Prototypes

The built-in prototype "Imported From Finder" is now wider, so the full note is visible.

Color Schemes

When changing color schemes, Tinderbox does a better job of recognizing text foreground and background colors that need to be revised for the new scheme. Differences in color representation and color spaces make color comparison tricky.

Displayed Attributes

After using the Values pulldown menu in the Key Attributes table, the chosen value is now formatted to reflect whether it is inherited or not.

Display Expression

If an item with a Display Expression was open for editing its name, clicking inside the editable text field could improperly reevaluate the display expression, in some cases leading to unwanted duplication.

Document Settings

Many labels and tooltips have been changed to reflect the shift from "Key Attributes" to "Displayed Attributes".

Find

The search field in the Find bar displayed its insertion point in the $Color of the selected note. We now use the current textColor.

Flags

In flags, the characters **>*+.$%}** have a special significance; for example, the flag **red$blue** is a red and blue checkered flag. If the special character appears as an annotation, however, it is now treated as a normal character. The flag **+.red** draws a plus sign on a red field.

HTML Export

In HTML Export, an ^else^ clause that began with a parenthetical expression mistakenly considered the parentheses to be an argument to `^else^`. For example, `^if($Subtitle)^...^else^(none)^endIf^` returned **endIf**.

Infrastructure

Several methods that check resources on the internet have been updated to use better APIs.

Addressed an intermittent crash when closing documents, caused when a pending throttled update was attempted after the view was released.

Attribute menus now show deprecated attributes as struck through.

During save and autosave, Tinderbox concurrently processes the styled text of every note in order to speed up saving. This could, with bad luck, lead to collisions in resolving each note's text font. We now serialize resolution of NodeFont objects.

Addressed a potential deadlock that might arise when importing watched folders.

Inspector

Many labels and tooltips have been changed to reflect the shift from "Key Attributes" to "Displayed Attributes".

Map View

In adornments, the grid widget could be difficult to see if the adornment itself was sufficiently light and neutral. In this case, a darker color is now used to draw the grid widget.

Containers and Agents once again respect the $Fill attribute by filling the title bar rectangle.

Prototypes

The prototype menu no longer refuses to set prototypes if the prototype name has a non-breaking space. In French, it is conventional to place a non-breaking space before some punctuation marks, including **?** and **!.** As a result, non-breaking spaces are common in note names *en français.*

Simplenote

Simplenote syncing has been discontinued. Simplenote account credentials no long appear in Document Settings.

8.7.1

NOTABLE

DEVONthink Watch Folders are again available in the **File ▸ Watch** menu.

Watch ▸ DEVONthink now uses the DEVONthink group picker.

Watch folders are now checked more often.

Watch folder updates could fail if the first item checked didn't need to be updated.

PDF documents dragged from DEVONthink are now imported as styled text if possible. Previously, they were imported as images.

A new menu command **Format ▸ Font ▸ Code Font** changes the font of the current text selection to the font designated by $CodeFont. The default value of $CodeFont is currently Courier New.

When switching tabs, the scroll position was not always consistently restored.

Text windows no longer scroll to the top when activated. Instead, the scroll to make the selection visible. If there is no selected text, the text window scrolls to its position when it was last activated.

DETAILS

When a Boolean attribute is selected in the displayed attributes table, pressing the spacebar will toggle its value.

If an adornment is added inside an agent, the agent's $CleanupAction is set to "none".

The Find Bar no longer leaves animation artifacts when dismissed from a top-level map. The animation has been improved.

HTML Export: link markup such as ^basicLinks^ and ^outboundTextLinks^ now exports relative to **current** rather than **this**, facilitating work with included files.

The **Done** button in the **Find Bar** again works as expected.

The Reset button in the **displayed attribute table** removes the viewed note's immediate displayed attribute settings, so that the inherited or default displayed attributes will be used. Formerly, it could copy those key attributes as an immediate value, temporarily interfering with inherited changes to the prototype or default.

Notes created when typing a ziplink now observe document settings to expand vertically or horizontally if additional space is required for the name.

Notes created when typing a ziplink are now positioned to avoid adornments as well as other notes.

Displayed attribute characteristics such as enabling or disabling the URL and File buttons are now updated after editing the key attribute's textual value.

Changing the text of a key attribute immediately updates that key attribute's pulldown value menu.

Text thumbnails in map view are now set tight, without an added line spacing.

Tinderbox generates text thumbnails for map views asynchronously in order to make map scrolling smoother. Occasionally, after Tinderbox requested a new thumbnail for a note, it might proceed to recreate or delete the entire view before the thumbnail creator has finished its work. The thumbnail creator would then try to store the thumbnail in a structure that was no longer extant, causing a crash. Now, when the thumbnail creator has completed its labors, it asks the map view where to store its result.

Cutting a passage in a text window that contained text links could create a race condition as separate threads sought to put the link on the pasteboard and to erase it.

Shrinking the text pane could cause layout problems with the Links pane lists, even if the links pane was hidden. To avoid this, the lists have a minimum width and the Suggestions list slides offscreen if there is insufficient space.

8.7.0

HIGHLIGHTS

COVID

A new operator **covid()** returns information about the covid outbreak.

> **covid(zip code, date, keyword)**

Returns the number of cases, deaths, and recoveries reported that day in the US country that contains this zip code. For example

```
covid("02148", date(4,1,2020), "cases")
```

Returns the number of cases reported in Middlesex County, Massachusetts. The keyword argument may be

> cases
>
> deaths
>
> recoveries
>
> name (the name of the county)

A four-argument variant allows you to query results by US state:

```
covid("MA","US",date(4,1,2020),"recoveries")
```

Reports the number of recoveries reported for that date from Massachusetts.

Data are as reported by the Johns Hopkins Center For Systems Science and Engineering, and are provided by CovidNearMe.org and are provided strictly for educational and academic research purposes. Data are copyright 12020 Johns Hopkins University.

INSPECTOR

Added Read Only, Searchable, and Ziplinks to the Name Inspector's Text pane.

The **Outline** tab of the **Appearance Inspector** gains a checkbox for displaying the note's outline color swatch.

The **Query** pane of the **Action Inspector** gains a popup menu button for changing the agent's cleanup action.

The HTML Inspector has been renamed the Export Inspector.

The **sequential** checkbox in the User Attribute inspector was not effectual.

NOTABLE

ACTIONS

$CleanupAction, which controls automatic arrangement of aliases inside agents, adds support for values of "row" and "column" in addition to "grid", "box" and "none".

Adjusted path interpretation in actions. If a path contains slashes or parentheses, we now check for note names that exactly match the path. For example, if the path is **S/Z**, this matches either the note Z inside container S *or* the note named "S/Z".

BROWSE LINKS

Browse Links… now lets you drag links to reorder them.

COMPOSITES

When a note joins a composite, its moves in the outline to become the younger sibling of the last member of the composite. Thus, all members of a composite are adjacent in outline order. The relative order of items within a composite should no longer change.

FIND

The Find Bar now appears after you press ⌘-F from the view pane. If the Find bar is already visible, ⌘-F moves the keyboard focus to the Find bar's search field. Esc dismisses the Find bar.

LINK PANE

Selecting a note from the **inbound** list in the **Links pane** now previews the source of that link. Selecting a note from the **outbound** list of the Links pane now previews the destination of that note.

Dragging the text preview, opened when previewing notes using the Link pane, creates a new text window for the note. Simply clicking the text preview continues to dismiss the preview.

The **Links** lists are now empty when more than one note is selected.

MAP VIEW

In map view, double-clicking the body of a locked container now zooms into the container. Formerly, a new note was created.

In map view, pressing down-arrow no longer zooms into an adornment

TEXT

In the text pane, ⌘⌥-down-arrow moves to the next note in outline order, but ⌘⌥-up-arrow failed to move to the previous note in outline order.

ZIP LINKS

Ziplinks can now create backlinks from the destination note. [[]] will create a link to "that note", and then will append the name of this note to the text of "that note" and link the name back to this note.

When typing a zip link, Tinderbox now prefers to link to the original note rather than an alias.

When typing a zip link, pressing [tab] selects the option currently selected in the popup menu. If no item is selected but there is only one item in the

menu, pressing [tab] selects it. If there are several items in the menu but none is selected, [tab] selects the first item.

The text preview window now has a label that explains how to dismiss the window or drag it to create a new text window.

The hint panel in the text preview pane now slides out of view after three seconds. Moving the mouse over the text preview displays the hint again, as does previewing a different note.

DETAILS

ACTIONS

Creating a note inside a container with an OnAdd action invoked the OnAdd action twice.

The recent revision to ensure that creating a note calls its OnAdd action exactly once created a race condition, where the OnAdd action could be invoked before the note was fully created. We now wait (in the NodeCreateAction constructor) to ensure that the OnAdd action, which must be performed on the agent queue, is in fact performed at a time when the note is valid.

The **delete()** operator now accepts lists of designators to be deleted.

The delete() operator now return **true** if at least one note was deleted, and **false** otherwise.

Tinderbox could crash on Catalina if creating a note triggered a delete() action on a note that is currently visible's, because delete() recycles the note and Catalina requires part of the recycling to be performed on the main thread. This is complicated because neither the HTPath nor the current context will exist when the asynchronous deletion is performed, and so both have to be copied and captured by the asynchronous block.

COLOR SCHEMES

Color scheme files (.tbc) place in the user's **color schemes** folder were not correctly listed in Document Settings.

DARK MODE

When changing to dark mode or light mode with Tinderbox open, the text color of the tabs is updated immediately, rather than when Tinderbox is relaunched.

GET INFO

In a Get Info popover that has not been torn off, selecting from the attribute pane's search menu autocomplete dismissed the dialog. To avoid this, the popover behavior is now semi-transient, requiring you to press Esc or click in a pane other than the view pane to dismiss the popover.

HTML EXPORT

Link lists such as ^outboundLinks^ now give their arguments access to the link source and destination keywords. Previously, the evaluation contexts used for these markup elements lacked this necessary context.

HYPERBOLIC VIEW

In some cases, the hyperbolic view did unnecessary work in response to frame change notifications during startup.

IMPORT

Text files dragged from Finder failed to import their text if dragged into a container with an OnAdd action setting the prototype.

INFRASTRUCTURE

Corrected a thread race in Base64Decoder, used when reading documents.

Changed the way the map view's quad tree — used for guides, composites, and much else — is updated in the background, avoiding potential crashes when closing documents while a complex map update is underway.

Identified a tiny memory leak in drawing the small note icon in outlines and other views.

Identified a small leak when building the shared prototype menu.

Outlines and charts do less work to keep track of link information they don't require.

INSPECTOR

The "sticky" button in the Quickstamp Inpector's **More…** pane is disabled when the selected note is not an adornment.

The Inspector's window position was restored incorrectly because the initial height of the Inspector window needs to take into account the size change caused by the Appearance pane.

KEY ATTRIBUTES

In key attributes, the File button now remains enabled even if the note is read-only. The **Browse File…** menu option is not available for read-only notes.

In key attributes, the URL button now remains enabled even if the note is read-only.

After a key attribute is edited in such a way that an inherited value is replaced by an immediate value, the key attribute table value for that attribute is now emboldened to reflect that it is immediate.

MAP VIEW

Corrected a situation in map view draggy-hand scrolling where the map thought it had reached the edge of the view when it had not.

Draggy-hand scrolling in views now updates the position of the scroll bar thumb.

$InteriorScale no longer impacts the font size of text thumbnails in map view.

When switching a tab to map view from another view, links are revised immediately. Previously, revision could be deferred to the next change of selection.

OUTLINE VIEW

Toggling View ▸ Use Checkboxes failed to update the outline properly because it was not informing the individual map items that they needed to change their appearance. In addition, toggling this menu while editing an item's title could crash because the procedure to end the editing session failed to end the editing session.

Corrected a hang when creating a note in outline view when the corresponding map view contains a geographic adornment.

QUICKSTAMP

Quickstamp layout is more robust in the face of lengthy inherited or default values.

TEXT

A few seconds after editing a note's text, the note will be reindexed.

Selected text in the text pane is again draggable in Catalina and Mojave.

TOT

Built-in prototype “Imported from Tot” no longer sets TextFont.

When watching Tot notes, the font used for the note text is the $TextFont of the “Imported From Tot” prototype.

Refactored NodeCreateAction; no significant changes should be observed, though pasting into non-map views may choose different (and better) map locations for the pasted notes.

Tinderbox Help failed to export a contents page to Tinderbox 7 release notes.

Revised ValueCell’s updateAndSelectNextRow to avoid reentrant updates, which Catalina prohibits.

8.6.1

NOTABLE

LINKS PANE

Fixed additional issues with making link suggestions while the index is being updated.

MAP VIEW

Fixed a crash in map view when zooming in twice in rapid succession.

TEXT WINDOWS

Text windows, having no Links pane, disable **Window ▸ Links**

TOT

Address a crash when watching Tot if Tot is not installed.

ZIPLINKS

When using the popup menu to create ziplinks, Tinderbox sometimes failed to update the position of other links that have anchors after the new zip link.

Ziplink length was limited to 50 characters, which is far too short. It is no longer limited.

8.6.0

HIGHLIGHTS

LINKS PANE

A new **Links Pane** may be added beneath the text pane. To see the a list of inbound and outbound links and a list of suggested links from the current note, choose **Links** (⌘**-7)** from the **Window** menu. Choose **Window ▸ Links** again to hide the links pane.

When the links pane is visible, double-click any inbound or outbound link link to follow it and select its source.

Select a link and press [Delete] to delete the link.

Click any suggested link destination to preview its text. Pressing [Esc] will cancel the links pane text preview. Double-click to add anew text link to that note. Or, select some text in your text pane and drag a link to any link suggestion to make a text link.

TOT

Tinderbox can now watch notes from IconFactory’s new application, Tot, for casual notes. To watch Tot notes in a document, choose **File ▸ Watch ▸ Tot**.

The new Tot container will initially watch all seven Tot notes. You can watch fewer notes by editing the container’s **$Tot** attribute.

Tot has a free Macintosh client and syncs automatically with other Macintoshes as well as iPhone and iPad clients.

ZIPLINKS

Ziplinks facilitate rapid linking to old and new notes while brainstorming and writing *zettelkästen* or speedy notecards. In older versions of Tinderbox, they were called “quicklinks”.

To make a link to a named note, precede the name of the note with two brackets and follow its name with a two brackets

> [[like this]]

If there is no note with this name in the document, Tinderbox will create one as the sibling of the note you are editing.

If you wish to make a ziplink to a note in a different container, you may enter the path to the destination in the ziplink:

> [[/container/note]]

The source text of the new link anchor is typically the name of the destination note, but you can choose a different anchor text to be replace the [[bracket]]:

[[note|explanation]]

You may also set the text of the quicklink note:

> [[note|explanation::see this]]

If the destination already exists, the new text is appended to the note.

As you are typing a ziplink, Tinderbox will show a list of related note names to the left of the text pane. Click on any of these notes to make a link. Alternatively, press the up-arrow or down-arrow keys to select the note you prefer, then press [Return] to make the link.

The ziplink popup menu displays possible ziplink destinations as you type.

[[shows a list of all siblings of this note. Containers are bolded.

[[/ shows a list of all top-level containers

[[/D shows a list of all top-level notes that begin with a “d”

[[/Dog/ shows a list of all notes in /Dog

[[/Dog/R shows a list of all notes in /Dog that begin with “r”

[[D shows a list of all notes with names that contain a “d”

The tooltip for each item in the ziplink popup menu displays that item’s full path.

The Ziplinks popup menu now takes into account your locale, so *étude* will sort appropriately for French users and Ångstrom will sort correctly in Norway.

To disable ziplinks in a note, set $Ziplinks to false.

Ziplinks are always created using the link type ***untitled**.

NOTABLE

Actions

$Flags is now an intrinsic attribute; it is not inherited from prototypes and an alias may have different flags than the original.

Attribute Browser

In attribute browser, dates are now formatted as specified by $KeyAttributeFormat. Previously they used the system Medium format.

Crosstabs

The agent query formed by the crosstabs action menu was improperly formed for string attributes, if the attribute’s values contained multiple words or if unlimited bins were selected.

Export

$HTMLExportExtension is no longer a preference attribute, so its default value is now settable.

Find

Results from the Find popover can be dragged into a text pane. The display name of the dragged item will be inserted into the text and linked to the corresponding note.

Prototypes

When assigning a note to a new prototype, the note’s initial height and width are set to match the prototype’s height and width ***provided that the note’s dimensions have not already been changed*** from the default. If the note has been resized, the note’s dimensions are retained.

Text

When a note without key attributes gains a key attribute, or when its last key attribute is removed, the visibility of the key attributes disclosure triangle is updated immediately.

Show/Hide Key Attributes departs the View menu and now appears exclusively (and nightly!) on the Window menu.

Within the text pane, you can now drag text containing text links and move the links along with the text.

The attribute $ReadOnly is exempted from being read-only, even true. This allows you to turn off ReadOnly from the key attributes table or from **Get Inf**
‣ Attributes.

In key attribute tables, date values are again updated as soon as they are edited. The method previously used proved incompatible with Catalina.

Views

The view pane now understands non-ASCII characters for locating items by blind typing.

DETAILS

Actions

If the user’s medium date format began with the day of the week in English, then Tinderbox recognized the day of the week before trying the system parser. Now, the day of the week is processed specially only if the system parser fails.

Agents

Agents that find numerous notes and apply changes to their appearance could trigger numerous screen updates. These now perform better.

Chart View

In chart view, the selection highlight is less subtle.

Hyperbolic View

The menu choices for **View ‣ Hyperbolic** and for **Hyperbolic** in the tab contextual menu are now validated to ensure that the selection is not empty and has at least one non-prototype link.

The tab title for a hyperbolic view is no longer empty if the hyperbolic view lacks a focus.

Get Info

Get Info ‣ Words results were skewed because the indexer was including values of attributes other than user attributes.

Infrastructure

Addressed a data race when activating the new suggestions pane after Tinderbox had been in the background.

Addressed a data race when activating Tinderbox and rebuilding the prototypes menu.

Fixed a layer instantiation issue that caused the background view to be drawn in black in macOS 10.13 and before.

Map View

Pressing [space] is again a shortcut for switching focus from a selected map item to the text pane.

When becoming a map view, scroll to make sure the selected note is in view.

Changing the shape of a note, either in the Appearance inspector or using the note’s contextual menu, is now undoable.

Drag scrolling in maps is smoother and more accurate.

The map pane now autoscrolls when dragging a link outside, but near, the map pane. Autoscrolling is not performed when the mouse is near a parking place. Autoscroll in map views has been revamped, and should now be much nicer — especially on fast machines.

In map views, when a note is dragged out of the view pane and then is dropped back into the map, Tinderbox no longer makes a new copy of the note.

Instead, Tinderbox moves the dragged note to a new map position.

Outline View

Outline drag highlights are now clearer.

Improved text wrap measurement in outlines with columns.

In outlines, untitled notes are no longer autodeleted if they have outbound links, or if they have inbound links other than prototype links.

In outlines, when a note is dragged out of the view pane and then dropped back into the outline, Tinderbox no longer makes a new copy of the note. Instead, Tinderbox moves the note to a new place in the outline if possible. If the dragged note is dropped on one of its own descendants, the drag is cancelled and the document is unchanged.

Views

The activation animation for the view pane is now performed when the selected note is reselected; previously, it was performed only when activation changed the selection.

ZipLinks

The built-in prototypes for Code and HTML have $ZipLinks set to false.

When creating a ziplink, press [Tab] to complete the ziplink if (a) there is only one remaining choice in the quicklink window, or (b) an item in the quicklink window has been selected with the up-arrow and down-arrow keys.

When typing a ziplink, if there is only one choice in the ziplink menu you can choose that option by pressing [Tab].

When typing a ziplink, we ignore diacritics when looking for matches.

The vertical placement of the ziplink menu is now relative to the text being edited, if the menu can reasonably be placed at the position.

8.5.0

HIGHLIGHTS

CROSSTABS

The Crosstabs View, now available in the View menu. (Shortcut: ⌘⌥-B), lets you explore the values of two selected attributes in your document, or in a portion of your document. It displays a table in which one attribute's values are listed horizontally and the second attribute's values are displayed vertically.

Crosstabs view is ideal for exploring relationships among two attributes. For example, you might explore which products sell best on different days of the week, or which students performed poorly on the midterm but well on the final exam. Crosstabs can also examine co-ocurrence in sets and lists; a crosstabs of $Tags could reveal, for example, which notes are tagged both with "Italy" and "Urgent".

SHARED PROTOTYPES

Tinderbox now lets you add your own built-in prototypes, helping you create new documents more easily.

Your own built-in prototypes are stored as top-level notes in a Tinderbox document named **Prototypes.tbx** in subfolder "prototypes" of the Tinderbox support folder.

```
~/Application Support/Tinderbox/prototypes/Prototypes.tbx
```

These prototypes will appear at the bottom of the Built-In Prototypes menu. Choosing one of these prototypes will add it to your document's Prototypes container.

If the prototypes container already holds a prototype with the chosen name, the command has no effect.

The new prototype shares most attributes with the prototype in the chosen support folder. Exceptions include:

- User attributes that do not exist in your document
- Intrinsic attributes such as Xpos and Container, that are never inherited
- Readonly attributes

NOTABLE

Dragging From Views

You can now drag notes out of Tinderbox views and drop them in other Tinderbox documents, in other applications, or on the desktop. When dragging to a different Tinderbox document, Tinderbox will make a copy of the dragged note and will copy the dragged note's attributes to the copy. Attributes not defined in the destination document will be ignored. If a note is dragged to another application, the note's text and styled text are available to that application.

Import

DEVONthink import now uses a dedicated built-in prototype, rather than setting key attributes directly. This allows you to customize Tinderbox's behavior when importing from DEVONthink 3.

Text

Added green and magenta highlight commands to the Format ▸ Style menu.

DETAILS

Actions

Owing to some obsolete code, values(*targets, attribname*) failed to return any values when the targets expression was a find() expression.

Catalina

Tinderbox is better at understanding whether it's in light or dark mode when opening new documents, if you have changed appearance settings after opening Tinderbox.

Worked around a Catalina limitation that could crash Tinderbox when changing the key attributes table.

Read-only key attribute values are now drawn in a more legible color when selected and when using dark mode.

Drags from Bookends are now coerced to TextColor, which improves interoperability in dark mode.

Color Schemes

The built-in color schemes now explicitly set CaptionColor to blue, with the exception of Dark Coral which sets CaptionColor to “lightest blue”.

Dragging from Views

When dragging notes out of Tinderbox maps, we return the dragged note to its original position.

Dragging notes out of outlines and charts is now supported.

Autoscroll is more restrained.

Filtered Outlines

Filtered outline sometimes failed to filter in recent builds.

Find

Torn-off Find windows no longer try to insist that their selection remain selected even after you select a different note.

When selecting a note in the Find window, Tinderbox once again highlights the matching text in the text pane.

HTML

The export tag ^url^ used the extension specified in $HTMLLinkExtension rather than the $HTMLExportExtension.

Infrastructure

Additional work (alas) on the Action Recycling Pool. Each document now has its own pool, and actions are always recycled on the agent thread; this should prevent some highly-intermittent crashes when Tinderbox is activated from the background.

Key Attributes

If a note is ReadOnly, Tinderbox now disables editing of key attributes and removes the key attributes value menu.

Maps

In map view, when tabbing between title, subtitle and caption of a note that is displaying a text thumbnail, the thumbnail would sometimes be placed above the title when tabbing from caption back to the title. This arose because tabbing to empty fields clears the title cache, and Tinderbox did not always recomputed the title cache value.

When creating notes, if the selected note was part of a composite, Tinderbox could incorrectly make the new note part of the composite, or create it beneath the composite.

Adornment OnAdd and OnRemove actions were not always applied, due to an over-enthusiastic drag optimization.

Outlines

In outlines, separators with titles are more conservative in estimating the vertical space required for their display names.

Outlines mistakenly allowed space for borders, which disturbed title placement for notes with broad borders.

Prototypes

When a note adopts a prototype, the note’s Height and Width are initially to match the prototype. Note that Height and Width are not inherited; this change affects only the initial dimensions of the note.

Roadmap

In the Roadmap, pressing **delete** will delete the selected link.

Stamps

Stamps can once again be dragged from the Stamp pane of the Document Inspector to Finder windows or the desktop.

Text

The text pane of an inactive window now accepts mouse clicks. Previously, the text pane followed the traditional Macintosh behavior of treating the first click as a request to activate the window. The change is useful when switching between a torn-off Find window, Roadmap, or the inspector, or when moving between documents.

Views

When editing the name of a note, Tinderbox did not consistently replace the display name with the name.

8.2.3

DETAILS

Actions

When dragging a note to a new contained, the OnRemove action of the destination container was applied instead of the OnRemove action of the note’s previous container.

OnLink actions could crash, because a Node that ought to have been captured in an asynchronous block was an instance variable.

Introduced a new format code **“n”** (for “normal”), which displays the system medium date string and short time string.

isort($Name) could crash if the list was not a list of paths

Attribute Browser

Attribute browser is more conservative about updating in the background, working around a complex crash at startup involving a complex attribute browser.

In attribute browser, dates are now formatted using the medium system format. Previously, they were formatted as ISO-8601 dates.

Links

Undoing a note deletion in outline view could leave an unwanted MapItemView onscreen.

When dragging a link to the map background in Catalina, Tinderbox opened an editing session to name the newly-created note and this session remained open if the popover was dismissed without changing first responders. As a result, a renamed note could retain its Untitled state and be deleted. We now set the map view as first responder.

Extensive revisions to the link creation popover to clean up logic and rectify errors during cancellation.

Maps

The Edit Background… popover for the map view now uses symbolic names for colors when possible. Previously, it considered most colors “custom”.

Fixed a crash when resizing or moving items in complex maps with composites, where asynchronous composite updates were interfering with access to the composite for drawing.

Miscellaneous

Added additional protection against starting an agent query after the document has begun close itself.

Moving full-screen Tinderbox into a split-screen environment could lead to a crash when activating Tinderbox, because the LayoutInfo for the note clicked to activate the window might have been deleted when the window was resized.

Outline

Clicking in an outline name field whilst editing again sets the insertion point.

Text

When editing dates in the key attributes table, the date string is changed before editing to its value in **normal** format (medium date, short time). This avoid ambiguity in short dates, where the default US style uses 2-digit years. Does 12/7/41 represent a date in 1941 or 2041?

8.2.2

DETAILS

Actions

The numeric function rand() failed to assert that it returns a numeric value, leading to incorrect behavior in complex expressions.

The attribute **MapTextSize** has been renamed **MapNameSize**, since there is lots of other text in the map view. Similarly, **OutlineTextSize** is renamed **OutlineNameSize.** The old names are deprecated but will continue to function.

Import

When importing a **vcard** by dragging from Contacts or elsewhere, Tinderbox places a copy of the vcard text in $Text to allow you to extract specialized fields unique to your work. In some cases, however, the vcard includes a base-64-encoded image, which isn’t useful in the text. That data is now removed from the imported text.

Key Attributes

The width of the first column of the key attribute table now adjusts itself to accommodate long attribute names.

The height of the key attribute table can now grow to as much as half the height of the text panel, reducing the need to scroll long lists of key attributes if using a sufficiently-large display.

In the Key Attributes Table, ⌘⌥-down-arrow now selects the next note, and ⌘⌥-up-arrow now selects the previous note, just as they do from the next pane.

Links

When creating notes by dragging a link to the map background, if the name of the note has been edited and is neither “untitled” nor the empty string, then [Return] no longer needs to be pressed to create the link and new note.

When selecting text links in the **Browse Links…,** Tinderbox could fail to highlight the source of the link in the text pane and could fail to update the link properties if several windows were open. Other windows, attempting to highlight the selected link, could throw an exception if the text span to be highlighted did not exist.

Changed the text link highlight from thick to double-underscore, because thick highlighting was unsatisfactory with some fonts.

Fixed a crash when deleting links via the link delete widget in map view, caused by an asynchronous update to the link view.

Tinderbox could sometimes refuse to create a new text link in a note that posessed “smart links” such as URLs embedded in the text.

Fixed a potential crash when deleting links in the text pane when they are displayed in the map view.

Maps

Mousewheel/pinch zooming is somewhat smoother in map view.

In map views, the contextual item menu allows you to create notes and adornments. This is especially nice when working in maps with large adornments.

Outlines

Fixed a rare crash when expanding or collapsing a note, because we were using the view’s LayoutInfo after updateGeometry had an opportunity to scavenge it.

when adding a new item, if the name occupies two or more lines, Tinderbox now extends the space for the new item immediately rather than waiting for the next layout update. This is tricky, because the element growth is animated and we can’t expand the components of the view until the animation is complete.

Miscellaneous

Cleaned up code in Roadmap.

Addressed a crash during shutdown that could arise if the agent queue was deleted while an agent was evaluating its query.

Addressed a crash when switching tabs, arising because we were trying to animate a transition even though we explicitly requested no animation.

Re-addressed a crash when switching tabs, caused by animating the presentation of notes after the view had switch to a different view.

Fixed a rare crash during shutdown if a queue we require no longer exists.

Selecting a new note during an open editing session could crash Catalina with a warning about NSTableView reentrancy.

Hyperbolic view no longer offers a non-functional Find panel.

Stamps

runCommand applied as a stamp would only run on a single note, even if several notes were selected, because an asynchronous block failed to capture the action

Reordering stamps in the Stamps pane of the document inspector threw an exception in Catalina because it failed to clear the pasteboard. This

eliminates, for now, dragging stamps to the desktop or to Finder windows.

8.2.0

NOTABLE

Actions

Actions: adding an interval to a string formerly returned an interval. This was not very useful, and contradicts adding a number to a string

```
"The answer is:" + $MyNumber
```

Which returns a string. The result of applying a binary operation where the left-hand side is a string and the right-hand side is an interval is now a string.

Added an attribute **$Twitter**.

Attribute Browser

Separators no longer appear in the attribute browser.

Hyperbolic View

Separators no longer appear in hyperbolic view.

Links

Links now have an optional *comment* that can be used to explain or clarify the purpose of the link. Comments may be view and edited in **View ▸ Browse Links…** or in the torn-off variant of **View ▸ Roadmap**.

Link visibility can now be controlled for individual links as well as through link types.

Parking space popovers now accept either the name of a destination node or the destination URL of a web link.

The source of the parked link is now drawn more legibly, especially in dark mode.

Link visibility rules have been modified. If a link type is not visible, all links of that type are hidden. If a link type is visible, then individual links may be hidden or shown.

Maps

Maps: the inbound-stub menu and the outbound-stub menu now show the display names of linked note, prefixed by the display names of their containers. This may help disambiguate situations where it's convenient to have several notes with identical names, or where we link to different aliases of the note.

Roadmap

Roadmap now allows editing of the properties of the selected link.

Stamps

The Stamp menu now supports hierarchical organization of stamps.

If a stamp's name contains a colon

```
Color:red
```

then Tinderbox places the stamp in a submenu named "Color". This lets you organize a family of related stamps without creating unwieldy menus. (Stamps cannot be nested more deeply.)

Text

Added a Highlight submenu to the text pane contextual menu

DETAILS

Actions

When creating an adornment, the container's OnAdd rule was applied too soon, before the adornment object was fully instantiated. As a result, the OnAdd did not know that the object would ultimately be an adornment. If the OnAdd set the adornment's prototype, the prototype did not know that the object would ultimately become an adornment and proceeded to bequeath children to the future adornment, which cannot possess children.

Export

In 8.1.1, Tinderbox cleared the agent queue when saving. That could cause delays when an autosave takes place during an HTML Export.

Infrastructure

Extensive engineering to more efficiently use your Macintosh's numerous processors.

Addressed a longstanding, intermittent crash involving rules or agents, typically encountered either when Tinderbox was in the background or when activating Tinderbox. The problem involved clearing the action recycling pool while one of the recycled actions was still extant and in use, either by a rule running on a different thread or by an agent. Agents no longer cache actions, and all evaluations are now performed on the agent queue — even those involving user-interface updates such as updating the key attributes table.

The action recycling pool could occasionally recycle an action while the agent manager was still using it. To make sure the left hand of darkness knows what the right hand is doing, we now hand the action to the agent work queue to recycle, since if the agent queue is available for to do this it cannot simultaneously be using an action.

Changes in managing the sort queue (which has now been abolished) and the view queue (which serializes access to the list of updatable views, and which now becomes a private implementation detail of TbxViewManager).

Revised the indexer (again) to clean up queue usage.

The attribute browser now updates its model in the background, rather than doing the work asynchronously on the main thread. This should make it more responsive.

When switching windows in Catalina, Tinderbox could crash if the activated window had been deactivated while a key attribute value was open for editing.

Added a privacy item to the plist to support Calendar drags in Catalina.

Removed obsolete attributes RSSChannelTemplate, RSSItemTemplate, and RSSItemLimit.

Resolved a crash when deleting a link from a note to the same note, caused by an asynchronous update that ran after the animator had tried to draw

the deleted link.

Fixed a crash when using

[list].sort(*attribname)*

when the list items are not, in fact, paths to notes in the document. Revised the underlying code and data structures for performance when sorting long lists.

Fix a crash in Get Info : Words arising because the new asynchronous indexing facility invalidated our iterators while we were updating the word frequencies.

TanseyTab's **willDelete:** method used an inefficient approach to checking whether deleting a note should also delete the tab. This doesn't matter in small documents, but could be very costly in large documents, especially those which have high-turnover agents that contain numerous notes. We now use NodeFinder's far more efficient index.

DateValue resolved IS8601 strings asynchronously, which saved time but created problems when we resolved the DateValue because we need to use the string at once.

Revised analytics proxy to prevent initialization of a session if the user has opted out.

$EstimatedNoteSize was unreasonably slow, because it inadvertently typeset the text as a side-effect.

Maps

The layout of map items did not correctly account for the size of badges, which could lead to unwanted truncation of names.

If using the Document Settings option to **use smaller type** if a note name is too long in map view, Tinderbox will no longer make the type larger than the default size if the note is later made larger. Adornments are no longer affected by this document setting.

Outlines

After changing the outline scale, Tinderbox continued to use cached measurements, preventing appropriate resizing of outline elements.

Roadmap

At the top of the Roadmap, Tinderbox shows the display name of the note for which inbound and outbound links are being shown. The tooltip of this name is now the full path of that note.

8.1.1

NOTABLE

Actions

$EstimatedNoteSize now provides a rough estimate of the space that the note will require on disk.

Outline

The selection highlight in outline mode is now more visible, especially when using the document settings Black Titles or White Titles.

Speed

Change management has been revised to limit the rate at which Tinderbox attempts to update its layout, especially in response to agent updates which can trigger numerous layout requests as agents add and remove aliases. The result should be a significant performance boost for rules and agents in large documents, without visible degradation elsewhere.

Avoided repeated updates to the outline order when reading documents, saving substantial time.

Text

⌥-esc now displays autocompletion choices in the text pane.

DETAILS

Corrected some Chinese localization.

Extensive revisions make the action parser much less likely to crash in the presence of syntax errors. The parser remains very permissive, but some formerly-permitted errors are now detected. Of particular note:

- Single-argument operators like **sin()** now require an argument, instead of silently assuming an argument of zero.
- **collect_if(), sum_if()** and **avg_if()** are now invalid if their test is invalid.
- Some failed parses are now detected and marked as invalid. For example, the expression

```
$Width!==3
```

is now invalid; formerly, it was interpreted as `$Width!= "="` .

Changed the policy of the ChangeManager. All change handling is now performed on the update thread.

Color schemes are now saved with the current text background color; previously, this was omitted. Color schemes also now save and restore the current text color, and the default value of $Color.

The built-in prototype for a code page is now named HTML rather than HTML Template.

When closing a filter using the close button (but not the Use Filter menu), the outline was left with an incorrect indentation, and newly-added items failed to align with other items.

In map views, after zooming into a container (down-arrow), the first responder and keyboard focus is left on the map view, facilitating further keyboard interaction (such as up-arrow).

Addressed a crash when attempting to perform an unparseable AgentAction.

Additional measures to increase action robustness in the face of unexpected failures.

The Document Settings window title again contains the document name, which has been absent since Tinderbox 7.5. Welcome back.

An interesting race condition during reading could intermittently update $Modified to the time at which the document was opened.

8.1.0

NOTABLE

Tinderbox is now significantly faster on more powerful machines, especially when opening and saving documents and when selecting numerous notes. Adding notes to a large outline view is approximately three times faster. Reading documents with numerous notes is about 40% faster. Dragging notes in map view is also much faster.

Geographic Adornments are adornments that depict maps. An adornment is a map adornment if all of the following are true:

- it is sufficiently large
- its latitude and longitude are not zero

The latitude and longitude may be set directly, or may be computed indirectly because $Address has been set. The scale of the map is determined by the adornment's $Range attribute, which represents the approximate size of the map in kilometers.

If a note lies on a map adornment and has a latitude and longitude, it will be placed near the corresponding place on the map adornment.

We use a new geocoder to deduce the latitude and longitude.

Internationalization

First steps are under way to support Chinese versions of Tinderbox menus and interface elements.

Actions

substr() and **.substr()** sometimes mishandled Unicode sequences. Both now operate as expected for both ASCII and two-byte Unicode code points.

New list/set commands .**any** and **.every** let you test each item in a list or set.

```
$MyList.every(x,x>5)
```

is true if every element in $MyList is greater than 5.

```
$MyList.any(x,x>5)
```

is true if any element in $MyList is greater than 5. The comparison may also be applied to lists of strings:

```
"apple; pear; plum".every(x, x>"aardvark")
```

Is true because every element follows "aardvark" in alphabetical order.

If the target list or set is empty, **.any()** always returns false, and **.every** always returns true.

unlinkToOriginal and **unlinkFromOriginal** are now offered as autocompletions in code fields.

The two-argument form of the stamp operator now takes the designator as the first argument, consistent with other operators that take an argument. **stamp(designator, name)**. The one-argument form **stamp(name)** operates as before.

.remove() removes items from a list or set. For example,

```
$MyList.remove(0)
```

returns a new list from which all elements equal to zero have been removed, and

```
$MyList.remove("cat")
```

returns a new list from which call elements equal to "cat" have been removed.

```
$MyList.remove("cat; dog; badger")
```

removes cats, dogs, and badgers.

Attribute Browser

Attribute browser uses less liberal line spacing.

Attribute browser no longer defaults to DisplayName as a sort option, as this causes severe performance problems in large data sets. Instead, it defaults to $Name.

OnAdd and OnLink actions no longer fire when a document is opened.

Attributes

New attribute, **Range**, in the People category, represents the approximate range of a geographic display, in kilometers, for use by Geographic Adornments.

$AutoFetchCommand is now an action attribute rather than a string.

Dark Mode

Many small adjustments improve legibility in dark mode.

Outline icons are more legible in dark mode.

Find: highlighting the found string in the context of the find view is now more legible in dark mode.

Document Settings

When changing the default text font in **Document Settings: Text**, Tinderbox now updates existing all font runs where the text font family matches the old default, and replaces them with the new text family. For example, if changing from Mercury to Ideal, passages set in Mercury Italic will now be updated to Ideal Italic.

Filtered Outlines

In filtered outlines, range-select (e.g. shift-click) and **Edit ▸ Select All** selected notes that were filtered out of the outline. Notes that are hidden by the filter no longer are selected.

Find

The search menu in the view pane's find bar now offers an option for case-sensitive search, and an option to turn off regular expression search.

Links

The **Create Link** dialog suffered layout problems when the source or destination note name was exceptionally long. The name is now truncated as required.

In outline view, link stubs are no longer drawn to represent invisible links.

Map View

Some shapes, notably clouds, were hard to drag because they mishandled clicks in the note body but outside the title area.

Text Export

Text export of selected notes now exports those notes in outline order. Previously, it exported notes in the sequence in which they were selected.

DETAILS

A late update to Catalina broke Tinderbox's approach to handling text links. This update reimplements text links in a manner compatible with macOS Catalina as well as previous versions of macOS.

Addressed a hang when working with two main windows, when adding notes in the front window while the back window has an outline view, because the update was computing heights for the outline while the new note was locked.

The title of the Document Settings window is now "Document Settings"; it had been "Preferences" since the Flood if not before.

The command **Note ▸ Open Text Window** is now available in hyperbolic view.

When using macOS 10.14 or later, the pane activation highlight uses the user interface accent color, rather than the blue-green guide color.

View panes reflect changes to color scheme immediately, rather than when their parent next changes.

In map view, dragging an adornment temporarily brings it to the foreground to ensure it is plainly visible. On mouse up, Tinderbox now restores it to its proper layer; previously, this might be deferred until the tab was clicked.

The hyperbolic view's axes and boundary are drawn to be less obtrusive in dark mode.

The link parking space's "contents" label, which shows the source of the parked link, is easier to read in dark mode.

Took measures to guard against crashes in **.format** and **collect** under unusual error conditions.

The activation highlight in the view pane now uses the correct rectangle, even at startup. Previously, it failed to account for animation in progress during window assembly. The activation highlight in the text pane is now drawn without unwanted clipping.

Fixed a crash when comparing two dates, if the right-hand date expression was not parseable.

Revised error management methods to separate management of the error list from instantiation of the view controller, since the view controller must be instantiated on the main thread but the document, which owns the error manager, might not be.

Cleaned up font cache code, and avoided potential concurrency problems when adding or removing views.

Increased the size of the action recycling pool, which should reduce mishaps when running edicts on emergence from background.

Reorganized the bottom of the Tinderbox Inspector's Agents & Rules pane to make it more clear that the update timer refers to network updates rather than the edict cycle.

Manual agent updates are now enqueued on the agent queue, avoiding potential conflict when they are invoked while an agent update is under way.

Removed a layer in the scroll pane for the key attributes table TextPaneController.xib ; this may account for some inexplicable refresh issues in the key attributes table.

In the Tinderbox inspector's Rules and Agents pane, slowly-changing facets like the number of outstanding edicts are now updated periodically.

Fixes a hang when saving from outline view, arising because changing the document name prematurely forces a deadlocking outline re-rendering.

Prevented a hang when adding an agent in outline view, caused by a premature layout update.

Disabled ISO8601 caching, a performance improvement that may be leading to incorrectly-saved date attributes.

Fixed a deadlock when creating built-in templates in outline view, arising when layout updates were recalculated recursively. Fixed an intermittent crash in documents with more than 5000 rules, often triggered by updating agents manually or bringing the document to the foreground.

Scripted unlinkToOriginal could crash because the change manager tried to update text links whilst the link caches were in an inconsistent state. The parser failed to recognize unlinkFrom, though it handled unlinkFromOriginal correctly.

Documents with flawed xml now open as Untitled, avoiding a worrisome (though fortunately rare!) situation where Tinderbox could overwrite the damaged file with an valid but empty file.

Fixed a crash when pasting notes, arising when Tinderbox failed to wait for concurrent processes to finish before disposing of the underlying XML parser data.

Fixed a deadlock when creating built-in prototypes in outline view, arising when layout updates were recalculated recursively.

Fixed a very intermittent crash during revert or when opening a document, where the old hypertext might try to process pending changes after its deletion.

Corrected some display issues in Treemap view, related to previous revisions to improve performance of link drawing. There ain't no rest for the wicked.

Reduced the frequency of rule updates slightly when Tinderbox is active, and more significantly when Tinderbox is inactive.

Selecting multiple notes is now much faster because the text display system performs less unnecessary work.

Found and fixed several memory leaks, notably one that failed to dispose of closed TbxMaps.

We do less indexing when typing, saving some processor cycles. This should save a little energy when running on battery.

Improved the efficiency of Tansey Tab drawing, and standardized the truncation of tab labels.

Corrected an update problem in map view that affected redrawing after cleanup.

Corrected a crash when opening multiple documents at the same time.

Adjustments to natural language tagging, which should slightly improve overall performance on 10.14 and 10.15.

The text in the text pane shader, which is displayed when no note is selected, no longer has a shadow.

The neural network behind NLTags detection became very slow when dealing with large quantities of Chinese text, which it does not understand. We no

longer attempt to tag non-English text.

On macOS 10.12 and later, Tinderbox respects time zones in dates saved in the Tinderbox document. On earlier systems, Tinderbox continues to assume that the time zone is the user's local time. Some anomalies might be encountered in dates before 1BCE.

On macOS 10.14 and later, we encountered a memory leak because one of the natural language framework objects appears to leak internally, and we were creating these profligately. Instead, we now share one plan tagger for all tasks; this should also help restrain load when opening the document.

Removed Simplenote support, though some vestiges (such as the Simplenote attributes) remain to avoid problems with existing documents.

The natural language analyzer was not run on note with empty text. This is wrong, since a planning note might have the plan in the title ("get some bagels on the way home") and no text.

8.0.4

NOTABLE

Actions

The operator **drivingTimeTo(*path*)** estimates the driving time between the location specified by **this** note and the location specified by the note *path.* This operator is available on computers using macOS 10.12 and later. The estimated driving time is zero if either note's location cannot be determined, if no route can be found between the two locations, if the operating system is too old, or if the internet is not accessible. Otherwise, the result is a time interval estimating the approximate driving time, based on traffic conditions prevailing when the action was invoked. Note that this function can be slow; consider using an edict rather than a rule, and avoid recalculating the driving time if you already know it.

```
if ($MyInterval==0) {  $MyInterval = drivingTimeTo(/places/favorite/Swarthmore));  }
```

The new action **distanceTo** computes the approximate distance in kilometers between two notes for which $Latitude and $Longitude are known. For example, if **this** note has the latitude and longitude of Boston and **Paris** has the latitude a longitude of Paris, then distanceTo(Paris) is about 5582.

The behavior of accessors for time intervals has changed. For example, suppose $MyInterval holds an interval of 5:30 — five minutes, 30 seconds. `$MyInterval.minute` now returns 5 and `$MyInterval.second` now returns 30. Formerly, `$MyInterval.minute` converted the interval to the number of minutes, returning 5.5, and `$MyInterval.second` returned converted the interval to the number of seconds, returning 330.

When setting $Text in an action from an unstyled string, Tinderbox now uses the note's $TextColor rather than black.

Attribute Browser

The name column of the attribute browser now uses the same font and type size as used in the outline view at standard magnification, allowing you to choose the font size.

Attributes

The new attribute $DominantLanguage records Tinderbox's guess at the primary language used in the text of each note. Language analysis is only performed on macOS 10.13 and later. Languages appear as two-letter ISO-639-1 codes such as "en" for English, "de" for German, and "zh" for Chinese.

A new category, Places, now contains attributes related to geography. Most of these were formerly in the People category.

A new built-in prototype for Places is now available.

Scripting

Tinderbox now lets you evaluate an expression.

```
evaluate note with expression
```

The first argument *note* is an AppleScript specifier for the note, which will be bound to **this** for the evaluation. The **evaluate** command may be issued to either the document or to a specific note.

Nodes now possess a property, **links,** which is a list of all outbound basic and text links from that note. Links are read-only and have three properties: the source note, the destination note, and the path name.

The application now has a read-only string property, **build,** that represents the build identifier of the application.

The Applescript construct

```
Make new note with properties {name: "..."}
```

now works as expected.

The **category** of renamed and deprecated attributes is now returned as the category of their replacement.

DETAILS

Reduced number of indexing threads that may be active during document loading.

The HTML Export progress bar now shows the full path of the note being exported.

The natural language processing system now takes advantage of improved models and performance when running on macOS 10.14, while continuing to operate as before on older versions of macOS.

Corrected a longstanding potential crash when closing large documents. Agents periodically delete aliases that no longer meet their criteria. Those deletions must occur on the main thread, not the agent thread. In stressed cases, the pending deletion task could be deferred until after Tinderbox has closed the hypertext and deleted the alias, leaving the pending task to delete a note that had already been deleted.

Removed disused setting to underling WikiLinks from the Text pane of Document Settings.

ChangeDocket's copy constructor could crash because it copies an internal data structure without respecting the objects' synchronization queues.

The attribute AIM has been removed, as has the service.

Fixed a long-standing intermittent crash at startup or when activating a complex Tinderbox document.

In scripts, attribute creation and renaming frequently failed because the attribute index was not correctly managed by AppleScript.

In AppleScript, the type of an attribute is now its **type** rather than its **kind**.

In AppleScript, when creating new attributes, Tinderbox now accepts the **properties** construct:

```
make attribute in doc with properties {type:"list", name:"fox", defaultValue: "hen"}
```

The Attribute Browser is now more legible in dark mode.

The Export Progress bar is now legible in dark mode.

The link editor now shows the source and destination names rather than their display names, because editing the names renames the notes in brainstorming mode.

The user attribute inspector's name field provides validation for proposed names, but the validation logic assumed that normal text is black, creating problems in dark mode.

When viewing a export template note, there's little point to using the HTML or Preview panes. When a template becomes the selected note, we now automatically shift to the text pane.

Note ▸ Go Back no longer hoists the view if the destination note is already visible in the current view.

8.0.3

NOTABLE

New documents created in Dark mode use a dark theme.

Tinderbox anonymously reports some simple usage statistics to help us make features easier to discover, and to help us understand what languages and which versions of macOS are actually in use. These are very general features, such as what kinds of views you use or how frequently you delete a tab. No identifying information and no information about the contents of your notes is transferred. Care has been taken to ensure that this won't impede you when you're offline or when the server is busy. You may opt out at any time in Tinderbox Preferences, but we hope you'll share this information to help us improve Tinderbox.

Scripts can now create new user attributes.

```
tell application "Tinderbox 8"
 set doc to document "xtest.tbx"
 set a to make attribute in doc
 set the kind of a to "date"
 set the name of a to "myNewAttribute"
end tell
```

The **kind** of attribute determines the attribute type, and may be any of the following values: string, color, number, file, boolean, date, set, URL, list, interval.

Note that references to attributes specify attributes by name, and so existing references are invalidated after the attribute is renamed. You can get a new reference to the renamed attribute thus:

```
set a to attribute named "theNewName" in document "theDoc.tbx"
```

DETAILS

Corrected a logical error that sometimes set WhiteOutlineTitles automatically in dark mode.

Improved text color updates when applying color schemes; difficulties previously arose when color schemes redefined "black" and "white", or did not define the color used for the text color.

Registered Tinderbox users can opt out of analytics reporting in **Tinderbox ▸ Tinderbox Preferences**.

Inspector code fields and key attribute value fields now adopt an insertion point color that matches the text color, rather than insisting on black.

Fixed an exception thrown on macOS 10.11 when performing a deferred reindexing; on old systems, we simply reindex right away.

Added an entitlement for calendars, since some users aren't getting prompted to allow calendar access.

Removed a disused *lowFidelity* mechanism from the link animator.

HTMLView adapts its colors to dark mode.

The Badge Picker now adopts the prevailing background color in dark mode. Monochrome icon families are drawn in white over a dark background, and in black over a light background; color families are drawn normally. One would scarcely credit how much trouble this entailed.

Adornment OnAdd actions are again performed as expected.

Separators now respect their indentation level, and leave space for the expand widget.

Corrected a layout problem in centered charts that could lead to overlapping notes.

In dark mode, the text pane's title field is now drawn in white if the note's body color would provide insufficient contrast, just as in light mode the title is black if $Color is too close to white.

Dark mode improvements for Tansey tabs, text pane shader, and browse links.

The attribute browser control panel is more legible in dark mode.

Added a menu option **View ▸ Tab ▸ Close Tab** to close the current tab.

8.0.1

NOTABLE

Tinderbox again launches on macOS 10.10 and 10.11.

DETAILS

Actions

Changing $StartDate, $EndDate, or $DueDate marked the note as changed in appearance, causing unnecessary work after rule updates.

The Rule and Edict inspectors' Run Now buttons now update the key attributes table.

Attribute Browser

In Attribute Browser, clicking the Query button displays a popover. Clicking the button while the popover is displayed now dismisses the popover; previously, it dismisses the popover and then opens a new popover, which is inconvenient.

Color Schemes

Corrected saving of default $TextColor and $TextBackgroundColor, especially if these have been changed by applying a color scheme.

Dark Mode

The filter bar is less colorful, but now is legible in dark mode

The calendar icon denoting dates in the key attribute table is now visible in dark mode.

The icons for the text inspector and the Export inspector are easier to see in dark mode.

The badge popover now has a gray background in both light and dark mode, avoiding trouble when drawing monochromatic icon families.

Find

When tearing off a FIND window, the states of the aliases and case-sensitive buttons are copied from the current settings in the popover.

Help

Revised and reindexed.

Hyperbolic View

After renaming a note in the text pane, the hyperbolic view is now updated to reflect the new display name.

After changing the color or shape of a note, the hyperbolic view is now updated.

Import

Scrivener import is much faster for large documents. Tinderbox was checking for composites after each note was imported, and doing extra work to ensure that each new note did not overlaps with other notes. We can short-circuit much of this unnecessary work.

Infrastructure

More indexing is performed on the agent queue, which should reduce intermittent, unreproducible crashes at startup of larger documents.

The reindexing system is now better protected against situations where Tinderbox rules force too-frequent reindexing, as well as situations where avoiding constant reindexing causes changes to go unnoticed indefinitely

Revamped the process of shutting down a document when closing it, to ensure that pending operations are either cancelled or completed before we demolish the underlying structures. This should reduce crashes when quitting or closing documents with lots of actions.

Tinderbox could crash when changing views, if text thumbnails were still in the process of creation. We now wait for pending thumbnails to be created or canceled before deleting LayoutInfo records.

Maps

If a container's title bar was pulled down to the bottom of the container, the line marking its bottom was not properly clipped to the confines of the container shape.

If several notes were moved in the map view, the sibling order of those notes was changed on mouseUp. The sibling order no longer changes.

In map and hyperbolic view, after dragging a link to the background, Tinderbox creates an untitled note and lets you name it and assign a link. Now, if you change your mind and don't name the new note, the newly-created note and its link are automatically deleted.

When making links in map view by dragging links to the background, the view pane for the hyperbolic view could appear inappropriately if the hyperbolic view had been previously viewed in another tab. This happened because the hyperbolic view updates itself when it is not hidden after links are created, and the view did not understand it had been hidden.

When moving notes in map view, it was possible for a note to form a composite with a separator, though the separator does not appear in map view.

Cleanup: since the staggered button only applies to grid layouts, it is now disabled when other layout methods are chosen.

In map view, when pressing Return to create a note with another note selected, if the note would be created on an adornment, Tinderbox moved the newly-created note off the adornment. Tinderbox now allows Return to create the note on the adornment, adjacent to the selected note.

In map view, the body text is again scaled proportionally to $InteriorScale.

In ap view, increased the length of time you must hold the mouse down without dragging to edit the note title.

Outlines

Minor change to CeresOutlineSurveyor, which sometimes failed to allow sufficient vertical space for the full outline title.

After using the ⌘⇧-down-arrow shortcut to Focus or Hoist a note in outline view, the selection and keyboard focus is restored to the hoisted note.

Text

Cutting text that contains a text link no longer throws an exception in Mojave.

Separators with text more accurately accommodate outline leading.

After switching to HTML or Preview in the text pane, clicking on a note activated its title editor without selecting the note. This should no longer happen.

Views

The timestamp commands Format ▸ Text ▸ Insert Date ⌘-/, Format ▸ Text ▸ Insert Time, and Format ▸ Text ▸ Insert Date and Time ⌘⌥-/ are now available when editing note titles.

Watch Folders

Tinderbox could crash when a watched folder contained a pdf document that the system cannot read, or that is not actually a pdf.

8.0.0

HIGHLIGHTS

Hyperbolic View

An entirely new way to explore your notes, and a great tool for brainstorming.

Filtered Outlines

By adding a filter to any outline tab, you can focus exclusively on notes that meet your criteria.

Improved Maps

Greatly improved performance lets you build larger maps than ever before. New color schemes give you fresh options and a clean new look.

Quicklook

Tinderbox now lets Finder preview your documents.

Scripting

Tinderbox is now scriptable, giving you new ways to collaborate with other tools and to streamline your workflow.

NOTABLE

Actions

Group designators are now allowed in attribute references. For example:

```
$MyList=$Colors(children)
```

finds a list of the colors of each child of **this** note, and

```
$MyNumber=$Width(children).max
```

will find the maximum width of the container's children. When applied to the attribute $Text

```
$Text=$Text(children)
```

the texts of each child are appended, separated by paragraph breaks.

Normally, actions and stamps require an action — an assignment, or an if(), each(), or var() statement. Actions may now be expressions, which are evaluated for their side-effects. For example, `"frogs".speak()` is technically an expression, not an action, but now speaks.

create

The expression

```
create(path)
```

creates a new note at the designated location, and returns the full path to that note. If the designated note already exists, no new note is created and the operator returns the empty string. The designator is typically a complete path

```
create("/hardware/faucets")
```

A two-argument variant:

```
create(/path/to/container, name)
```

allows you to separately specify the container's path and the new note's name; this may be useful if you need to create several notes in the same container, or if the path is most easily specified relative to this note

```
create(child,"name of new note")
```

createAgent

Creates a new agent.

```
createAgent(/agents/urgentTasks)
```

```
createAgent(/path/to/container,urgentTasks)
```

delete(*designator)*

Deletes the designated note. If the designated note does not exist, the expression has no effect and return false. If the designated note exists, it will be deleted and the expression returns its former path.

Avoid using this operator when possible. It can automatically delete notes you intended to create, and it can potentially saw off the branch you are standing on. In almost all circumstances, it is better to move the unwanted note to a container, and then to delete the note manually if you really need to delete it at all.

if

Assignment to an inline if() expression is now permitted. For example:

```
$Color=if($MyBoolean){"red"} else {"blue"}
```

When practical, the more conventional and idiomatic style is preferred:

```
if($MyBoolean){$Color="red") else {$Color="blue"}
```

But, since if() does return a value in expressions, this usage is now explicitly allowed. Note: this usage has been allowed for some time if the if() expression was enclosed in parentheses.

isDuplicateName

The action isDuplicateName is true if another note in the document has the same $Name as **this** note.

lookup tables

Lookup tables have been extended to permit ranges. For example, suppose the value of $MyString is

`Alaska-Connecticut: 1; Delaware-Nebraska: 2:default: 3`

Then $MyString.at("Alabama") would return 1, and $MyString.at("Illinois") would return 2. Lookup tables can also work with numeric ranges. If $MyString is

`0-10: red; 10-20: blue; default: green`

Then $MyString.at(5.0) return red. Note that $MyString(5) — without a decimal point — returns the fifth element of the list.

originalLinkedFrom/To

```
originalLinkedTo(designator[,link path])

originalLinkedFrom(designator[, link type])
```

Return true if the original note corresponding to this is linked to or from the designated note. This is especially useful in agents, where `this` may be bound to an alias owned by the agent but the user is interested in links to the original note.

stamp(*theStamp*)

The command **stamp()** applies a stamp.

```
stamp(designator, stampName)

stamp(stampName)
```

The designator may be any note or group designator. If the designator is omitted, the stamp is applied to this note. **stamp()** returns true if the stamp was found, and false otherwise.

.lookup()

The command **.lookup()** behaves like **.at()** but always uses a lookup table. Prefer **.lookup()** to **.at()** for clarity, and to avoid ambiguity when the argument is numeric.

```
$MyString.at(5) ← the fifth element of the list

$MyString.lookup(5) ← the lookup result for “5”
```

.sum()

The operator .sum adds up lists of numbers. If $MyList is 1;2;3;4, then `$MyList.sum` is 1+2+3+4=10.

Agents

A new menu command, **Create Agent As Child** , has been added to the Note menu.

AI

Tinderbox now extracts the first phone number found in the $Text and automatically stores it in $Telephone.

Automatic conversion of $Address to find latitude and longitude is back, at least for now. This may prove too costly, but we’ll give it a try.

Attribute Browser

The Attribute Browser now allows multiple selections for many purposes.

Added a new option to display the Count Fraction for each category, showing the category count along with the total count.

Cleanup

Cleanup has been completely rewritten. The old Cleanup submenu has been replaced by a more flexible cleanup sheet. Cleanup is now available both from the menu bar **View ▸ Cleanup** and from the note’s contextual menu.

Dancing (Force-Directed Layout)

A new control panel provides controls over the “dancing” force-directed layout algorithm.

- Link Length controls the distance between linked notes.
- Stiffness controls the tension of the spring between linked notes.
- Even unlinked notes attract each other at long distances. Notes also repel each other at short distances. Compactness controls the distance at which unlinked notes neither attract nor repel each other.
- Jostle adds some random motion to each note, and continues to add decreasing amounts of random motion for the next several seconds. The magnitude of the random motion can be adjusted from **gentle** to **firm**.

If more than one note is selected, only the selected notes will be moved by the Dance command. If no note is selected, or only a single note it selected, then all unlocked notes are moved by the Dance command. This is very convenient when arranging a cluster of related notes in a large map.

Find

Results may now be sorted by outline order, creation or modification date, or by name.

Get Info

Repetition: This Get Info pane gains a button that lets you copy the list of repeated words and their word counts to the pasteboard as tab-separated data. The current scope and sorting are used.

Links

OnLink actions: LinkTypes can now define an OnLink action that will be performed whenever a new link of that type is created. When running the OnLink action, **source** is bound to the link’s source note and **destination** is bound to the link’s destination note. The designator **this** is also bound to the source note.

Maps

Links and guides are now drawn in their own view, which is placed above all adornments and beneath all other notes. One would scarcely credit how much effort this required.

In previous versions, extremely large maps, typically maps viewed at significant magnification, could fail to display links because macOS couldn’t believe anyone would need a view that large. Tinderbox no longer requires such a large view.

A performance issue in rendering and scrolling map views arose when a note contained numerous large images. All the images were rendered and

resized for the thumbnail, although it is most unlikely that any but the first image would be even partially visible. Tinderbox is now much smarter about undertaking this work, and performs the chores asynchronously.

Option-scrollwheel magnification is gentler and easier to use.

When creating links in map view or hyperbolic view, releasing the mouse when not pointing to a note will create a new note at that location.

Corrected the scroll position when holding down ⌘⌥^ to preview the full map, ensuring the full map is in fact visible.

Roadmap

The torn-off Roadmap window now has a button to allow you to refocus the roadmap on the selected note.

Stamps

Stamps may be moved between documents by dragging them out of the Stamp pane of the document inspector and dropping them into the new document's view pane. If the document receiving the stamp already has a stamp of the same name, the drag will be ignored. If the stamp refers to user attributes not present in the new document, actions involving those attributes will have no effect.

Stamps may be dragged from the inspector list to the Finder, where they become files with the extension **.tbxstamp**. Dragging a **.tbxstamp** to a new Tinderbox document's view pane will add that stamp to the document, provided that a stamp doesn't already have that name.

Templates

The built-in templates container is now required to be a top-level note; an interior note names "templates" will not be adopted as the built-in templates container.

Working With Other Tools

DEVONthink items dragged into Tinderbox now import their DEVONthink label string to a new attribute, DEVONthinkLabels.

Watch ▸ Folder From Finder: now imports each file's Finder tags into $Tags

DETAILS

Actions

The expression

```
(date("today")-$StartDate) + 1
```

Returned an unexpected value. The difference between two dates was treated as a date, but it should be treated as a number — the number of days between the two dates.

The expression **inside(container)** is now true if the note or any of its aliases is inside the container. Previously, it was true if the note was inside the container or **this note**, an alias, was inside the container. The change affects expressions where we want to know if a note is inside() several different containers or agents.

When cutting a note, the OnRemove action of its container is now applied before the note is copied to the pasteboard.

collect_if() now resets its regular expression match list for each note it tests. Thus, $1 will be the first matched subexpression for **this** note, rather than the first matched subexpression for the entire collect_if statement.

Fixed a crash when evaluating a **sort** expression

$MyList.sort(*expression*)

when the values of $MyList are not paths to notes, as expected.

When using **.replace** on text, Tinderbox could crash if multiple matches were found and some replaced text was near the end of the string.

Agents

AgentState::QueryHasChanged was revised to reduce or eliminate a race condition on quit. When closing the application, an agent might deleting its aliases because the query has changed at the same time that the the Document Closer was deleting those aliases because the document has been closed.

Autosave could reenable agents when agents ought to have been suspended. A possible but unlikely explanation of a cryptic crash.

Tinderbox is less eager to update view layouts when making changes to notes that are currently hidden. This greatly reduces computational load in documents that make heavy use of agents.

Newly-created agents are not longer flagged for having no query, since the user hasn't had time to add one

The default value of $AgentPriority is now 1, corresponding to "normal" priority, rather than 0, corresponding to "highest" priority.

Addressed a hang in map view, when an agent listed its own parent.

Attributes

The read-only system attribute **$Aliases** returns a list of paths to each alias of the original of **this** note. This information is seldom if ever necessary; if you find yourself tempted to use $Aliases, consider whether a different approach might be superior.

A new Boolean attribute, **HTMLMarkdown**, is used to allow processing of ^value()^ tags while not treating headings (## heading) as ordered lists.

When using the search field to locate an attribute in QuickStamp or System attributes, Tinderbox could choose the wrong attribute if several attributes share a common prefix — e.g. Caption and CaptionAlignment.

Attribute Browser

The color of the note's icon was not updated promptly when the color of the note changed.

The attribute browser now works better with ⌥-tab for switching between the attribute browser and the text pane.

Applies its action, if any, to every note that satisfies the view's query.

When displaying the text of a multiple selection from the attribute browser, the texts are displayed in the sequence in which they appear in the attribute browser. In other views, the texts appear in outline order.

The Attribute Browser now allows dragging notes into empty numeric categories, and into the first item in empty numeric and string categories. These drags were previously rejected without explanation.

In the attribute browser, clicking on the autocomplete menu in the query and action popovers no longer dismisses the popover.

Attribute Browser tab titles could use the wrong character encoding, creating trouble if the note name contained an umlaut or other non-ASCII character.

The attribute browser now has an attribute search field.

ColumnEditor needs to be semi-transient popover; if it's transient, clicking on the autocomplete menu dismisses the choice before it can be made.

Switch from an outline view to an attribute browser generated an unwanted beep in some situations because the first responder became the window and the window responds to a cancelEditing: operation with a beep.

Badges

Aliases in the user badge folder are again handled correctly.

The names of all Avatar badges are now lower-case for consistency with other badge names.

Browse Links

After deleting notes, we promptly revise the link view. In the previous test release, deleted notes might appear until the selection was changed.

Color Schemes

New documents now default to the Modern color scheme.

All color schemes have been revised and made more comprehensive.

Three new color schemes have been added: Green. Coral, and Dark Coral.

The Modern scheme now sets Darker Outline Colors.

Some older color schemes have been dropped. The retained color schemes are Franchi, Solarized, Sorolla, Standard, Storyspace, and Modern.

The action menu in the Colors pane of the Document Inspector now provides an option to **Save Color Scheme.** The saved color scheme files may be dropped into the view pane of any other Tinderbox document, or added to the color schemes folder in the Tinderbox support folder to make them accessible through Document Settings.

When applying a new color scheme, if the scheme changes $TextFont, notes that use the old text font will be changed to use the scheme's text font.

Dancing

Tinderbox was too conservative when starting force-directed layout from a highly non-optimal starting point, and in extreme cases refused to move any notes at all.

Document Settings

The checkbox for Darker Outline Colors did not always reflect the actual state of the preference.

Explode

When attempting to Explode At Delimiter, Tinderbox could hang if the regular expression matched a delimiter of zero length. Tinderbox now declines to attempt to explode at delimiters of zero length.

Eliminated a potential crash when working with an invalid regular expression

Corrected a failure to delete the delimiter in the final element, when the delimiter precedes the title.

Changing **Delete delimiter** failed to clear the chunked cache and so had no effect on the first explode operation.

Export

When using Markdown, Tags embedded in the text such as ^value()^ are now evaluated before being passed to Markdown.

The built-in prototype for Markdown preview now sets $HTMLParagraphStart and $HTMLParagraphEnd to "" in order to eliminate unwanted paragraph markup tags.

The default value of $HTMLEntities is now false. This should reduce confusion for those accustomed to using utf-8 in HTML.

The **Preview** pane gains a "No selected note" shader.

HTML and Preview pane shaders are now drawn in the correct shade of gray.

The built-in Markdown stylesheet is changed to fix an incorrect formatting command.

Export: when exporting text, toggling the "include titles" checkbox would display controls regulating the style of exported titles, although text export is unstyled.

Find

Fixed a problem in Find when searching Name or User Attributes but not searching Text, that caused all results to appear as empty section header cells.

The torn-off Find window now retains its selection when clicking on a note. Its title reflects the number of notes found in the document, and the number in the scope of the current view. Selecting a note in the torn-off find window will select that notes in the view, if the note is available.

Changed the highlight style in the Find window in order to avoid some unsightly artifacts. Added a count of matches to the display.

Get Info

Get Info does a better job of setting the initial focus when you select a pane.

The Repetition panel displays a progress indicator, as indexing large documents can be slow.

The Repetition panel's list of notes and occurrences is now sortable.

The Agent panel is now selectable only when the current node is an agent or an adornment.

Help

Added help for new features, **that** designator.

Infrastructure

Extensive changes to NodeIndex, a system that indexes (and reindexes) your document to speed up all sorts of operations. All this happens automatically in the background and seeks not to interfere with concurrent tasks that sometimes rely on the index, such as agent handling.

Tinderbox could crash — typically after running in the background for a long time — if several documents with rules or other actions were open simultaneously because each document might try to CloneNullValue() at the same time.

Edit ▸ Duplicate correctly updated the duplicated note's ID, but failed to reflect that update in the attribute $ID, which retained the ID of the original note.

Extensive revisions to facilitate asynchronous handling of agents, rules, edicts, and some time-consuming display tasks. Rewrote many queues to use a common queue factory.

Rewrote the animation for the text pane shader, possibly correcting artifacts when drawing the key attributes table.

Avoid a crash if Tinderbox is unable to resolve an alias in the user badge folder.

Inspector

Corrected text encoding of the document name in the inspector window title, and ensured that the window title is updated when the inspector appears.

In the Action Inspector, if a rule or edict is disabled, the Run Now button is also disabled.

In the Edict pane of the Action Inspector, the Run Now button is correctly positioned.

The Tinderbox Inspector no longer counts hidden attributes and categories as System attributes, since the user cannot see them.

Tinderbox inspector separately reports User and System attributes.

Key Attributes Table

Removed the grid lines.

Link Parking Space

The link-creating button in the link parking space popover now recognizes the Return key as a shortcut.

Maps

If a note begins with an image, that image is drawn on the face of the note. If the note's $NameColor is **automatic**, the color of the note is changed based on a sample color from the image, so a contrasting title color may be chosen automatically. (Previously, $Color was always changed, whether or not $NameColor was **automatic.**)

Modified note layout to improve typography.

When dragging items in the map, many different guides may compete with suggestions. Often, several guides may concur in the same suggestion: for example, when dragging a note near a note of the same size, guides the align the note tops, note midpoints, and note bottoms might all concur. Previously, Tinderbox drew all the guides, even though some higher-priority guides might overrule lower-priority guides. Now, when dragging items Tinderbox draws only the guides that will actually take effect.

Tabbing from the name field selects the subtitle field. Tabbing from the subtitle selects the caption field. Tabbing from the caption selects the name field.

Link positions were not immediately updated, and map guides were not immediately removed, after mouseUp when resizing a note.

The default MapBodyTextSize is about 20% larger than previously.

We now require the user, when dragging a link out of the link widget and *not* dragging to an existing note, to move at least 32px from the tip of the link widget before Tinderbox creates a new note.

Text thumbnails are now generated asynchronously, improving map performance when scrolling.

Substantially improved performance, especially when drawing containers and when scrolling.

The length of time required to hold the mouse down before editing a note name has been reduced from 1sec to ¼sec.

Notes linked from the current selection — the link neighborhood — are no longer drawn differently from other notes. These features were unloved and imposed a significant performance cost.

In map view, up-arrow retains the map scroll position of the view.

Changing the scale of map views could leave misplaced arrows in the view which remained until the next redraw. The arrow cache (!) was not cleared when the scale changed; now it is.

Changing the map title font size in the Text Inspector could clip descenders and temporarily misalign the title because stale cache information continued to be used for layout.

Double-clicking on a locked adornment created *two* notes.

The horizontal centering guide was miscalculating the desired position, leading to small misalignments when used any of the horizontal alignment guides.

Outlines

When a note is dragged in outline view, the dragged note might be placed before, after, or as a child of the note on which it is dropped. If the cursor is in the left third of the title, or to the left of the note title, the note will be placed before or after the target note depending on whether the cursor is in the upper or lower half of the target. If the cursor is to the right of this point, the note will become a child of the target note.

When adding a new note to an outline, or when renaming a note in an outline, Tinderbox should be about twice as fast to close the editor when you press Return.

Adding an item to an outline with many (>1000) siblings was slow because Tinderbox made O(n) rectangle comparisons to find a suitable map position. We now use a quadtree to reduce the number of comparisons to O(log n).

When dragging in outlines, notes are added as children if the drag position is (a) in the rightmost 2/3 of the note's title rect, or (b) more than 100px to the to the right of the left edge of the title.

When clicking on an outline item that uses columns, clicking on an unselected item selects that item. Clicking on a selected item opens the column for editing

Numeric columns use lining figures for greater legibility. The column rectangle is no longer displaced from the title baseline.

Tinderbox adopts a new approach to drawing its outline collapse/expand widgets, allowing the color to adapt when the outline background is dark.

Pressing ⌘⌥ generated an unwanted console message because updating the text pane links was being too aggressive in trying to force a prompt display update.

Outlines drawing geometry has been revised to allow descenders a little more breathing room at tight leading, and to give the icon, badge, and link widgets a bit of headroom if space allows

A crash occurred if the outline pane was resized whilst editing the name of a newly-created note, because Tinderbox won't recycle views with an editing session under way.

Outline items could occasionally overlap other items by one pixel, due to a rounding error when placing them. This became visible when the overlapping items had OutlineBackgroundColors, because the overlapping translucent overlays appeared as a darker line.

Outlines were computing their font size incorrectly, leading to truncated outline titles.

Corrected layout and font size of separators.

In outlines, moved the name field slightly to better align with the badge.

Adjusted handling of leading to prevent Tinderbox from allocating extra horizontal space as vertical leading increases.

In outlines which use checkboxes but not columns, Tinderbox now allows sufficient space for the badge.

Outlines: pressing tab while editing an item in the outline now closes the editing session before indenting the selected note.

Quit

Eliminated some (mostly harmless) crashes on quitting that arose because asynchronous change updates were being attempted on closed views.

Revert

Avoid a possible crash when reverting. Resizing the window to the previous window size forced redrawing the tab bar at a time when it referred to objects in the closed version of the document.

Revert should no longer crash when the Inspector is open.

Sorting

Sorting by date no longer regards dates within 1 minute of each other as equal.

Tinderbox is more attentive to the need to sort containers after changing a key attribute's value, changing an attribute value in Quickstamp, or in the attributes pane of the Get Info popover. This matters most when automatic agent updates are disabled.

Editing a column value will update sorting.

Changing the sort criteria in the Action Inspector will update sorting.

Tabs

Resolved a long-standing problem when hit-testing during hover over the Tab bar. The old method required a device context to hit test for the path, but sometimes no device context could be made. We no longer use the device context.

Text

After making a new text link using ⌘⇧-L and typing the destination name, Tinderbox updates the text pane so the new text link is visible.

Rules and Edict are updated when the text of a note is edited. Some care has been taken to ensure that this does not intrude on performance.

Corrected an indexing error that could cause pasted links to mishandled.

The animated transition removing the text pane shader (which appears when no notes are selected) is faster and so less obtrusive.

If the text pane is collapsed, Tinderbox no longer takes time to format text which you cannot see.

Tinderbox could unintentionally delete a note when the user intended to delete text. This occurred if the view pane was the first responder — that is, the text pane was inactive — if the user selected text by clicking in the margin of the text pane and then immediately pressed [Delete], because clicking in the text pane margin did not activate the text pane itself.

Option-clicking a text link inhibits following that link, allowing you to edit the anchor text. Tinderbox was mistaking the option key press used in the keyboard shortcut for **Browse Links** (⌥⌘-L) as an option key press, and the key release was (correctly) seemed to the browse links popover, not the text pane. This is now straightened out.

Complex drags into the text pane were sometimes ignored because they were not yet inscribed when the window was activated. The view controller's windowDidBecomeKey method no longer forces an update on activate if text work is pending.

Footnotes, if created in an auxiliary text window, could fail to create their outbound text link if the same note was selected in a different window.

The text link parking space's autocomplete now handles multi-word note names more appropriately.

The text link parking space's popover is now dismissed by pressing the Esc key as well as by clicking outside the popover.

Timeline

Timeline drags are again smooth, without distracting and confusing "flashing" of the dragged note.

Views

$NameColor may now be "transparent". Naturally, this renders the note's display name invisible, which is sometimes desirable. Use at your own risk: invisible things can be hard to see.

Fixed a rare crash when resizing a window causes Tinderbox to manage its views.

Working With Other Tools

The prototypes of watched notes now specify that watched notes are imported with a $TextBackgroundColor of white, even in dark styles, because external documents are most likely compatible with light background colors.

When watching a new folder from Notes or Evernote, the menu of folders is now sorted alphabetically.

Autofetching from DEVONthink rtf documents failed to update notes that had been viewed.

Tinderbox 7

Changes during Tinderbox v7.x.

Version 7 releases are listed below, in reverse chronological order.

7.5.5

MISCELLANEOUS

ACTIONS

$CleanupAction moves from the General to the Agent category.

The collect() family of operators has modified its behavior. If the collected attribute is a set or a list, collect() adds its elements to the result. If the collected attribute is not a set or a list, but contains a semicolon, quotation mark, or parentheses, the value will be added to the result as a quoted string. This should correct a variety of confusing cases.

ATTRIBUTE BROWSER

Attribute Browser categories are now sorted by locale; diacritical marks no longer create unexpected sort orders. When categories are sorted by the summary, ties are sorted by the category name. (2307, 2449)

EXPORT

In complex documents, HTMLView could fall behind in its update cycle. We now perform the update synchronously from a background thread, avoiding potential resource starvation. (2452)

Addressed a crash when switching away from HTMLView in documents that keep Tinderbox very busy. (2451)

If the preview or HTMLView had the focus and the user switch to the text pane, Tinderbox gave the focus to the Title editor rather than the text pane. (2450)

KEY ATTRIBUTES

Fix over-ambitious percent encoding when trying to ViewInBrowser.

The key attributes table's method updatePreservingSelection was not, in fact, preserving the selection. In consequence, after editing a key attribute, the first row was selected instead of the next row.

The View menu now allows you to Edit Key Attributes…

MAPS

When using a tab to switch from a different view to a map view, the scroll position of the map view was sometimes drawn incorrectly, and some notes might be treated as being offscreen even though they were, in fact, on-screen. This arose because TbxMap's setFrame:withOffset: method changes constraints on the map view but did not immediately update the layout; in consequence, the system sometimes found itself measuring the layout against the previous boundaries and origin of the map.

In maps, the command Expand Horizontally erred if the first word in the note could not fit in the available width without being broken, yet if the word were broken the entire display name would fit in the available width and height. Notes are now expanded so that, at minimum, the available width is sufficient to display the longest word in their $DisplayName. (2442)

Bold links are bold again.

Lozenge, diamond, and hex shapes choose the placement of their text more intelligently, and are more stable in placing text during resizing. (2441)

Right-clicking the link widget brings up a contextual menu listing inbound and outbound links for the note. (2382)

OUTLINES

In Outline view, the highlight has been revised to make it easier to identify the highlighted item at a glance.

When using columns, Tinderbox failed to account properly for the margins between columns, and therefore sometimes allocated too little vertical space for the note name.

In outlines, the selection highlight is emphasized by rules at the sides of the selected note.

MISCELLANEOUS

Fixed a crash when evaluating $DisplayName in an agent query, because TbxDisplayNameQueue returned the wrong queue.

The Inspector may now be opened and closed from text windows.

The DEVONthink importer no longer sets key attributes of the container for the imported notes has an OnAdd action, since that container may well be setting key attributes itself.

Sorting notes is now performed on the agent thread, which should alleviate intermittent crashes in documents that spend a lot of time running rules and agent actions.

Tinderbox resets its date translator apparatus when the time zone changes during a Tinderbox session. The objects were reset on the main thread, which could create trouble if a date calculation was in progress on the date translator's own work queue. The objects are now also reset on the work queue.

Avoid a potential crash that could occur when selecting an alias inside an agent which the agent is about to delete.

When reopening documents, text windows are opened on the main thread (as they must be) even when reading in the background.

Document Settings failed to recognize its tab bar after it had been closed and then reopened.(2426)

After a link type is edited in the Links pane of the document inspector, map views will update themselves even if no note is selected.

7.5.3

MISCELLANEOUS

ACTIONS

If collect() collects items that contain an ampersand &, quotation marks " or an open parenthesis, the item is now enclosed in quotation marks to prevent parsing problem with the resulting list. (2410)

Styled Text

Several string operations can now be applied to styled text from $Text, or destined to $Text, in order to modify the text style. Note that while any string may have its style set, this only has an effect on $Text.

When an action assigns the $Text of one note to another, Tinderbox preserves styling.

```
$Text = $Text(/configuration/example)
```

When the text of two notes are appended, Tinderbox preserves styling.

```
$Text = $Text(/a)+$Text(/b)
```

A string may be emboldened or italicized:

```
$Text=$DisplayName.bold
```

```
$Text=$MyString.italic
```

The font size of a string may be changed:

```
$Text=$MyString.fontSize(36)
```

The following operators now respect style information:

.replace , .substr , .paragraphs()

EXPORT

HTML Export is much faster, especially for documents that contain images or lengthy texts.

Avoided a deadlock in HTMLView that could stall the agent queue and cause memory congestion, and that could also result in Tinderbox displaying stale HTML and Preview data.

Reduced the memory footprint required when exporting by reducing the number of attributed strings we create, and releasing them more promptly when they are no longer needed.

FORCE-DIRECTED LAYOUT

View ▸ Arrange ▸ Dance is now undoable.

IMPORT

Fixed a crash when autofetching some DEVONthink items.

INFRASTRUCTURE

Fixed a deadlock, most likely to be encountered at startup, when several parts of Tinderbox contend to access attributes for indexing, for rule and agent initialization, and for display.

Addressed a crash when deleting a text link in map view and then copying and pasting the previously-linked text to a new note.

Agent expressions that manipulate DisplayName could deadlock and lock the agent queue. Moved DisplayName evaluation to its own serial queue.

Refactored TbxAttributeBrowserModel to move TbxAttributePicker handling from the model back to the controller.

Fixed a potential crash when trying to compute $DisplayName while Tinderbox is closing a document.

Fixed a startup crash in macOS 10.11, occurring when TbxTextViewActivationAnimation attempted to store a weak reference to a TbxTextView.

INSPECTOR

Arrow keys did not work as expected in the autocomplete of the search fields of the Action Inspector's Sort pane; the search fields failed to use TbxAutocompletingSearchField. (2409)

A new Appearance Inspector pane lets you set the outline background color.

MAPS

New notes created by double-clicking or right-clicking on an adornment were moved next to the adornment; Tinderbox now creates the note where you told it to create it.

The vertical and horizontal centering kibbitzers now have higher priority than the edge-alignment kibbitzers.

Fixed a crash when then first item in a note's text is a pdf attachment.

OUTLINES

Outline layout has been revised to use $OutlineBackgroundColor more attractively. Outlines now use $OutlineBackgroundColor to tint the background behind the entire note, not just the note title.

If editing a value in an outline column, clicking inside the edit field no longer reselects the entire text. Instead, the insertion point is places appropriately.

TEXT

Basic links, like text links, can link to a specific place in the destination text. The Navigate command and double-clicking the link in Browse Links now scroll to the destination and highlight the destination word.

Copied text links pasted into a new note failed to understand that their source note is the new note, not the note from which they were copied.

In the key attribute table, moved the Values popup rightward to allow space for the scroll bar.

Agent updates that refresh the screen no longer switch the focus away from the key attribute table if the key attribute is being edited.

Avoid double-escaping %-encoded URLs when opening.

In the key attributes view, URLs with non-ASCII characters now respond to the View In Browser button.

Note ▸ Explode is now available from the text pane as well as the view pane.

TextView again respects paragraph spacing.

After pasting an image, Tinderbox restores the typing attributes to what they were before the paste. (2412)

VIEWS

After turning off View ▸ Use Columns in outline view, columns are immediately removed. Previously, they were removed from the tab but not from the view. (2423)

Added a title to the Error button, since many people don't know what it is.

Fixed an intermittent crash when entering full-screen mode, caused by TbxMap's attempt to cancel the current editing session when adjusting the split pane.

7.5.0

HIGHLIGHTS

FORCE DIRECTED LAYOUT

In map view, **View ▸ Arrange ▸ Dance (⇧⌘-D)** initiates an automated layout of the view based on a physical simulation.

- Each link among notes in the map is treated as a spring that pulls linked notes together.
- All notes exert a gravitation attraction for other notes.
- Notes that overlap repel each other.
- At the beginning of the simulation, each note is subject to a random force, much as if it were heated. This force is reduced progressively over time. This process, known as simulated annealing, helps the simulation from getting tangled up in local minima.

Dancing automatically stops when a note is dragged or the selection is change. Dancing also stops when they layout ceases to change significantly.

Not all maps will benefit from automatic layout; the famously tangled map of Mary-Kim Arnold's "Lust", for example, does not. Performance may be

unsatisfactory in maps with more than a few dozen notes. Nonetheless, this may prove useful in many cases.

FLAGS

One common Tinderbox task is qualitative analysis of existing materials, such as letters, surveys, diaries, and personal papers. An important preliminary step in this work is *coding* —identifying occurrences of special interest for the study. For example, if we were analyzing a collection of nineteenth-century diaries to study what people recorded about food and drink, we might want to code where the food was consumed. We might mark every passage that discussed eating at home with the code P1, eating at the residence of another family member with the code P2, eating at a pub with P3, and so forth. We might also note places where money is discussed: C1 might indicate that the writer paid for their meal, C2 that someone else explicitly paid for the writer's meal, and so forth.

Flags offer a convenient and flexible way to foreground selected codes in map view. $Flags is a new set attribute; when not empty, small "flags" are displayed above the note in map view. (Flags do not appear in other views).

For simple coding tasks, using $Badge may be adequate. Flags provide a wider range of visual cues, and new flags can be improvised quickly when coding needs change.

Note that $Flags is a list attribute.

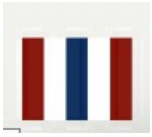
Flags are described using a concise textual shorthand.

The simplest flag is simply a color. The flag ' red' looks like this:

For horizontal stripes, list the colors separated by hyphens: red-white-blue

For vertical stripes, list the colors separated by the vertical bar character: blue|white|red

red|white|blue|white|red



Diagonal stripes use the '/' symbol: yellow/black

A checkered flag uses the '$' symbol: black$white

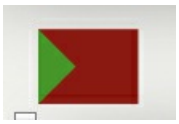
A variety of symbols may be overlaid on a flag. A cross, for example, uses the '+' sign: white+red

white+yellow/black

For a saltire, use an asterisk '*': white*light blue

For a diagonal line, use the percent sign: `white%light blue/red`

The '>' symbol adds a chevron: `green>red`



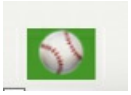
The '}' character adds a pall. `white}red`

The pall and chevron work together: Code>white}green>blue

Finally, the period "." adds a short textual annotation. `A1.red`

The text color is normally white, but may be specified: `black:C.lighter blue`

The text may be an emoji: `.green`



The color 'none' represents a transparent flag. `black:T1.none`

Web-style colors #RRGGBB and #RGB may be used in place of symbolic colors. `#FF0` displays a bright yellow flag.

CHART VIEW

The chart view has been rewritten and greatly improved.

Chart View now has its own options popover, accessible by clicking the Info button ⓘ on the chart view's tab. The popover allows you to change the chart style and adjust the width and spacing of chart items. Most significantly, you can now choose either a left-to-right or top-to-bottom arrangement of the chart.

Subtitles now appear in chart view.

NATURAL LANGUAGE PROCESSING

The text of notes is now scanned to extract information that might be useful for agents. These results include:

$NLNames: a set of personal names found in the text.

$NLOrganizations: a set of the names of organizations found in the text.

$NLPlaces: a set of place names found in the text.

These attributes ("NL" is short for "natural language") are part of the new "AI" attribute category.

Note that these values are extracted automatically and are subject to a variety of errors. Values are extracted asynchronously after a note has been edited; they won't automatically be extracted from existing notes.

REPETITION

Get Info ▸ Repetition offers insight into words that are used repeatedly in the selected notes, sections, or in the entire document. Consistent usage may be needful or desirable, of course, but noting repetition can call attention to opportunities to adopt more precise language. The pane lists words that occur between 2-10 times; words that appear more frequently are not listed. Tinderbox also omits all words with fewer than four characters, that appear in the built-in stoplist of 100 common English words, or that appear in the note named stoplist if one exists. The indexing process tries to treat words derived from a common stem as repetitions, so plurals and verb conjugations are often handled intelligently.

NOTABLE

MAPS

If a note begins with an image, that image will be drawn to cover the face of the note in map view. This applies to all shapes, where normal text thumbnails are only drawn for rectangular notes.

To see an overview of the entire map, press and hold the ctrl, option, and command keys simultaneously. The overview has been improved to ensure the entire map is visible. To zoom to a different part of the map from the overview, move the mouse cursor to the area of the map in which you are interested before you release the ^⌘⌥ keys.

When a note is selected, a new link widget appears for each outbound link. Dragging this widget allows you to vary the curvature of the link.

A new map guide notices when a note or adornment is approximate the same size as one of its neighbors, and snaps the note size to match the neighbor. If no nearby note is approximately the same size, the guide looks for notes with approximately the same height or width.

A new map guide looks for 45° diagonal alignments between nearby notes.

A new color scheme, Modern, introduces some fresh defaults and a muted color scheme.

TEXT

Text links and web links may now be copied and pasted within a document.

Tinderbox now supports text links that automatically scroll to a specific point in a note — including the note from which the link originates.

To specify a specific destination of the link, select the source and drag the text link to the link parking space. Then, select the destination and scroll so the destination text is visible. Finally, drag the link out of the parking space and click at the destination location. When the link is followed, Tinderbox will scroll the text view so the destination text is visible.

When following a text link with an explicit destination, Tinderbox highlights the word at the destination.

The size of a note's text window is set by the attributes $TextWindowWidth and $TextWindowHeight. Resizing the note's text window automatically sets these attributes, so that the text window for each note remembers its width and height.

The text size for the key attributes table is now determined by the attribute $KeyAttributeFontSize. This defaults to 11pt but can be set to whatever size is desired. Changing the default value (in the System Attributes Inspector) will change the value throughout the document. (2392)

The **Window ▸ KeyAttributes ▸** menu lets you choose some convenient sizes for the key attribute table. This menu controls the default value, used for all notes for which a specific size has not been set; use $KeyAttributeFontSize to set a different size for specific notes.

MISCELLANEOUS

ACTIONS

Fixed a crash when evaluating paths of the form **./path/to/note**, in which "." stands for for **this** note. Tinderbox customarily represented paths relative to this note as "path/to/note", using the omission of the initial "/" to indicate that the path was relative to **this** rather than to the root, but the " **./**" notation is familiar from Unix.

OnJoin actions are again performed when notes are added to a composite.

When using an action to move a note to a new $Container, Tinderbox avoids collisions with adornments as well as accidental composites with other notes.

Performing an action that changed $Name cancelled editing of names in map or outline view.

Code fields in the Inspector and Get Info: Agents now offer autocompletion of dot operators for expressions with designators: $MySet(/path).cou(nt) .

When autocompleting a dot operator, the autocompletions offered for the first character include only operators that begin for character. For example, $MyString.c will offer "contains" and "count" but not "icontains". If the operator being autocompleted age 2 or more characters, such as "$myString.co", all operators that contain that substring will be offered.

The operators **.paragraph(n)** and **.paragraphCount** now ignore empty paragraphs.

The new operator **weeks(before, after)** returns the number of weeks between two dates. If the dates are within seven days of each other, the result is zero.

The new operator **.paragraphCount** returns the number of paragraphs in a string. Thus, $Text.paragraphCount is the number of paragraphs in the text of this note. (Only available on macOS 10.13 and later)

The new operator **.paragraph(n)** returns the test of the nth paragraph in the string. The first paragraph is .paragraph(0). If the string does not contain n paragraphs, the result is the empty string. If n is negative, Tinderbox counts from the last paragraph; $Text.paragraph(-1) is the last paragraph in the text. (Only available on macOS 10.13 and later)

The new date operator **.week()** returns the number of the week in the current year. For example, January 1 is in week 1. **.week()** is read-only.

ATTRIBUTE BROWSER

When using the attribute browser, the key attribute disclosure triangle again updates the key attribute table's visibility appropriately

The Attribute Browser gains a contextual menu. **Open In New Tab** opens the note in a new Map view tab. Also available are **Text Window, Get Info, and Roadmap.**

Empty category labels now appear as "[no value]"

If an attribute browser tab has a query description, that description is used for the tab label.

The rubber-band link is now drawn correctly when creating links in Attribute Browser.

When defining date categories, two categories were incorrectly merged if they represented dates exactly one month apart — that is, if the category following January 11, 2006 happened to be February 11, 2006.

BROWSE LINKS

Browse Links now lets you type a link type to select it, or to create a new link type

CHART

The conventional Chart View is now somewhat more efficient in its layout.

Chart tabs now remember and restore the expansion state of their view.

DATES

The time component of ISO8601 dates with a comma separating the date and time (2018-05-01, 15:38) was not recognized. High Sierra began adding

the optional comma to its locale date repertoire, and this caused confusion.

ISO 8601 dates (dates like 2001-11-27 06:15) with times followed by “PM” are now interpreted as you might expect.

A note’s $ID value may be used as a designator. For example, $Name(1524673590) locates the $Name of the note whose ID is 1524673590. This usage should be avoided whenever possible, as it is hard to read and prone to confusion, but may be useful in some cases where unique paths are impractical.

The date parser now accepts “a” as an abbreviation for “am” and “p” as an abbreviation for “pm”.

Tinderbox actions that changed date attributes could, in some situations, inappropriately coerce dates to the current century if the user’s short date setting used two-digit dates.

DOCUMENT SETTINGS

In the General pane of **Document Settings,** changing the User Name updates the document setting immediately, without waiting for the user to press Return or Enter.

Closing the last Tinderbox window no longer requests unwanted confirmation if the Document Settings window is open.

Document Settings displayed an unexpected sheet when closed if the document was dirty and unsaved, because it implemented a convenience method **document** that shadowed a method of NSWindowController. Oh, my aching head.

In Document Settings, the list of color schemes is now alphabetized.

EXPLODE

Explode has been rewritten to cope with a long-standing source of confusion. The original expectation for the use of Explode was that delimiters would mark the end of each chunk:

> Winken•Blinken•Nod

Some users, not unreasonably, placed the delimiter instead at the start of each chunk:

1. Winken
2. Blinken
3. Nod

Here, the digit followed by a period makes a perfectly reasonable cue for exploding the note, but Tinderbox insisted on creating an empty note terminated by the first delimiter. Tinderbox now recognized both styles of delimiter placement and acts accordingly.

Explode now handles Unicode characters more reliably, and no runs in linear rather than quadratic time.

In the Explode popover, pressing **return** now activates the Explode button. This saves time when exploding several notes using the same parameters.

Exploded Notes is available in the built-in prototypes menu .

EXPORT

When exporting ordered lists, items are delimited by $HTMLOrderedListItemStart and $HTMLOrderedListItemEnd. Formerly, $HTMLListItemStart/End were used by both ordered and unordered lists.

Resolved a crash when previewing a note that uses ^action()^

FIND

In the Find results window, a new contextual menu choice lets you open the result in a new tab. Shift-double-click will also open the result in a new tab.

The **Edit** menu correctly reflects that **Find** is disabled in attribute browser.

GET INFO

When invoked from map view, the popover is positioned with respect to the map view, not to the selected note.

When searching for attributes in the Get Info|Attributes popover, Tinderbox looks for the search string at any position in the note. Thus, “URL” offers potential completions of $URL, $ReferenceURL, $SourceURL, and $NoteURL.

The word cloud (Get Info:Words) failed to acknowledge the document’s own stoplist if present. Now, the configuration file stoplist.txt is augmented by the words in the note named “stoplist”.

IMPORT

Tinderbox imports $Tags from items watched items in DEVONthink folders.

DEVONthink watch folders now import styled or structured text from a wider range of sources.

Finder watch folders now import from a far wider range of sources, too.

We no longer automatically import DEVONthink text items that exceed ~10,000 words.

Improved the handling of references to journal articles and conference proceedings imported from Bookends, Zotero, and other sources of RIS data.

Avoided a potential crash if two different documents updated watched folders at the same time.

Several different folders from the same source (e.g. in Notes) may now be watched.

Dragging Freemind (.mm) items from DEVONthink to Tinderbox now imports the Freemind maps. Previously, the DEVONthink item, had to be dropped onto the desktop first, in order to create a file.

Dragging image items from DEVONthink to Tinderbox creates a note containing the image.

Watched DEVONthink folders import the item URL in $URL; formerly they imported the NoteID, which is less useful.

Files with the extension **.ris** are now understood to contain RIS bibliographic data.

INFRASTRUCTURE

DisplayExpression is now evaluated on the agent queue, reducing thew risk of concurrency problems and resolving a reported deadlock.

Agent queries are now evaluated on the agent queue; previously, they were evaluated on the main queue, and this could cause trouble if (for example) the agent had a rule that was evaluated simultaneously.

Addressed a deadlock when editing key attributes that change the document’s sorting, when the edited note has a rule or edict.

Deferred sorting could be deferred indefinitely. Now, the agent dispatcher checks to ensure that deferred sorts are in fact carried out.

Timelines that used display expressions deadlocked during display, because both timeline layout and display expression evaluate were in contention for the agent queue.

setParentSafely no longer suspends the agent queue, because the agent queue is needed to evaluate DisplayExpression. Deadlocks should not longer occur when changing the parent of a view that uses display expressions.

Displaying the General category of Get Info:Attributes deadlocked if the selected note had a display expression, because building the values menu suspended the agent queue. Instead, we now build the values menu on the agent queue, *and* we also avoid building the values menu for read-only attributes like $DisplayName, since we cannot use the menu to change the value.

Corrected a bad constructor for LayoutInfo(Node*), which caused intermittent failures in top adornment guide tests.

Tinderbox would sometimes engage in numerous redundant rule and edict updates when the selected note had rules or edicts, if the note also had a date as a key attribute.

Dragging a note over a very large note or container caused sluggish drag tracking, because Tinderbox continually and unnecessarily redrew the drag highlight.

Added further checking to prevent rules from running while Tinderbox is trying to close the document.

Update Agents Now could crash because it was run on the agent queue, which is now needed in order to evaluate agent queries. We still want to get this task off the main queue, so now we use the global queue as an ad hoc expedient.

When dragging notes in map view, we no longer rebuild the entire quad tree when notes move. Instead, we update only the notes that move. This can leave the tree slightly out of balance, so we rebuild the tree entirely after the drag is complete.

Fixed several minor memory leaks involving read-only and Preference attributes.

Addressed a potential crash when using $Prototype simultaneously in running rules or agents and in a display expression or as an outline column.

INSPECTOR

When searching for attributes in the system attribute inspector and the quickstamp inspector, Tinderbox looks for the search string at any position in the note. Thus, “URL” offers potential completions of $URL, $ReferenceURL, $SourceURL, and $NoteURL

The “case insensitive” checkbox in the Action Inspector’s Query pane has been removed. This caused much confusion because it only applied to an obsolete syntax; .contains and .icontains are clearer.

The Sort pane of the Action Inspector now offers search fields for locating the sort attribute.

The Texture menu is now alphabetized.

The User Attribute pane of the Document Inspector now rejects attempts to create an attribute name that contains operators like + or punctuation like “,”.

Addressed problems when selected “normal” from the appearance inspector’s Shape popup.

LINK EDITOR

Changing the link type in the links inspector popover no longer deselects the current link.

When creating a link, after the link popover is expanded to show the Title, Class, and Target fields, those fields are properly enabled.

MAP

Greatly improved the performance of maps while dragging notes. Also improved mouse-wheel scroll, two-finger scroll, and pinch-zoom performance.

When dragging notes in map view, only links to and from the dragged notes are drawn. This focuses attention on the most relevant links and significantly improves performance.

Using map resize handles for the left and top edge to reduce a note’s width or height to less than a plausible minimum to longer moves the noteDragging a locked adornment no longer displaces the map position.

Improved text layout in the hex shape.

Locked adornments are again selectable.

Summary table expressions could crash when truncating cells that don’t fit, if the cells contained Unicode characters.

If the text of a container or agent begins with an image, the image is drawn in its title bar.

In map view, if a container contained an alias of itself, Tinderbox did not display it. If we displayed the contents of containers within containers, this could create a hall of mirrors, in which the container’s alias would be displayed inside itself indefinitely. But since we don’t display the contents of containers within containers, this is no longer a concern.

Changing the font size of a selected alias no longer italicizes the font of the original. Previously, Tinderbox saw that the alias was italic and assumed that you wanted to italicize the original, too.

The edge alignment guides now examine notes within a radius based on the size of the dragged note, rather than based on a fixed radius.

We now draw stub count indicators when only one link is inbound or outbound, as it is useful to right-click the stub to see what the link connects to.

In map view while editing the note name, pressing [Tab] will move to the subtitle even if there was no subtitle previously.

Renaming notes in map view using click—hold could mislead Tinderbox into thinking that a drag was still in progress, suppressing link drawing.

The composite name widget now sets the cursor to the arrow cursor on hover, making the control easier to use.

Adornment grid labels reliably accommodate their descenders.

If an adornment inherits $Lock or $Sticky, the document was saved properly but the value was not correctly read when the document is reloaded.

Map borders are no longer suppressed when less than 1px wide.

Map View is smarter about displaying guides. For example, no amount of dragging an object can change its aspect ratio, so aspect ratio guides are now appear only when resizing an object.

The size of the map body text scales more appropriately with changing map magnification.

Applying a stamp to a note that is being renamed now ends the current editing session, rather than canceling it. Since you’ve made changes and then applied a stamp, it makes sense to keep those changes.

Format ▸ Text ▸ Align Left/Center/Right commands are now applicable to the titles of selected notes in map view.

Edit ▸ Copy View As Image omitted links that were outside the currently-visible window viewport. All links are again included.

For mouse-lovers, option-scroll-wheel is now synonymous with pinch zoom. When using the trackpad, option-two-finger-swipe is also synonymous with pinch zoom.

OUTLINES

After editing a column value in outline view, the key attribute table is updated in case the edited value is a key attribute.

In outline view, Tinderbox did not immediately remeasure the height of an item after double-clicking it to rename.

Adding or deleting columns in outline view immediately updates the horizontal scroll state of the view.

The parent node of a hoisted outline was not drawn after switching tabs.

When editing column values in outline view, values are updated as you type. Previously, updates only took place after pressing Return.

Improved pinch zoom in outlines.

PROTOTYPES

When setting the prototype of a note, Tinderbox could choose an unexpected prototype if several notes share the same name. Tinderbox will now use as the prototype a note with $IsPrototype set to true, if one exists. Previously, Tinderbox used the first note in outline order.

The OnAdd actions for the built-in containers for Prototypes and Templates now set $NeverComposite=true. This should reduce the incidence of unwanted accidental prototypes.

STAMPS

Stamps: the designator that is bound to this in stamps, as it is in actions and queries, facilitating its use in complex stamps.

TEXT

When the text pane or the view pane acquire the keyboard focus, either by being clicked or by ⌥-Tab, a focus ring briefly appears at its boundary.

Text windows no longer lose their selection when deactivated, scrolling to the top when activated.

Edit ▸ Duplicate correctly updates the text pane; previously, the text pane continued to view the original note.

When switching tabs, the text pane could be hidden inappropriately. This happened when switching away from a tab with a collapsed text pane and no note selected, when the destination tab had an uncollapsed text pane.

The key attribute table contextual menu is now available in text windows.

Text windows now offer a tooltip containing the path of the window.

When the Key Attribute Picker appears, the insertion point in the list of key attributes is now set at the end of the list. Standard behavior calls for the current list to be selected, but this suggests that the user would typically want to replace the entire list with a newly-typed list. That's probably unlikely. This is surprisingly difficult, requiring subclassing the token field and hijacking an NSResponder method. Yikes!

The token field of the key attribute picker offers autocompletion options for all attributes that contain the current string, not only those beginning with that string.

The key attribute picker would accept an invalid attribute name if no valid but unused new attribute name was also added in the same edit. It won't do that anymore.

Format ▸ Standard Font and Format ▸ Standard Size no longer require a text selection, as they can be useful for resetting the typing attributes even when no text is selected.

The key attributes table's contextual menu now allows you to Copy the value of an item, which is especially handy for read-only values.

Activating Tinderbox no longer scrolls the text pane to make the selection visible, unless the text has been changed while the Tinderbox window was not activated.

Double-clicking a key attribute's value in the text window failed to activate the editor, because the text window did not set up handling for the double-click.

The Quicklinks facility was rewritten for more reliable operation.

If a new user attribute is created directly in the key attribute table, attributes will now be sorted appropriately.

The text pane is updated after applying a stamp if the stamp changes $Text.

Renaming a note by editing the title in the text pane is now undoable.

TextColor and TextBackgroundColor are correctly observed when formatting a note's text.

TREEMAP

The treemap Info popover has an additional control, **maximum depth,** that allows you to display only the upper levels of the treemap. This can sometimes improve clarity of presentation.

Selecting a note in Treemap view now updates the text pane.

The layout of the treemap settings popover has been corrected, allowing sufficient space for the slider.

VIEWS

A new contextual menu in is available in views: **Open Prototype In New Tab**.

Badges and the badge widget are highlighted at mouse down, and editing the badge may be cancelled by dragging the mouse out of the badge area before mouse up.

7.3.0

HIGHLIGHTS

Watching Outside Resources

Tinderbox now provides a richer and more flexible mechanism to automatically import notes created elsewhere, including notes from other applications, other computers, or from your iPad and iPhone devices.

Three new items appear in the File menu.

File ▸ Watch ▸ Folder From Notes… will create a top-level container that holds notes imported from a designated folder in the Notes application.

File ▸ Watch ▸ Notebook From Evernote… will create a top-level container that holds notes imported from a designated Evernote notebook.

File ▸ Watch ▸ Folder from Finder… will create a top-level container that holds notes imported from a designated folder. These containers will watch the corresponding application periodically and will automatically add new notes and update notes which have changed. Changes made in Tinderbox are *not* forwarded to the application.

Tinderbox can now automatically import selected notes from the **Notes** application, which is installed in all macOS and iOS devices.

To connect Tinderbox to a Notes folder, create a top level container to hold the imported notes and set its $NotesFolder attribute to the name of the Notes folder you want to import. Tinderbox will now automatically import or update that container whenever your file is reopened, and periodically thereafter.

The imported notes inherit from a built-in prototype named “Imported From Notes,” allowing you to set common key attributes or visual appearance.

Changes to the imported notes will not be propagated to the application or to other devices.

The import process is comparatively time-consuming; it may be preferable to limit import to folders with no more than a few dozen notes.

Tinderbox can also automatically import selected notes from **Evernote**. Evernote import works much like import from Notes.

To connect Tinderbox to an Evernote notebook, create a top level container to hold the imported notes and set its **$EvernoteNotebook** attribute to the name of the notebook you want to import. Tinderbox will now automatically import or update that container whenever your file is reopened, and periodically thereafter.

The imported notes inherit from a built-in prototype named “Imported From Evernote,” allowing you to set common key attributes or visual appearance. The $URL of each imported note provides access to the corresponding data in Evernote.

Changes to the imported notes will not be propagated to the application or to other devices.

The import process is comparatively time-consuming; it may be preferable to limit import to notebooks with no more than a few dozen notes.

Tinderbox can now automatically import selected notes from the any Finder folder application, including folders in Dropbox or other remote servers.

To connect Tinderbox to a folder, create a top level container to hold the imported notes and set its $WatchFolder attribute to the name of the Notes folder you want to import. Tinderbox will now automatically import or update that container whenever your file is reopened, and periodically thereafter.

The imported notes inherit from a built-in prototype named “Imported From Finder,” allowing you to set common key attributes or visual appearance.

Changes to the imported notes will not be propagated to the application or to other devices.

Tinderbox can now automatically import selected groups of records from **DEVONthink Pro**.

To connect Tinderbox to a DEVONthink group, create a top level container to hold the imported notes and set its **$DEVONthinkGroup** attribute to the unique DEVONthink ID of the group you want to import. Tinderbox will now automatically import or update that container whenever your file is reopened, and periodically thereafter.

Groups inside groups are note currently imported.

The imported notes inherit from a built-in prototype named “Imported From DEVONthink,” allowing you to set common key attributes or visual appearance.

Changes to the imported notes will not be propagated to the application or to other devices.

NOTABLE

DEVONthink Pro

Multi-item drags from DEVONthink receive appropriate Name and $URL for each item.

Tinderbox does a better job of respecting pdf imports from DEVONthink Pro.

Tinderbox Autofetch failed to recognize DEVONthink tags.

Tinderbox AutoFetch now updates styled text from a wider range of DEVONthink record types.

Import

AutoFetch now works with a greatly expanded range of file types, including text, rtf, Microsoft Word®(.doc), pdf, and markdown (.md) files.

Taskpaper import now recognizes @due(*date*) and @start(*date*) tags, as well as the @done(*date*) tag.

For Evernote import, the new attribute $SourceURL is set to the source URL of the note, it the note has one.

Maps

Transparent adornments may now draw grid lines as well as borders.

Smart adornments now allow space for multi-line subtitles, assuming that the adornment is drawn at or near standard magnification.

Link stubs are drawn with the correct colors. The arrows of link stubs are drawn a little more nicely.

Option-drag to break composites no longer duplicates the dragged note.

Outlines

If outline columns are wider than the outline pane, the pane scrolls horizontally.

Promoting and demoting large numbers of notes is now much faster.

Text

Smart links in text are once again clickable.

When the text background is dark, the insertion point is now white rather than black.

MISCELLANEOUS

Reference import: the abstract is now stored in $Abstract, rather that $Text

Tinderbox again displays emoji badges.

Corrected a memory leak in disposing of actions, and ensured that the action disposal pool is thread safe.

Actions and Dashboards documentation: updated the URLs for the Tinderbox forum and aTbRef.

When Tinderbox reads a custom config.xml from the application support folder, it first initializes the configuration to the built-in config.xml. Formerly, configurations left unspecified in the custom file used undocumented defaults.

When duplicating a note with a name that ends in a number not preceded by a space, Tinderbox increments the number without inserting a space or changing the prefix characters. Thus, the duplicate of “test3” is “test4”.

The attribute browser now prefers not to interrupt an editing session to update the display.

Tinderbox no longer attempts to escape characters in the $URL field before passing them to the browser.

Views are more thoroughly updated after applying a stamp, because stamps might change the view layout in unanticipated ways.

Corrected a long-standing implementation error in &=

Moving the splitter crashed if the view pane held an outline view with an open editing session.

The notify() operator crashed when invoked from a stamp, because it deleted its delegate before the notification system used it.

Occasional Tinderbox crashes when closing complex documents should be more occasional.

In outline view, applying a stamp while editing the name of a newly-created note could crash

Plain text and markdown items imported from DEVONthink now respect the note's default $TextFont and $TextFontSize.

Increased the limit of PrototypeBequeathes children from 100 children to 500 children.

When items from Finder or DEVONthink are dragged into a Tinderbox outline, the appropriate highlights are displayed to indicate where the item will be created. Problems in choosing the location of the created item have been addressed.

Discovered a concurrency error in the implementation of $Container that could deadlock the agent thread.

Edits in column view failed to commit changes.

Calculations involving dates prior to 1970 could fail, because Tinderbox was using an obsolete intermediate representation for dates in some actions.

Avoid a rare source of crashes after closing documents on very busy machines by more promptly shutting down the TextPaneListener's work queue.

If a document has several open windows, Tinderbox asks for confirmation that you really want to close the window and its arrangement of tabs.

In complex maps, link animation can consume significant computational resources. Previous versions suppressed link animation when it became difficult; instead, we now slow down the link animation. Link drawing has been slightly hastened.

Subtitle sizes now again scale correctly with map magnification.

Tinderbox updates the column status immediately upon being instructed to use columns; previously, the update to layout could be delayed until after the view had been redrawn .

After editing a key attribute of a note, the note's edict and rule are performed on the agent update queue rather than the main queue, avoiding a rare crash.

Adornment OnAdd actions of the form $AttributeRef=; no longer generate spurious error messages.

Document Settings:Accounts no longer refuses to delete the last Simplenote tag or to clear account credentials.

The key attribute picker now ignores proposed key attribute names that cannot refer to attributes, such as "3cats" and "My sprocket."

Sticky adornments no longer adjust note positions when resized; notes are moved as before when a sticky adornment is moved.

Modernized logic for exporter sheets to avoid deprecated APIs.

Note ▸ Explode settings are now saved within a session, making it easier to experiment and allowing more efficient workflows.

The sheet displayed when closing one window in a document with multiple main windows now offers an option to close the entire document instead.

$DEVONthinkGroup may now contain either the DEVONthink UUID for a group or a DEVONthink URL (x-devonthink-item://…) for that group.

7.2.0

HIGHLIGHTS

DEVONthink Office Pro: When an item is imported to Tinderbox from DEVONthink Office Pro 2.9.15 or later, Tinderbox imports the DEVONthink Creation time as $SourceCreated and the DEVONthink Modification time as $SourceModified. OnAdd actions are applied to each item of a multiple-item drag. Multiple-item drags are now handled without adding unwanted empty notes.

Infrastructure

Extensive revisions help Tinderbox get more work done by using your Macintosh more efficiently and avoiding conflicts between separate tasks.

Edicts will no longer run at the same time as agents.

The internal design of computed attributes — attributes like $WordCount and $SiblingOrder that are computed from the properties of each note — has been changed substantially in an effort to address intermittent conflicts between Tinderbox threads. This may pose a variety of performance issues.

Several data race conditions have been resolved, most notably in handling trees, updating progress bars, in tracking shared value objects, and in disposing of unused values.

Markdown

A new built-in prototype **Markdown** allows notes to use Markdown in place of native Tinderbox markup.

The new **$HTMLPreviewCommand** now permits you to use Markdown syntax in your Tinderbox notes, and to either preview the formatted output or to export to HTML whenever you like.

To customize the appearance of Markdown previews, you may edit the CSS stylesheet **styles.css** found in the Markdown folder of the Tinderbox support folder.

Import

When importing from DevonThink, items that have text but not styled text now import the text.

Email message (.eml) files are now imported.

RIS reference import (and option-drags from Bookends) now extract the reference's abstract, if present, and place it in the attribute $Abstract.

RIS references from BibDesk are now handled appropriately. Bibdesk drags include two distinct text items, only the first of which contains a payload. As a result, dragging RIS from Bibdesk generated two Tinderbox notes, the expected reference and an empty note named "New." The unwanted note is no longer created.

OmniFocus import has been greatly improved. We import the correct number of notes, obtain their omnifocus:/ urls, and set their prototype to Task. OmniFocus import now imports the task's key dates: Defer Until becomes $StartDate, Due becomes $DueDate, and Completed becomes $EndDate. If a task was completed before the present time, the note is marked as $Checked. If the task has a note, the text of the note is placed in the Tinderbox note's text. Unspecified dates are now treated as never rather than now.

Text

When displaying the text of a composite or a multiple selection, each note is labelled. Formerly, the first note was not labelled.

The subscript and superscript commands in Format ▸ Font ▸ Baseline now reduce the font size of the selected text by 25%.

Format ▸ Style ▸ Standard Font now tries to respect passages that are bold or italic. It changes the font family to the note's default font family, using the note's text size.

When the text pane becomes too small to be practical, it once again fades. Changing the selection no longer reopens a closed text pane.

When a text selection is dragged from Safari into the Tinderbox text pane, Tinderbox records the source URL in the note's $URL attribute. If the note already has a $URL value, however, it will not be changed.

NOTABLE

Tinderbox is again compatible with macOS 10.9.

Browse Links

The URL field now displays the entire URL as a tooltip, and also allows additional space.

When a text link is selected, the text pane scrolls to make the link anchor visible and the link anchor is temporarily highlighted.

Experience

Large documents with numerous aliases will load a bit more quickly because the Hypertext class no longer maintains duplicate lists of notes.

Changes in protocol for deleting nodes and associated changes in marking documents as being in the process of closing improve stability when closing complex documents, especially documents with numerous agents.

The contextual menu for the key attributes table gains an additional choice, **Evaluate.** The current value of the key attribute will be evaluated and replaced by the result.

The suggested values of **$Shape** are pre-populated with the standard shape names. In the key attributes table, $Shape will always have a suitable pulldown menu and will offer suitable autocompletions.

If a built-in composite is added to a document in which the default value of $NeverComposite is `true`, the components of the built-in composite have $NeverComposite set to `false`. Presumably, if you're adding a composite, you want it to be a composite.

If we are *not* searching text, Tinderbox no longer attempts to provide context for the Find result in the text we are not searching.

In the key attributes table, font attributes now have a button for choosing the font, making it easier to discover that fonts can be changed here.

When a new Composites container is added to the document, its initial OnAdd action is now `$NeverComposite=false;`

File ▸ Export ▸ As HTML, **As Text**, and **As Outline** are no longer unavailable from Preview and HTML panes.

Document Settings:Map now allows you to set the default texture. The new Texture popup menu button is initialized to "none" when no $Fill texture is selected.

Full Screen Split View: Tinderbox and Storyspace now support full screen split view. Press and hold the green "full screen" button in the main window's upper left-hand corner; the window will shrink to permit you to select the right or the left half of the screen.

Composites: to avoid confusion, **Notes ▸ Create Composite** has been renamed **Notes ▸ Add Composite**. If no composites are available to be instantiated, this menu item is now disabled.

Get Info: The search box in the Attributes panel no longer automatically selects a completion if only a single completion is possible. Though only one completion is possible, the user might have changed her mind, or might have mistyped the intended attribute.

When Tinderbox recognizes a registration code, it celebrates more emphatically to make its gratitude clearer. In the past, some users failed to notice the change.

Key Attributes Table: improved alignment between attribute label and values

HTML Export

^**^** and **^outboundTextLinks^** now always operate on the original note, even when **this** is an alias. Aliases share their text with the original notes.

The export element ^docTitle^ again returns the current document's name.

An agent gathers a list of notes, and wants to use ^childLinks^ to create a list of those notes. This seems useful, but if the agent doesn't export a copy of its children, ^childLinks^ would export links to pages that don't exist. A new solution: if ^childlinks^ appears in a note which does not export its own children, it instead links to the original of each child.

$HTMLLinkExtension is now respected by ^childLinks^, ^path^, and ^linkTo^.

HTML Preview and Export: A new attribute, **$HTMLPreviewCommand,** allows you to pass the text of a note to an external script in place of using Tinderbox's text markup. If not empty, ^text^ passes the unprocessed text of the note to the script in $HTMLPreviewCommand instead of processing the text itself. This allows you to choose your preferred flavor of Markup or any other markup processor. Though primarily intended for use with the preview pane, this mechanism is also used for HTML Export.

Markdown: If your copy of markdown is installed, say, in /usr/local/bin/markdown.pl, any note that has a $HTMLPreviewCommand of /usr/local/bin/markdown.pl will be processed with Markdown.

Map

The document setting **If Note Name Is Too Long: shrink font** now calculates the font size properly.

Tinderbox's approach to scaling image adornments has changed. The image is now scaled to that it always fills the adornment frame, and is then cropped as necessary if the adornment's aspect ratio differs from the aspect ration of the original image.

When dropping multiple images into a map, the image adorments are placed more appropriately.

If an alias is selected, View ▸ Tab ▸ New Tab will open a new tab on the original note of the alias. Previously, this option was disabled when the selected note was an alias. Since aliases cannot have children, it is pointless to open a map of their interior, but it may sometimes be handy to open a map of the interior of a container for which we have an alias.

The map view prototype tab no longer clips descenders of the prototype name at standard magnification.

The command Edit ▸ Create Alias frequently created unwanted composites of aliases, especially when used when several notes were selected. The aliases are now placed in empty parts of the map, and the view automatically scrolls to place the first alias in view.

The contextual menu for items in map view now contains a Shape submenu.

Outlines

Moving the pane splitter promptly updates the internal geometry of separator titles.

If a note is outdented in a hoisted outline, so that it no longer falls inside the section of the outline being viewed, Tinderbox now removes the note's view from the outline.

Quickstamp

The popup values menu in the Quickstamp inspector now functions as expected when the text field does not have the keyboard focus. Formerly, it was necessary to select the value field in order for the popup menu to take effect.

The popup values menu will now show up to 199 distinct values.

MISCELLANEOUS

Actions

OnRemove: Immediately before a note is deleted, the OnRemove action of its parent container is performed. If the note lay on an adornment, the adornment's OnRemove action is performed. In the OnRemove action, **this** is bound to the note that is about to be deleted.

$OnAdd formerly did not affect adornments. Actions are now sufficiently expressive that we can allow this; use a conditional action if you want to avoid changing adornments in OnAdd.

Queries: Queries and other expressions do not require and should not have a closing semicolon. The semicolon separates multiple *actions;* on might changes several facets on several different notes in an action, but a queries must be either true or false and cannot be a list of things. Still, it is easy to add an unwanted semicolon at the end of a query

```
$MyNumber==5;
```

Tinderbox now ignores the semicolon. Formerly, this was parsed as

```
$MyNumber=="5;"
```

which is not very useful.

The system attribute **$Color2** has been renamed **$AccentColor**. The old name will continue as a synonym, but the $AccentColor is preferred.

$MapBackgroundColor2 becomes **$MapBackgroundAccentColor**. The former name remains, but is deprecated.

The **format()** operator failed to assert that its result was a string. As a result, some string operations applied to the results of the format() operator returned incorrect results.

Tinderbox no longer flags some valid actions as errors — notably, removing an attribute's value is no longer flagged.

The result of evaluating an attribute that does not exist is now *false.* Previously, evaluating an attribute that does not exist returned the attribute name.

.replace now work correctly with Unicode. Previously, replacing characters with characters of different width could create invalid strings.

Agents

If an agent was created inside a container, and if the container's OnAdd expression sets $DisplayExpression (or select additional attributes), the OnAdd expression failed to take effect. Other actions (such as setting $Color) were effective, and the action behaved normally when moving an agent into the container from another location.

When editing agent queries, autocomplete is now smart enough to avoid suggesting == as an autocompletion of !=.

Attribute Browser

When browsing numeric attributes in the attribute browser, notes that were within 0.5% of the maximum of the previous bin were placed in the wrong bin.

The configuration popups of the attribute browser mishandled **⌘-delete**, passing it to the attribute browser and deleting the selected not rather than deleting the text to the start of the text field.

Badges: Missing badge names were mistaken for emoticons and drawn as text.

Help

Help ▸ Tinderbox Forum links to the new Tinderbox forum

Help: with regard to export templates, clarified the distinction between **this** and **current**.

HTML Export

^value()^ failed to initially bind **that** to the note referred to by **this**, and so some expressions of the form ^value(find(…(that)))^ failed to find any notes.

Import

Addressed a crash when dragging a DEVONthink .webarchive item into Tinderbox.

Inspector

The Tinderbox Inspector's Agents and Rules pane will update its edict update time immediately after edicts are run, without requiring a manual refresh.

Improved layout of Links inspector, moving the action menu to the upper right-hand corner.

Selecting a new attribute in the system attribute inspector immediately updates the default value.

In the System Attributes and User Attributes pane, the behavior of the "Key Attribute For Selected Notes" checkbox has changed. If some selected notes have this key attribute and others do not, the checkbox will initially be unchecked. Clicking the checkbox will assign this key attribute for all selected notes. Clicking it again will remove this key attribute for all selected notes.

The System Attribute Inspector failed to update the visibility of the Suggested Values field after selecting an attribute by using the search field.

The System Attribute Inspector reset its selection incorrectly after updating the selected attribute's default value.

Revised layout of the Document Inspctor to eliminate unsightly overhang of "Suggested" values field in System and User attribute inspector panes.

When a new attribute is created, or when the type of an existing attribute is changed, the default value of that attribute is now reset to a conventional value. For example, the default value for a new numerical attribute is 0, and the default value for a new Boolean attribute is false.

When a user attribute is renamed in the Document Inspector, the values of the old attribute are moved to the new attribute, and key attributes referring to the new attribute are updated to use the new attribute.

Maps

Image adornments bequeathed to new instances of a prototype failed to clone their images and were created as plain adornments.

Paste

Tinderbox could confuse itself when pasting an alias into a new container, because the bookkeeping semantics of CeresProxyDeleter were incorrectly returning the hidden proxy rather than the newly-created alias copy.

When a note is copied and pasted, the newly-created note has $Created and $Modified set to the time it was created. Formerly, the pasted note retained the $Created and $Modified values of the original note.

7.1.0

NOTABLE

IMPORT

DEVONthink import now fetches styled text, and requires DEVONthink Pro Office 2.9.11 or later.

MAPS

In summary tables, characters with diacritical marks or in languages other than English could generate encoding errors when long strings were truncated to fit columns.

Improved scaling of xy plots so that the point at the right edge of the graph can be drawn fully. Previously, half of the dot was outside the clip area.

PERFORMANCE

A variety of small changes in AgentManager, RuleManager, and AgentState avoid some wasted effort and computation. For example, we can avoid worrying about checking for collisions if we have a cleanup method, since the cleanup will break any accidental composites. We can be more efficient, too, in checking whether we have adornment actions that might need to be applied, and in requesting re-indexing of notes when we haven't actually changed anything.

TEXT

When activated, Tinderbox reset the text cursor to the start of the selected note if the previous selection was at the end of the note. It now leaves the cursor at the end of the note.

The text pane can again be fully collapsed. At narrow widths, the text shader message is no longer drawn as an unsightly jumble; instead, it begins to fade at 100px and fades fully at 90px.

Quicklinks now accepts multi-character cues.

Quicklinks now always links to the original note. Previously, Quicklinks could link either to the original or to an alias, which was confusing.

MISCELLANEOUS

In the Action Inspector, the sample queries concluded with semi-colons. That's incorrect.

Avoid a reported crash when validating the browseLinks: menu item in treemap view.

URL buttons on attribute browser columns failed to open the appropriate URL.

Edicts now post a notification for the use of the Tinderbox Inspector's Agents & Rules pane, just as rules do.

Changing the parent of a map view moved all adornments descended from the new parent into that map view. An improvement to the code, in this case, failed to improve the code.

The agent and edict queues now have more informative names; this may help with debugging an apparent problem where two different agent queues apparently step on each others' toes.

A variety of changes seek to identify and avoid rare problems when evaluating rules.

Avoid a crash when Tinderbox begins an animation to present a Hover panel, but the window is closed before the hover panel appears. Closing a window now immediately cancels the pending Hover.

Avoid opening a new tab when processing a `tinderbox://`URL that selects a note descended from, but not an immediate child of, the parent of an existing outline, treemap, or chart tab.

When the text pane is very narrow, Tinderbox no longer hides the text pane, nor does it try to format the text in the available space. Instead, Tinderbox formats the text with a reasonable line length and shows the left edge.

In the usual form for exportedString, the second argument specifies a template note

```
$MyString=exportedString(this,"/myTemplate")
```

The second argument may also supply the template inline:

```
$MyString=exportedString(this,"^value($MyNumber)^")
```

```
$MyString=exportedString(this,$MyTemplateString)
```

The alternate form returned the empty string; it now functions as expected.

In a sample file, a link from a prototype to another note used an eccentric routing because Tinderbox expected the many outbound prototype links to be visible, and all these links happened to have higher link priority than the one visible link. Routine now ignores web links and prototype links.

In the text pane when multiple notes are selected, the separator between each notes is now annotated with the following note's DisplayName.

The orange window bug. Tinderbox would find itself in an inconsistent state – readily visible in map view because everything would be drawn in the orange "error color" – if multiple windows were open in a document and one window were closed without closing the document. If no other documents were open, the single document assumed the application was about to quit when it saw the main window closing, forgetting that other windows might remain.

Clicking on a composite's rename widget failed if the widget happened to be on an adornment; instead, the adornment received the click. Now, the small widget receives the click instead of the large adornment.

The HTML Export progress bar now updates correctly on the main thread, and has the appropriate height.

7.0.2

NOTABLE

Composites

Moving a note in outline view could lead Tinderbox to change its map position if the note lay atop an adornment.

Notes lying atop sticky adornments could shift position slightly after the adornment was dragged if the scale was zoomed out. The Adornment Spacing kibbitzers incorrectly tried to snap an adornment to be positioned inside its own boundary.

Move To Front and **Send To Back** no longer reposition adornments that overlap other items.

Import

Corrected handling of multiple-item drags from Bookends 12.8 and DEVONthink Pro.

Maps

Notes dropped into other notes are now placed more visibly in the new container's viewport.

Speed

Some users – often with older Macs and often using macOS 10.11 – experienced poor performance in map view when some views were not drawn on integer boundaries.

The indent and outdent animations in outline view are snappier

Identified a possible bottleneck that may make changing the selection slow for some users, as updatePolicies was incorrectly forcing an update to the width of the text pane even though the text pane width had not changed.

The tab bar was redrawn during drags, which may have caused some performance degradation.

Windows

Hold the command key ⌘ while resizing the main window to change the window size without changing the width of the text window.

After creating a new web link, the text pane no longer scrolls to the top.

The HTML pane now opens at the start, rather than the end, of the HTML.

Activating Tinderbox no longer scrolls the selected text to the top.

MISCELLANEOUS

In outline view, copying a container and then pasting failed to assign prototypes to the pasted descendants of the container, while in map view the prototypes were assigned correctly. All views now handle prototypes correctly.

The **Note ▸ Create Composites s** ubmenu was not being updated corrected.

At very low magnification in Map view, the link widget could be difficult or impossible to click because Tinderbox dispatched its clicks to the outbound link stub handler, which should lie beneath the link widget.

Corrected a typo in "What's New?"

The Appearance and Tinderbox inspector panes again run the full width of the window.

Closing an inspector no longer prevents undoing prior inspector actions.

The name of named composites is drawn slightly higher in the map view to allow space for descenders.

Edit ▸ Copy View As Image now draws links that lie outside the current viewport.

Corrected the layout of the plot pane of the Appearance Inspector .

Changing the view type could deselect some selected notes.

Extensive refactoring of view pane management, especially concerning the pane splitter.

Adjusted the layout of the Interior pane of the Appearance inspector so the "large badge" checkbox is correctly placed.

Guarded against a possible deadlock when building the value list for Quickstamp, when Tinderbox is sorting on an attribute we are inspecting with Quickstamp

$HTMLLinkExtension now works.

$HTMLLinkExtension: some users use a post-processor like Jekyll to process their Tinderbox files after export, and want to link to the processed and modified files that will be created after export rather than to the exported files. When constructing text links, $HTMLLinkExtension specifies the extension to be used for link, and replaces $HTMLExportExtension. For example, exported files might have $HTMLExportExtension of .md; these files would be processed to create .html files and contain links to the .html files that will be uploaded to the Web server.

$ViewInBrowser failed when invoked by selecting a note. Sigh. Too many levels of indirection.

Problems moving and resizing the window have been traced to an apparent bug in macOS, which fails to call the NSSplitView delegate

```
- (BOOL)splitView:(NSSplitView *)splitView shouldAdjustSizeOfSubview:(NSView *)subview
```

and instead declines to resize the window. This method was used to constrain the width of the text pane when resizing the window while pressing the Command (⌘) key; that task is now performed by the TbxMapCoordinator, which is invoked by the window delegate.

The "General" tab in Document Settings was inadvertently disabled.

The Preferences window gains a note guiding new users to Edit ▸ Document Settings .

Inexplicably, a change in Tinderbox 7 that removed the requirement that SRect::AsNSRect return integral rects causes performance trouble for some users in map view.

Tinderbox could crash on close because the new splitView delegate was not released properly on close.

Fixed a possible crash when drawing a plot that has no specified target line.

Extensive text drawing modernization.

When browsing files in the key attributes table, you can now select folders as well as files. (1999)

When a note's **$ReadOnly** is true, its text pane is not editable.

In the Inspector, the export folder button is correctly labeled even if no note is selected.

An action may once again be simply a stamp name.

Quicklinks now accept multiple characters, which is convenient for whittling down large menu lists in big documents. For example if **[[c** display too many characters, type **[[can** for a menu of destinations starting with “can”.

Sierra tabs are enabled again.

7.0.0

*With **175** visible changes (and many, many more improvements for speed, accuracy, and maintenance), Tinderbox 7 is a big release. If you are in the habit of reading release notes, some news you'll want may be buried in the Details section in order to keep the Highlights concise.*

HIGHLIGHTS

Composites

Composites are clusters of notes that move together in map view, and that can know about the other members of a composite.

To form a composite, simply move two notes together so they touch. (To break up a composite, option-drag any member of the composite or select the composite and choose **Edit ▸ Break Composite**.

When selected, composites are outlined with a darker and thicker bounding box. The composite name and its edit widget are also displayed when the composite is selected.

When a note joins a composite, its outline position is changed to fall after the last item in that composite.

Several **Built-In Composites** are available in the File menu and provide useful examples. The built-in composite for **lists** has a single dark header, followed by one or more items. When a new item is dragged to the bottom of the list, its color, size, and position are set automatically by the list item's $OnJoin action. This technique can be very useful. The built-in composite for **lectures** is useful for conferences and syllabus planners. It's typical of a number of situations where we want to capture various facets about an event. Again, techniques used here may be useful in many other contexts.

Edit ▸ Break Composite can be used to break up a composite into individual components.

A new boolean attribute, **$NeverComposite**, allows you to exclude a note from all composites. By setting the default value of $NeverComposite to true, composites can be turned off throughout a document.

Fonts

Four new fonts – Mercury, Ideal Sans, Ringside Condensed and Tungsten – are now built into Tinderbox. These superb fonts are carefully designed for reading on the screen. The elegant **Ideal Sans Book** is now the default map font, and highly-legible **Mercury** is now the default text font. **Ringside Condensed** is recommended for adornments, **Tungsten** is great for dashboards where you want big numbers in a small space.

Guides

A new system of guides of kibbitzers works in Tinderbox maps to help keep your notes neatly aligned. In contrast to the old Tinderbox grid, these guides work to figure out what you're currently doing, and help you do it consistently. For example, if you have a note that's nearly centered between two other notes, Tinderbox will help align it precisely.

Quick Links

When typing in the text pane, you can quickly add a text link to a note by typing two left brackets – [[– and the initial letter of the note's name. Tinderbox will display a menu of notes with that initial; selecting a note from this menu will add a text link to that note. This is particularly useful for adding references to glossary terms, frequently-used sources, or oft-mentioned people and places.

ALSO NOTABLE

New application icon. Many thanks to strategy cartoonist Robert Black.

Actions

Actions and queries can now use information about composites. All are read-only unless specified.

compositeFor(theNode) returns a list of paths of notes in a composite.

compositeWithName("theName")returns the a list of paths of notes in a composite with a given name. If several composites have the same name, one of those composites is returned.

The path list returned by **compositeFor()** or **compositeWithName()** may be used as a designator, or to obtain additional information about the composite.

compositeFor(theNode):countreturn the number of notes in a composite.

compositeFor(theNode):namereturns the composite name. This expression is read.write: compositeFor(this):name="example" renames the composite.

compositeFor(theNode):role(theRole)returns a list of paths of notes in a composite, including only the note or notes with the designated role. For example, to set the color of all notes with the role “author”: $Color(compositeFor(great books):role(author))="red"

compositeFor(theNode):kindreturns the name of the composite from which this composite was instantiated. For example, if a composite was created from the built-in list composite, this function will return "list". The query **compositeFor(this):kind=="book"** would locate all notes that participate in composites instantiated from "book" – all books.

The expression **compositeFor(theNode):roles** returns a set of roles that appear in the composite.

In Tinderbox expressions, **my** is a shorthand for **compositeFor(this)**. For example,

```
$MyNumber=$Price(my:role(catalog))
```

find the note in my composite that has the $Role of “catalog” and sets $MyNumber to the $Price listed in that note.

$MyString.replace(pattern, replacement) returns a copy of $MyString in which every occurrence of *pattern* is replaced with *replacement*. If the replacement string is omitted, the one-argument form **$MyString.replace(pattern)** returns a copy of $MyString in which every occurrence of the pattern is removed.

The numeric formatting operators format() and **.format()** now allow an additional argument that specifies the padding character. For example: (493)

```
7.format(0,3) ➛ “ 7”  7.format(0,3,0). ➛ “007”  7.format(0,3,#). ➛ “##7”
```

Autocomplete: In the key attributes table and in Get Info:Attributes, autocomplete is far more flexible than before. For example, if you type “kin,”

autocompletion will now offer “Laurie R. King.” Previously, autocompletion required you to have typed the beginning of the value.

Badges: The Appearance Inspector now provides an image well into which you can drag images in order to add them as user badges. The inspector also provides a checkbox that lets you set $BadgeSize to “large” (64 pixels).

Email may be dragged from mail.app into Tinderbox.

Hover

A new attribute **$HoverImage** allows you to display an image when hovering over a note. $HoverImage contains the path to an image file. The easiest way to set $HoverImage is to drag the file into the image area in the Hover inspector.

A note may now have both a $HoverExpression and a $HoverImage. If a note has both, the value of the $HoverExpression is displayed in large type on top of the image.

When using *both* text and image, the text is drawn at a larger size using $HoverFont, which defaults to Tungsten.

The Hover pane of the Name Inspector now has a small button that clears the hover image.

Import

Notes copied or dragged from **DEVONthink** have their URL value automatically set to a special URL that DEVONthink Pro recognizes. If a note with a DEVONthink URL has $AutoFetch set, then when Tinderbox routinely autofetches Web data, it will also reimport the text and name of the DEVONthink note.

If a note’s AutoFetch value is true, Tinderbox now automatically updates notes which have (a) URLs accessible via the Web, (b) DEVONthink Pro URLs, or (c) file:/// URLs. This facilitates automatically updating notes from text files, including text files shared through Dropbox® or iCloud.

Map View

In map views, hold down the shift key while dragging to constrain movement to either the horizontal or vertical axes. Holding the shift key also disables guides, which are drawn in gray rather than the usual blue-green to show that they are inactive.

In the map view, a *link stub* represents a link to or from a note that does not appear in the map because that note is inside a different container. When a note has more than one outbound link to notes that are not represented in the current map, Tinderbox displays a link stub to represent the links not shown. If a note has more than one such link, the number of link stubs appears beneath the link stub. Clicking this link count reveals a menu of destinations from which you can select.

When a note has more than one *inbound* link to notes that are not represented in the current map, Tinderbox displays a link stub to represent the links not shown. If a note has more than one such link, the number of link stubs appears above the link stub. Clicking this link count reveals a menu of sources from which you can select.

In map view, ctrl-option-Return (**^⌥-Return)** now creates a note below the selected note. As before, Return creates a note to the right of the selected note, and ^+Return creates a note to the left of the selected note. If there is already a note where Tinderbox would place the new note, Tinderbox seeks a suitable location.

OnRemove: Containers, agents, and adornments now have an OnRemove action which is performed when a note is removed from the container. The OnRemove action for agents is applied to the original note, since the alias will be deleted. Note, too, that agents may remove notes and add them again at any time.

OnVisit: An early feature request proposed an action that would allow, for example, Tinderbox to speak the name or the text of a note when the note was selected. The $OnVisit action accomplishes this. For example, $DisplayName.speak("Moira"). will speak the name or the note, using Moira’s voice.

Tab bar: dragging a tab down from the tab bar once again opens a new window with that tab.

Text

When multiple notes are selected, Tinderbox displays the text of all the selected notes. This text cannot be edited, but it can be selected, copied, and pasted elsewhere. When text windows display the texts of a composite or multiple selection, a gray rule is drawn beneath each note.

Note ▸ Split will split a note in two based on the selection in the text pane. If the selection is empty, the note is split at the insertion point, otherwise the selection is extracted to a new note. The title of the new note is taken from the first sentence of its text.

When a document is reopened, any text windows that were open when the document was saved are also reopened. Text windows are moved in front of other windows lest they be hidden and forgotten.

WORKFLOW

When copying from an outline view with columns, Tinderbox now copies the column data for simple sharing with spreadsheets and other programs that use tab-separated values.

DETAILS

Actions

Tinderbox actions could be confused if two notes had identical paths, even if the designator was not ambiguous. For example, the rule

```
$Color(this)="orange";
```

For the note at path `/Trump` could turn the wrong note orange if the document contained two top-level notes, both named “Trump.”

A new operator **hasLocalValue()** lets you determine whether a note has a specific value for an attribute, or whether that value is inherited from a prototype or a default.

```
hasLocalValue("attributeName" [,target] )
```

Note that the attribute name should be enclosed in quotes and should not be prefixed with a $ sign.

```
hasLocalValue("MyString")
```

The arguments are evaluated, so

```
hasLocalValue($MyString)
```

returns information about the attribute whose name is currently saved as the value of $MyString.

The key attributes picker now ignores prefixing an attribute name with “$”. If you enter “$Width”, for example, the picker will silently correct the entry to

“Width.”

The query pane of the Action Inspector and of the Agents panel of Get Info now offers “==” as an autocompletion for “=”. This might encourage users to prefer the unambiguous comparison operation == to the ambitious assignment operator =.

Adornments

Adornments now correctly inherit the color “transparent” from their prototypes.

A new font attribute, $AdornmentFont, determines the font used by adornments. If $AdornmentFont is empty, then the value of $NameFont is used. The default value of $AdornmentFont is now Tungsten Medium.

Agents

Agents which do not use automatic cleanup methods no longer pile multiple notes in the same location.

Copying an agent and pasting it into a new document could crash, because Tinderbox might try to index the proxies for the pasted agent’s aliases in one thread while deleting them in another. Tinderbox now performs the deletion on the agent update thread.

Documents with numerous smart adornments could bog down because agent updates forced numerous animated layout updates on the main thread. Tinderbox is now more judicious about scheduling those updates and avoiding redundant layouts.

Smart Adornments failed to allow spaces between the notes they gathered.

Attribute Browser

Attribute Browsers that use category summaries now sort themselves by the summary value when appropriate.

When using the Attribute Browser to examine numeric attributes, you may now set the number of bins into which your notes will be divided.

Attributes: the default value of $TextAlign is now “left”. Formerly, the empty string represented left-alignment.

Breadcrumb bar gains a tool tip.

Browse Links now displays link labels for each link.

Built-in Prototypes: The built-in prototypes for HTML Template and Code now provides more tab stops.

Chart View

In chart views, expanding a collapsed branch sometimes failed to draw its children.

Composites

When opening legacy files created in previous versions, Tinderbox scans the file in search of overlapping notes that will now be interpreted as composites. This situation most often arises in documents that are seldom if ever used with maps. If Tinderbox finds apparently accidental composites, it marks the notes in the container as $NeverComposite.

Configuration

The configuration file config.xml in the Support folder now accepts an additional command line

```
<DarkenOutlineColors> 0</DarkenOutlineColors>
```

To turn off the **Darken OutlineColors** in new documents.

Document Settings

The Document Settings value for text alignment is now observed as the default for $TextAlign.

When the default text font is changed in **Document Setting: Text**, Tinderbox now scans the text of every note in the document and changes each usage of the former text font to adopt the new font family while retaining the current size.

Initial text sizes specified in Document Settings:Text were previously limited to specified “standard” sizes. Any reasonable size may now be chosen; in particular, 11 and 13pt text is now available. Some minor changes may be required in custom configuration and color scheme files. The method for specifying the default text size in config.xml files has changed. The old <TextSize> tag is superseded by <TextSizeNew>, which selected the text size in points.

Changing the default map background color, after having changed the default map font, caused the background color to be set to the error color.

Duplicate

If a note’s name is a number, duplicating the note now creates a note by appending the word “copy” to the name. Previously, the number was incremented. If a note’s name contained characters other than digits but ends in a digit, the number is incremented. For example, duplicating “42” creates “42 copy”, while duplicating “Catch 22” creates “Catch 23.”

Explode

Explode would hang when trying to use the delimiters ^, $, or *. These delimiters have special meaning as regular expressions, and the naked character would not be useful in the context of explode. For example, “^” matches the start of any text, which would generate an infinite number of empty notes. Instead, we recognize these special cases and silently replace these delimiters with their escaped equivalents \^, \$, and *.

If the explode delimiter contained an escaped character such as \n or \t, the next invocation of Explode would restore the character without escaping it. As a result, the delimiter field could acquire unexpected whitespace characters which confused subsequent use of Explode.

Find

The **Find Bar** now permits you to search a user attribute as well as Name and Text. Press the “User” button to turn on user attribute search, and select the attribute of interest. Dismiss the popover – double-clicking the chosen attribute is a convenient shortcut – and proceed to search.

The Find results window now shows the context for each text match. Matches for the note name don’t show context, as the name is already displayed.

Tinderbox crashed when opening a text window from the Find popover or Find window, if the destination note was not descended from the parent of the current view.

HTML Export

File ▸ Export Selected Note now respects $HTMLExportFileName. Previously, is exported to a file based on the note’s DisplayName, which would often differ from the file name used for a full export.

The HTML view pane now responds to the font panel, allowing you to set the font and font size used for viewing HTML. The font and font size may also be set using $HTMLFont and $HTMLFontSize. The default has been changed to Andale Mono 13.

Some HTML export error messages were enclosed in angle brackets. This wasn’t a problem in HTML, which tolerates unexpected markup elements,

but XML applications such as Scrivener import are less tolerant and can bail at the unexpected element. Better to avoid the angle brackets entirely.

The ^value()^ element no longer encodes non-ASCII characters in the attribute value.

Some users use a post-processor like Jekyll to process their Tinderbox files after export, and want to link to the *processed and modified* files that will be created after export rather than to the exported files. When constructing text links, $**HTMLLinkExtension** specifies the extension to be used for link, and replaces $HTMLExportExtension. For example, exported files might have $HTMLExportExtension of .md; these files would be processed to create .html files and contain links to the .html files that will be uploaded to the Web server.

The warning message regarding a missing or unspecified HTML export template in the HTML pane is now darker and easier to read.

The ^value()^ operator failed to evaluate expressions the refer to notes with non-ASCII names, such as ^value($DueDate(/Hückel))^, because HTML entitification was performed too early in the parse.

Get Info

The Attribute pane of the Get Info popover better accommodates long attribute names.

Import

When URLs are dragged into a view pane to create a note, the new note has $URL as a key attribute.

Pasting an alias into a different document would frequently crash, as the indexer tried to index the proxy for the unresolvable alias.

Items pasted or dragged from **DEVONthink Pro** now retain their tags. In addition, multiple items can be dragged or pasted from DEVONthink Pro.

Infrastructure

The **File ▸ Open Recent** menu now lists more files.

The quality of service priority of Change Updates has been demoted from UserInteractive to UserInitiated. This is probably more appropriate and may reduce unwanted interactions with animations during drags.

Extensive rewrites in classes that read documents. Large documents often load significantly faster.

Fixed a new crash after reverting, especially when multiple windows were open, which was caused by sending deletion notifications to the old document windows when loading the new hypertext.

A call to setUnderlyingQueue prevented 6.6.5 from running on macOS 10.9 (Mavericks). So Tinderbox 7 should support Mavericks as well as newer systems.

A regrettable hack named **updateRulers,** intended to workaround a macOS bug in updating the text ruler when the text pane is not selected, generated unwanted and alarming console messages. These should now be quieted.

An exception could occur when closing a Tinderbox document, with logs indicating an attempt to access a menu item beyond the end of a menu. This could, in turn, lead to failure to properly close a window; when the window was closed a second time, Tinderbox would hang. The underlying problem lay in ColorMenus, and should now be addressed.

Fixed a minor memory leak when drawing badges.

Fixed a minor memory leak that arose when constructing the AttributeBrowserModel, because each category in the model helps a strong pointer to the model.

Resolved a retain cycle with the Parking Space view, its Help controller, and its popover. This now restores two disabled tests in DeletionTests.

Resolved minor memory leaks when importing text files, opml files, and rtf. Corrected an error that prevented properly recognizing RTF data on the clipboard.

Fixed a crash when creating an agent in a map view, if the container set the agent's prototype.

New IDE, compiler, and tooling.

Inspector

The inspector is no longer hidden when Tinderbox is not the foreground application, making it easier to drag images into the Appearance Inspector's badge well.

The menu command **Window ▸ Prototype Inspector (⌘-3)** provides easier access to the prototype inspector.

The Link Types inspector gains popup menus for named colors.

Fixed a cosmetic glitch in the Action inspector by making the inspector window about ¼" wider.

Corrected a crash when updating the document inspector's colors pane when no document was active.

The Plot Expression field of the Plot Inspector is now hidden unless a plot type is selected.

If no note is selected, the appearance inspector no longer displays color wells with the error color.

Quickstamp now offers an autocomplete menu for string, set, and list values.

The Quickstamp Inspector gains a pulldown *values* menu that lets you select from any current value of string, set, and list attributes.

The Tinderbox Inspector's Info pane reports the total number of links in the document. This count no longer includes Prototype links.

Key Attributes

URLs dragged from Safari may now be dropped on URL attributes in the key attributes table.

The calendar button was sometimes disabled for key attributes when it ought to have been enabled.

If you need lots of key attributes, the token field of the Key Attributes will expand to accommodate the list as it grows.

When typing in the key attribute picker, if your partial typing excluded all but one existing attribute, Tinderbox automatically selected that attribute. This is over-eager, since you might want to define a new attribute.

if you are adding attributes to the Key Attributes table and enter an invalid attribute name, Tinderbox no longer displays a popover for creating that attribute. Previously, the popover appeared, but since the attribute was invalid, it offered no attributes to create.

The calendar button was sometimes disabled for key attributes when it ought to have been enabled.

Files may now be dropped onto URL attributes as well as File attributes.

Autocomplete for **set and list attributes** failed to operate beyond the first element of the set or list.

A common convention in *Getting Things Done* represents work contexts by prefacing them with the character "@". When editing the values of set or list attributes, autocompletion regards the character "@" as starting a word.

Removed AIM from the Person prototype's $KeyAttributes.

Percent-encoded URLs in key attributes were double-encoded before being dispatched to the browser.

The width of the key attribute value column is no longer limited to 1000 pixels, and so the pulldown menu icon will appear in the expected place in very wide windows.

The key attributes table's View In Browser button is more tolerant of URLs that contain non-ASCII character.

The key attributes table is again hidden when no note is selected. In addition, the key attributes picker button is hidden when no note is selected.

The date picker is no longer displayed for key attributes that are read-only.

Links

If a note's text contains "smart links" (e.g. links automatically created by the system from URLs), those links are converted to Tinderbox links when the note is next selected.

In the link popover, the "swap" button looks more like a button, and gains a tool tip.

Links that do not override the link color of their link type now continue to inherit that color, even after the document is saved and reloaded.

Dismissing the link parking space with the Esc key could attempt to make a link, despite the cancellation, if the destination that had been typed corresponds to an existing note.

Corrected erroneous link colors observed after a new link type had been defined.

When no link is selected in the Link Browser, the link terminator popup shows the common case (arrow) rather than the less common value (circle).

Map View

Notes newly dragged into a container are now placed and drawn correctly. Previously, child notes might not be drawn immediately, and child notes were placed incorrectly when drawn into a note that had no children, because the height of the title bar was treated incorrectly.

Use **Note ▸ Navigate (⌘-Return)** to follow this first basic link from the selected note in map view. This can be invaluable in presentations and meetings, facilitating easy movement to new topics.

Duplicate (⌘-D) placed the new copy in the same position as the note being duplicated. Note that option-drag may now be a more convenient way to duplicate notes.

It's easier to edit subtitles in map view; previously, clicking in a selected subtitle selected the entire subtitle.

When displaying pie charts, the legend was misformatted if the pie chart's container also displayed a summary table.

In map view after zooming into a container, links were sometimes drawn in the wrong place or the wrong links were drawn, thanks to a race condition between the link animation and the zoom in animation. This long-standing cosmetic issue should now be resolved. The update is now split between willSetParent and didSetParent methods to prevent the race condition.

A variety of detailed enhancements reduce CPU load while grabby-hand scrolling and mouse-wheel scrolling in maps.

In maps containing one or more containers that hold hundreds of notes, dragging notes could be very slow because Tinderbox was doing unnecessary sorting of the outline during the drag, and sorting a thousand notes can be slow. Additionally, Tinderbox was recalculating the size of child notes during the drag, which involved a great deal of unnecessary work.

The tendency of some Tinderbox maps to twitch slightly after a new note is added has been diagnosed and corrected.

In the view pane, **option-drag** duplicates the note and drags the duplicate. **Shift-option-drag** makes an alias of the note, and drags the alias.

In map view, Tinderbox sometimes allowed too more vertical space for the title than was actually needed, creating a gap between title and subtitle.

The wrapping rectangle for drawing text in map items requires less minimum width, improving the appearance of smaller notes.

Various guides, including the aspect-ratio guide, that affect properties internal to the composite, are no longer displayed while dragging a composite.

The link count displayed in inbound stub links incorrectly included prototype links.

After items are dragged in the map, Tinderbox immediately restores them to their proper layer. Dragged items must be temporarily moved to the foreground; if they were not, the item you were dragging might be hidden beneath other notes. Tinderbox was insufficiently proactive in restoring the layering, which gave rise to transient anomalies.

The map contextual menu now offers a **Create Composite** submenu.

The outbound link count in map view is now drawn using lining figures, which are better-aligned with the boundaries of the circle in which they are drawn.

Clicking on a locked adornment did not select the adornment. This made some sense – there's not a lot you can do with a selected adornment when it's locked – but one of the things you ought to be able to do is unlock the adornment, and that was unnecessarily difficult. Now, you can select the locked adornment but you can't drag or resize it until you unlock it.

When an adornment action changes the prototype, the LinkAnimator can crash because the LinkAnimator still has a pointer to the old, deleted prototype link. We are now careful not to take action during the drag, and to reset the link animator on mouse up.

Improved map performance by reducing unneeded refreshing of tabs while dragging notes.

Changing $InteriorScale now forces a prompt new layout. Previously, the change only took effect after the layout had been updated.

Menus

The **View ▸ Align** menu is now named **View ▸ Arrange**.

Move Note Up/Down and **Send to Front/Back** commands have moved from the **Note** menu, which is overcrowded, to the **View ▸ Arrange** menu.

View ▸ Cleanup also moves to the **View ▸ Arrange** submenu (1859)

The note contextual menu now includes a submenu with all your stamps.

Distribute Horizontally, now in the Arrange menu, is now implemented correctly. The leftmost and rightmost notes remain fixed in place, and other notes move so the spacing between notes is equal.

Outline View

The menu command **Note ▸ Create Note** could auto delete the newly-created note in outline view if the text rule was being displayed.

In the view pane, **option-drag** duplicates the note and drags the duplicate. **Shift-option-drag** makes an alias of the note, and drags the alias.

Corrected the measurement of item heights in outlines, which occasionally chopped off the last line of an outline item.

Notes created in outline view are automatically assigned plausible map positions. In the past, if a container had numerous children, Tinderbox would sometimes give up the search for a plausible position and place all new notes at {0,0}. This should no longer occur. Extensive revisions to note positioning should reduce or eliminate Tinderbox's tendency to pile notes atop other notes when working outside map view.

Marquee selection has always worked in outlines, but pressing the option key in outline view failed to display the "+" cursor as it does in map view. The "+" cursor now appears when the cursor is not over an item – in the left margin of the window, and beneath the last item.

In documents that use Darker Colors in outlines, the selection highlight now also uses the darkened color.

In outlines, separators now respect document settings for Darker Colors and Black Titles. Formerly, the separator line was not darkened.

Roadmap

When links are added or removed from a document and a torn-off Roadmap window is open, the link counts in the roadmap are updated.

Stamps

Applying a stamp now refreshes the text pane, in case the stamp changed the title, key attributes, or text.

Stamp inspector now selects more sensible key views after adding or deleting stamps. **Tab bar:** the tab bar sometimes failed to show tabs on opening, or failed to show a newly-created tab, because the animation showing the tab appearing interfered with the animation showing the other tabs sliding into place.

Text

Tinderbox again respects $ViewInBrowser; if true, and if $URL is a valid network or file URL, then Tinderbox will automatically open that URL when the note is selected.

If an image pasted into the text is wider than available space, it will automatically be scaled. Previously, the image was displayed at full resolution but cropped.

The value of $TextAlign is now used when (a) editing an empty note, or (b) after applying **Format ▸ Style ▸ Reset Margins** to the note.

When a new note is selected, windows that are not the frontmost window no longer update their text pane to reflect the selection change. This makes working with multiple windows simpler.

A note that had a $TextFont representing a font not present on the system could crash in Content::StandardStyleFor() because it tried to insert a nil in a dictionary.

The text pane's title field always contains the $Name of the note and allows you to change the name. In the past, moving from editing the title to activating the text area left the $DisplayName in the title field.

The title field in the text pane is drawn in the note's $Color. This is unhelpful of $Color is "transparent"; in that case, the title is drawn in black.

Resolved visual artifacts in the text pane beneath the title and above the text area when initially opening a document. These arose from clashing animations while adjusting panes and choosing the size of the key attributes table.

Renaming a note now updates the title of any text windows open for that note.

Fixed a crash in Format ▸ Style ▸ Standard Font when $TextFont was mistyped or not installed.

Timeline: changing a note's $StartDate, $EndDate, or $DueDate now refreshes the timeline layout.

Tinderbox 6

Changes during Tinderbox v6.x.

Version 6 releases are listed below, in reverse chronological order.

6.6.5

DETAILS

Correct handling of escape key to cancel note creation in outline view

Correct bad link to release notes

Other minor fixes

6.6.2

HIGHLIGHTS

FIND

In the Text pane, Edit ▸ Find ▸ Use Selection for Find now loads the find string for both the text pane search bar and the view pane search bar.

The Find window now has a checkbox that allows you to search for aliases, or only to show results for the original notes. Formerly, Tinderbox always searched aliases.

MAPS

guides

The guide system has been rewritten, providing a foundation for research on Tinderbox 7. For the present, new guides help you to:

- align the vertical and horizontal centers of notes
- maintain uniform spacing between adjacent notes or place notes next to each other
- align the edges of notes
- maintain spacing between notes and the edges of the adornment that contains them
- place a note symmetrically between two horizontally-adjacent notes
- create square notes and notes using the golden ratio
- align left or right edges of notes to the midpoint of notes above or below them.

A new set of guides look for equal spacing between adjacent sets of notes. If we have two adjacent notes, A — B, and then move C next to B, the guide

will make the space between B and C the same as the space between A and B.

A new guide examines the neighborhood around notes with a circular shape, looking for other notes that are nearly the same distance from the circular note as the note being dragged.

The guide for centering a note between two other notes formerly considered only the notes closest to the dragged note. It now looks for matches among all pairs in the vicinity of the dragged note.

links

You can now drag either the source or the end point of a link to lock it to a specific edge of the source or destination note. Dragging the source or end to the map background unlocks the link, in which case Tinderbox will select the starting and ending locations automatically.

The Link editor also lets you choose the start and end points for links. The default setting, automatic, continues to choose the start and end points as before.

additional map improvements

Shift-drag constrains the note to move only horizontally or only vertically.

Expand Horizontal was inclined to be carried away if the height of the note was less than the height of a single line.

View ▸ Map now restores the scroll position of the map if the current container has defined a scroll position.

The number of outbound link stubs, which represent links from a note to notes not visible in the map, are now displayed in the map if the stub represents more than one link.

When dragging a link label, the boundaries in which the link label may be dragged are now displayed. The boundary link grows in intensity as the label approaches the boundary to avoid adding unnecessary distraction.

You may now set $NameLeading to a negative value to tighten the spacing between lines.

Items in maps are very slightly larger.

Text layout in non-rectangular shapes was not calculated correctly, causing unnecessary truncation.

PRINTING

Text pane printing now respects the right edge of the paper, when the text pane is wider than the page.

ROADMAP

Roadmap: if we double-click to follow a link, and if the destination is not visible in the current view, Tinderbox will now refocus the view to make it visible.

NOTABLE

DATES

New ways to specify the time for dates include:

- today 0930
- today 930pm
- today 11

EXPLODE

Corrected the handling of titles when the title is to be removed from the text.

EXPORT

^url(original)^ exported a spurious URL if the original note was not exported. It’s easy enough to export the URL that would be generated of the note were exported, and this allows consistent export when part of the export is temporarily disabled.

HTML export now considers a note that contains only white space to be empty.

$HTMLQuoteHTML was wildly over-enthusiastic, and quoted the markup that Tinderbox generates as will as the markup it ought to have quoted.

FOOTNOTES

Fixed a problem in following recently-created footnote links.

Footnote As Child did not account correctly for 2-byte Unicode characters.

INFRASTRUCTURE

Intermittent and infrequent crashes have been reported in complex documents that use agents, rules, and sorting extensively. Sorting based on synthtic attributes like $DisplayName and $WordCount has been especially vulnerable. This can arise when several modules step on each other’s toes when modifying a document simultaneously. A new work queue coordinates sorting to reduce interference while preventing rules and agents from slowing down your work.

LINKS

When first creating a web link, the URL field and label were improperly hidden in the link popover.

Avoid potential crash when deleting a link in the Browse Links popover while animating that link in a map view.

Browse Links: checkbox titles for link styles (bold, broad, etc.) are lowercase, as in inspector.

Browse Links sometimes failed to enable some controls, including the link type selector, of links created in the current session, because newly-created links could have uninitialized source and destination pad designators.

The opening animation of the link creation popover is no longer interleaved with the shrinking the popover to its small state

TEXT

When editing text that follows an automatically-recognized URL, the insertion point was spontaneously reset to the start of the note.

When creating a note, clicking on the text pane without first hitting Return now confirms the rename and allows you to select the text pane.

MISCELLANY

Tinderbox could crash after closing a document, when the current view is complex and the document had active agents or rules that frequently updated the screen. The crash was caused when the change management queue was not shut down soon enough.

Fixed a possible crash after deleting links in the link popover, since the link animator could potentially animate the deleted link before the LinkInfo was updated.

In the Appearance Inspector's Shape menu, the symbol for the tag shape was reversed.

The obsolete "tabs" chapter has been removed from Tinderbox files. Incremented file format version level from 2.9 to 2.10.

The maximum size of the values list in the Key Attributes tables was not being enforced. This could cause sluggish response in very large documents.

Fixed an intermittent crash on mouseUp after resizing.

Selecting "custom" background color from the map background popover selected errorColor as the background color.

Date parsing: the date "tomorrow 8" was previously interpreted as 8 days from tomorrow. Now, it will be treated as 8 o'clock tomorrow. "Tomorrow+8" continues to denote 8 days from tomorrow.

Revised link layout to respond better when the link must run through the note to reach the designated start or end point.

The color of links created using new link types was drawn incorrectly as the error color.

In the link creation popover, the URL field only appears for Web links.

Corrected a variety of user interface anomalies in the new link creation popover.

Spreadsheet import no longer copies the entire spreadsheet into the container's text field. This was not useful and, for large imports, caused performance problems.

Outline tabs again remember their scroll position.

Turned on the value deletion pool, trading some memory footprint for possible stability gains. Stay tuned.

Agents: when an alias is edited so that the item no longer meets the query, the next alias is selected in main view but the text pane was not refreshed.

Fixed a startup crash when checking registration code if the Preference file is damaged.

The text focused vanished mysteriously from newly-created notes in outline view, because an asynchronous text update added recently to force the update to the main thread was now coming after the focus had been moved to the new note. This had the effect of cancelling the initial transient editing session. The blame ultimately lies in the horrible updateRulers hack, but for now doing the text update synchronously if we're on the main thread will suffice.

Resolved some very intermittent crashes when closing, caused when the link Animator tries to animate while it is being disassembled.

Inspectors failed to notice some selection changes.

Clarified Tinderbox Help on the distinction between **linkFrom** and **linkFromOriginal**.

6.6.0

HIGHLIGHTS

INFRASTRUCTURE

Tinderbox uses your computer's multiple processors to update rules and agents behind the scenes and to keep your work organized and sorted. Tinderbox 6.6 introduces important new strategies for keeping the dance moving gracefully and to avoid having partners step on Tinderbox's toes.

Implemented a new approach to saving window information when closing documents, in order to ensure that the saved file will describe all windows that were open when the document was closed.

EXPORT

When exporting text or RTF, Tinderbox now interprets several useful markup elements.

^include(which)^ is replaced by the styled text of the designated note.

^value(expression)^ is replaced by the value of the expression. For example, ^value($Width)^ will be replaced with the value of the width of the exported note.

^if(expression)^ ... ^endIf^ and ^if(expression)^ ... ^else^ ... ^endIf^ permit conditional export; the material between ^if()^ and ^endif^ will be exported only if the expression is true.

IMPORT

Microsoft Word® files (.doc) and Word XML files (.docx) may now be dragged into Tinderbox.

Taskpaper files may now be dragged into Tinderbox. Tinderbox automatically recognizes Taskpaper tags and copies them to $Tags. The @done() tag also sets $Checked and $NameStrike; if the @done tag has a completion date, it is copies to $StartDate and $EndDate.

When importing spreadsheets, comma-separated tables, and tab-separated tables, the first column typically contains the name of each note. Tinderbox now allows you to override this; if a column is has the title "Name," that column will be used at the name column and the first column will be treated a a data column.

When importing spreadsheets, comma-separated tables, and tab-separated tables, if a column is named Prototypes, then prototypes will be assigned to the new notes. If a prototype does not already exist, a new prototype will be created in the Prototypes container.

KEY ATTRIBUTES

A new date picker is available in the key attribute table and in the Get Info attributes pane.

MAPS

Link types (and the Link Type Inspector) now let you choose a connector type. New links of a given type adopt the type's connector, and you can override the connector type for individual links as before.

Dragging the background of a map container now continuously updates the screen during the drag.

Dragging items out of containers is easier and more reliable, and moving among tabs and containers retains scroll position more intelligibly. When moving into and out of containers, maps do a better job of respecting the container's scroll position.

Duplicate is somewhat more considerate in choosing a name for the duplicated note. In particular, if the note name already ends in a number, that number is incremented rather than appending the word "copy".

NOTABLE

ACTIONS

A new action function, **inheritsFrom()**, checks whether a note uses a specific note as a prototype, either directly or through other prototypes. For example, suppose Flower has the prototype Plant, and Rose uses the prototype Flower. Then

```
inheritsFrom(/Plant)
```

is true for both Flower and Plant.

```
inheritsFrom(/Flower)
```

is true for Rose, but false for Plant. You can also write

```
inheritsFrom(which, prototype)
```

which returns true if the note *which* inherits from the note *prototype.*

The operators sum() and avg() now work as expected with interval attributes.

In some cases, we might prefer to display only intervals that are non-zero. $MyInterval.format() and format($MyInterval) display the interval without change, except that the interval 00:00 is displayed as the empty string. (1772)

In agents, back-references in regular expressions were not handled correctly; some notes received back-references from a different note's match.

Code fields offer autocompletion for dot operators. For example, typing **$MyNumber=$MyString.c** will offer a menu of completions that includes **.count**.

$InboundLinkCount and $OutboundLinkCount now report separate values for the alias and original note. (1769)

EXPORT

$HTMLQuoteHTML and the **Quote HTML** checkbox in the Export Inspector were not always observed in text markup.

Changing the Quote HTML checkbox in the Export Inspector will now refresh the export display.

GET INFO

The word count reported in Get Info no longer counts the contribution of aliases.

MAPS

In ring plots, the target is now drawn in $PlotBackgroundColor. It was formerly always black.

The text color used in prototype tabs is normally the background color of the map. This usually works well, but can be difficult to read in low-contrast color schemes where the note color is close to the background color. If the two colors have similar luma, Tinderbox instead chooses either black or white.

The "dotted" button in the link type inspector was incorrectly setting links to "broad".

Edit ▸ Duplicate in map view chooses a better location for the duplicated note. So does **Create Note**.

NAVIGATION

Note ▸ Go Back is now disabled when the history is empty, because there's nothing to go back to.

Note ▸ Cover Page now resets the the visit history.

OUTLINES

Revised the behavior of forward-delete (fn-Delete) in outlines. This key now deletes the selected notes and then selects the first younger sibling of the selected note that is not an adornment.

Interval attributes were drawn incorrectly in outline columns.

You may now "hoist" or focus on notes with no children. This can be useful when you want to create a container and then want to add notes to the container.

ROADMAP

The Inbound Links list in Roadmap failed to italicize the display name for aliases. (1769)

Torn-off Roadmap windows are now updated after a link's type has been changed. (1770)

SIMPLENOTE

Corrected extra escaping of simplenote tabs.

Double straight quotes could, when synced, acquire unwanted backslashes.

TEXT

After choosing Window ▸ Text Only, the keyboard focus will switch to the text pane. After choosing Window ▸ View only, the keyboard focus will switch to the view pane, or to the selected item.

If changing the prototype to one that has a distinct font but also has no text, the note's typing attributes would not be updated until the note was deselected and reselected. If you type in the note, the text would retain the attributes current before the prototype was changed.

Browse Links is again available when the selected note has links, even when the text pane has the focus.

Get Info is available when the text pane has the focus, and is available even if no text is selected.

Activating Tinderbox from the background no longer scrolls the current text pane to the top.

b203 had prohibited ^include(this,*template*)^. That's too restrictive.

The text pane title field now follows the document settings for **Darker Colors** and **Black Titles** found in the Outline pane.

Enable View ▸ Text Window when text pane has focus.

MISCELLANY

Resolved a crash in 6.5.0 that arose when opening a document in which the selected note's rule or edict forced an update to its key attributes table.

Reordering the columns in a summary table could crash if Use Headings was disabled or there were no headings.

Tinderbox runs on OS X 10.9

Updated layouts of many views and popovers.

Improved link creation popover.

The Document Inspector counted agents with very low priority as disabled.

Automatic agent updates could be silently suspended after autosave, causing agents to fail to register changes until forced to do so.

Note ▸ Go Back could crash when the view pane had the focus, if after navigating to a new note the user had changed the selection and no note was selected.

A possible crash when assembling timelines in the presence of agent updates has been corrected.

Pressing Esc to cancel editing the name of a new, untitled note deleted the next note, not the new note.

The insertion point in the key attribute table could be drawn in the note's $Color. It is now black.

Unwanted target values were sometimes drawn in ring indicators.

Revised the primitive operation that sets attributes, Node::DoSet(), to use the value pool for deletion.

Tinderbox could hang when closing documents while updating the Quickstamp pane of the inspector, because it failed to check whether, in the now-current document, the attribute chosen in Quickstamp actually exists. If it does not exist, checking whether the current value is an immediate value was problematic.

Tinderbox could hang when opening documents with a prototype cycle because Quickstamp inspector's updateInheritanceLabel: did not guard against this.

Fixed a variety of potential crashes on revert.

When reading a document, we no longer attempt to dispatch to a new thread when resolving aliases. This does not generally save time, and may cause occasional crashes if, for some reason, the secondary thread is delayed.

Add Footnote As Child now uses the appropriate path names for links. Formerly, both links were untitled.

The initial selection in the Get Info pane list is now completely visible; previously, if "words" was the initial selection, only part of the title was visible.

Problems with Autocomplete and the link Parking Space popover appear to have been corrected.

Spurious checkmarks in **Document Settings:Map:If note name is too long** popup have been corrected.

Fixed a crash when re-creating the agent queue if the agent queue wasn't empty. (

Rewrote CachedAttribute to address intermittent problems when sorting or when evaluating DisplayExpressions or Rules that involve computed attributes like $WordCount in different threads. CachedAttributes keeps a copy of their former value and delete it when creating a new result, thereby avoiding a memory leak. This could cause occasional problems for concurrency, where one thread might delete another thread's cached Value object. The new version defers the actual deletion to reduce or eliminate this interference.

Rewrote StChanges to avoid a rare concurrency error that can sometimes delay updates to the view and/or the inspector.

Rewrote StChanges (again) to eliminate a race condition which sometimes left Tinderbox views in a state where updates were delayed.

The key attribute disclosure button is hidden if a note has no key attributes, as there would be nothing to disclose

Duplicate selected the new duplicate, but left the text from the original note in the text pane.

Adding a built-in prototype to a document that had a top-level adornment named "Prototypes" caused many problems, as Tinderbox tried to add the prototype as a child of the adornment.

The prototype menu no longer permits you to create a cycle, where A inherits from B and B inherits from A.

LeftMargin and RightMargin are no longer Preference attributes. The default value of $LeftMargin may be set in the system attribute inspector; $RightMargin is ignored and the left and right margins are always identical.

If we copy and paste a container within a document, the notes in the pasted container will use the same prototypes as do the corresponding notes in the source container.

Dates prior to 1 AD/CE were not saved correctly.

6.5.0

HIGHLIGHTS

GET INFO

Common Words Get Info uses a new and better visualization method. Users can also choose a scope to examine: the selected note, the note and its descendants, all descendants of the note's container, or the entire document. The font used by Common Words is the NameFont of the selected note.

MAPS

Link labels may be dragged a short distance from their nominal position in order to improve their appearance.

A new link style, **broad**, is now available.

A new plot pattern, **ring()**, is now available.

> **ring(value[,min,max,target])**

In the simplest usage,

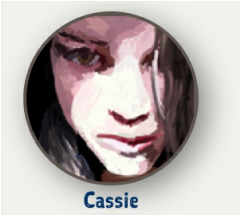
```
ring(33)
```

displays an arc representing 33% of a complete circle.

Optional arguments allow you to specify a minimum value, a maximum value, and a target value.

Notes with non-rectangular shapes may now use the bar, vbar, pie, and ring patterns.

Large badges with $BadgeSize > 32 now respect the shape of the note and are clipped to its border. Small badges, as before, may extend outside the drawn region of the note.



NOTABLE

Tinderbox now requires OS X 10.9.

ACTIONS

The new string operator **.find** returns the location of the first occurrence of a substring. If $MyString contains “We hold these truths to be self-evident”, **$MyString.find("hold")** will return 3. If the string is not found, **.find** returns -1.

Intervals now provide **day, hour, minute,** and **second** operators for converting the interval to familiar units. For example, suppose $MyInterval is 12 hours. $MyInterval.day will be 0.5, $MyInterval.hour will be 12, and $MyInterval.minute will be 720.

Numeric computations now use greater accuracy, which will reduce rounding errors.

Expressions of the form **date($MyString)** now evaluate correctly.

A new string operator, **$MyString.countOccurrencesOf(*pattern*)**

returns the number of times that *pattern* appears in the string. If $MyString contains the word “aardvark”, then **$MyString.countOccurrencesOf("a")** returns “3”.

Date canlculations could be off by one day when converting references to dates before the introduction of time zones, if the Tinderbox user’s local date differed from the date in Greenwich. This should no longer occur. (1699)

Expressions of the form

collect($MySet,$MyNumber)

were not evaluated correctly because the first argument was not evaluated.

Code field autocompletion recognizes typing “Prototype=” and offers suitable autocompletions.

If an each() clause changes a local variable in its enclosing scope, that change now takes effect. Formerly, each() clauses could read but not change local variables in the enclosing scope.

Back-reference found by .contains() regular expression matches are now available in **if()** clauses. For example, the rule

```
if($Name.contains("a(..)") ) {$MyString=$1}
```

will now set MyString to “pp” if $Name is “apple”, or to “rs’ if $Name is “pears”. Note that $0 contains the full matched expression.

COLORS

Unwanted colors can now be omitted from color menus and lists. To hide a color, select it in the Colors pane of the Document inspector and set the checkbox, “Hidden.” Hidden colors can still be used in rules and actions, and their definition can be changed in the Colors pane of the Document inspector.

EXPORT

Outline export now respects $NameBold.

In HTML Export, Tinderbox now exports images embedded in notes. Note that the images are now exported into the same directory as the note’s HTML file; in Tinderbox 5, notes were exported to a subdirectory.

When exporting more than one image, HTML export placed the second and subsequent image tags in the wrong place because it failed to enumerate images correctly, and because a __block variable was not initialized correctly. In addition, the image attachment entity character is now longer embedded in the exported HTML. (1690)

When not exporting titles in text export, Tinderbox hides the controls related to title export.

Normally, Tinderbox will not create an exported file if a note has nothing to export. However, if the note has an $HTMLExportCommand, the command is run anyway, because we might be obtaining the exported data from that command. However, if the result of $HTMLExportCommand is still empty, Tinderbox will no longer create an empty file. If the results from running the command do not change the exported file, the export file will no longer be marked as modified.

Outline export, when exporting doc or rtf files, now uses list styles. The export sheet also offers a popup that offers Harvard outlines and bullet lists in addition to numeric outlines.

FIND

In the Find window, headers now indicate how many results were found.

GET INFO

In the Get Info pane list, the **Common Words** pane is listed as “words” rather than “common.”

The **Info** pane of the Get Info popover now displays the note’s position and dimensions.

In the Get Info popover, you may now tab from the pane list to the controls of the current pane.

Tab can now be used to navigate among fields in the Agent, Book, Map, and URL panes of Get Info. In the Attribute pane of Get Info, tab activates the search field rather than the category list.

Similar words: double click on any item to dismiss the popover and select that item.

IMPORT

Dragging a folder from Finder into Tinderbox now creates a container and imports a note for each file in the folder. The $File attribute of the imported note holds the path from which the note was imported. The $LastFetched attribute of the imported note holds the modification date of the imported file.

HTML files, when dragged into Tinderbox, are formatted and imported as styled text.

Freemind files may again be dragged into Tinderbox. If a Freemind item has an associated note, that note is now imported as styled text in the Tinderbox text field.

Tinderbox now accepts .csv (comma-separated value) files as well as .tsv (tab-separated value) files.

Spreadsheet import now accommodates CSV files with old-fashioned CRLF line endings.

Modernized code for importing events dragged from the calendar.

INSPECTORS

In the Plot pane of the appearance inspector, the text field for the Target Value is hidden for pie charts.

MAPS

If Document Settings:Maps requests that Tinderbox should select the link destination after creating a link, then **Note ▸ Go Back** will now returns the selection to the link source.

Expand Horizontal is less aggressive in expanding notes where $Height is so small that the title might be clipped even though it doesn’t wrap.

Tinderbox attempted to draw names of adornments that were far too small to display them. Names are now hidden for adornments and notes if the available space is far from sufficient.

While blind-typing to locate notes in the view pane, the delete key now deletes the most recently-typed character, not the selected note. The esc key cancels blind typing.

When creating a link to or from an alias, the link creation popover allows you to link either to that alias or to the original note.

Improved (again) the routing of links in close quarters, where two linked notes are located close together. Improved placement of link labels.

In map views, dog ears were drawn even if the note’s $Opacity was zero. Dog ears now obey $Opacity.

In map views, note title and text were drawn when $Opacity was zero if their color was “automatic”, but not if a specific color was specified. Automatically-chosen colors now observe $Opacity.

In map view, dragging a note into a container makes it the container’s first child. To make the dragged note the last child of the container, hold the command ⌘ and shift ⇧ keys while dragging.

Map view hover expressions are no longer truncated if their text is forced to wrap.

In map view, icons are now clipped to the boundaries of the item. Layout errors have been corrected, and this leads to better behavior when resizing notes.

To suppress drawing text in maps, set $MapBodyTextSize to a negative value. The value of 1.0 also suppresses drawing body text.

Expand Horizontally expanded incorrectly when the map scale was not the standard magnification, because font scaling was applied twice.

Blurred borders work once again. These are chiefly of interest for adornments.

Image files dragged into maps once again create image adornments.

Stubs are no longer drawn in map view for web links. (1719)

A new attribute **HoverOpacity** allows you to specify the opacity of hover expressions on a scale of zero (invisible) to 1 (opaque). The Hover pane of the Text Inspector now has a slider to control opacity.

OUTLINES

In outline view, the left-arrow key selects the parent of the selected note. If the outline view is focused on a container and that container is selected, the view is now expanded to focus on the parent container. If the outline view is expanded to include the entire document and the selected note is at the top level, left-arrow has no effect.

Outlines layout is now updated after ending an editing session by selecting a new item. Previously, the layout was only updated if the user pressed Return to end the editing session.

Improved scrolling performance in large outlines.

In a hoisted outline, if the parent note is selected, Create Agent and Create Separator created siblings of the hoisted note, and thus the newly-created notes were outside the current view. The new items are now added as the eldest child of the hoisted note.

The Outline pane of **Document Settings** now offers a control for outline leading – additional space between outline items.

In outlines, forward-delete (fn-Delete) now deletes the *next* note in the outline.

Outline layout corrects the calculation of the available width; a 2px error occasionally failed to anticipate the need to wrap the text to an additional line.

PRINTING

Print should now respect paper choices in Page Setup.

SIMPLENOTE

Notes added to the Simplenote container in Tinderbox are now sync’d to Simplenote when the document is saved. Previously, notes could only be created in Simplenote.

Note text with embedded newline characters (e.g. notes copied from web browsers) could be sync’d with \n replacing the newline characters. This should no longer cause problems. Simplenote sync also failed to unescape escaped quotation marks and backslaches in json strings.

Users may choose a tag (or list of tags) they wish to sync in Document Settings ▸ Accounts. If this list of tags is not empty, only notes with one of the corresponding tags will be synced to Tinderbox. If empty, all notes are synced.

Simplenote synchronization no longer sets the key attributes of imported notes.

SORTING

When sorting notes alphabetically, Tinderbox now uses the prevailing locale's sorting rules for handling diacritics and accents.

TAB BAR

If the Tab bar is hidden, moving the mouse near it will reveal the tab bar. Moving away hides the tab bar again, but only after a brief delay; this delay lets you use the breadcrumb bar if you wish. Previously, the immediate hiding of the tab bar meant that Tinderbox played "keep away," snatching the breadcrumb bar away as soon as you approached.

Clicking in the tab bar to dismiss the Treemap options popover could switch tabs before the popover was closed, leaving the clicked tab as a treemap. The dismissing click is now ignored.

View ▸ Tab ▸ New Tab is now available when no note is selected. Previously, it required a selected note. (1711)

TEXT

The text pane can now be collapsed completely; previously, a very small residual width was always (barely) visible. (1645)

The Text Pane Selector allows you to view the note's text, HTML, and to preview its exported HTML. Some Tinderbox projects don't need to export HTML at all; **Window ▸ Hide Text Pane Selector** will hide this control. If the control is hidden, **Window ▸ Show Text Pane Selector** will restore it to its customary place.

Boolean attributes no longer interrupt the flow of keyboard editing of key attributes. To *change* the value of the selected checkbox from the keyboard, press the space bar. (1649)

When first opening the document, if the text pane selector is initially hidden, hiding it is deferred slightly to allow you to see it sliding away, helping to confirm that it's deliberately hidden (and suggesting that there's a way to recover it!). This also avoids artifacts from trying to animate while the window is being assembled.

Selecting the text pane removes search highlighting. Previously, it was possibly to begin typing at the end of a highlighted passage and for the new text to be permanently highlighted.

After changing tabs, the text pane slider always allowed at least 1px of text pane; this is no longer required.

When editing text, option-click clicks on the text but ignores any text links.

The key attributes table now respects $KeyAttributeFont.

Clicking in the left margin of the text pane selects the adjacent line. Double-clicking in the left margin on the text pane selects the adjacent paragraph. Shift-clicking and shift-double-clicking extends the current selection to encompass the adjacent line or paragraph.

Format ▸ Style ▸ Standard Size and Format ▸ Style ▸ Standard Font are now available when the map pane has the focus. In this case, the command is applied to the entire text.

Format ▸ Style ▸ Standard Font is now more aggressive, removing indentation, background colors, text colors, and embedded tables and list formatting. This is often what one wants when pasting from sources such as Web pages.

The text pane now respects the checkbox " **check spelling at you type"** in the General pane of Document Settings. If a note is selected, and if the note uses the default value of $NoSpelling, changing this setting immediately updates spell checking for that note.

TIMELINES

Timelines are more liberal in drawing links; they draw links among all descendants of the timeline container. Previous versions drew links only among immediate children of the container.

Some note titles were incorrectly hidden in Timeline view.

MISCELLANY

Note ▸ Expand View was incorrectly disabled for views whose parent is a top-level note.

The key attribute value cell no longer attempts to reduce inter-character spacing when the text won't fit, working around an apparent bug in OS X 10.11.1.

On startup, defer hiding the text pane selector control until after the tabs have been initialized.

The GetInfo popover was not always dismissed by clicking in the view pane, if the view pane had the focus when the Get Info popover was opened.

Crash when pressing ⌘-F when the text field Find bar has the focus.

The Map Item contextual menu in Tinderbox provided "New Writing Space" command rather than "New Note". (1672)

Fixed a problem parsing tab-separated value data if a field contains an unmatched quotation mark.

Document Settings: Accounts now allows tabbing between account name and password.

Tinderbox is now more aggressive in performing rules and edicts after editing a key attribute of the current note.

Tinderbox updates the layout of map items after clicking on them, in case you've added or removed all the text from the note since this requires that the item be laid out.

Autocomplete in the Key Attributes picker is no longer confused by extraneous spaces.

Corrected an ugly hack in handling **expand Horizontally**, which should now give more predictable results.

View ▸ Treemap is now available from the text pane

When files from the user's directory are dropped into Tinderbox or into a file attribute, Tinderbox uses the tilde abbreviation to represent the path to the user's directory. This should make it easier to share one Tinderbox document across several machines, each of which share files in common locations (e.g. Dropbox) but which have different user names. (1474)

The initial focus of the Quickstamp pane (⌘-2) is now the search field.

Improved title and subtitle placement in arrow, hex, and lozenge shapes.

When multiple documents are open, their agents share a single queue to reduce contention amongst open documents.

Fixed a crash when pasting rtf data from the clipboard. (1688)

Autocomplete is better behaved for values like names which include more than one word.

Fixed a crash when opening a complex document because the agent thread tried to prematurely instantiate the window in order to mark it as modified.

The operation queue for HTML export rendering now uses the agent work queue, which should avoid crashes during complex and slow export operations in documents whose rules mandate frequent reindexing.

Fixed a crash after revert when reverting the only open document. The **setModified** action checks that there's a window to set as modified, but failed to account for the possibility that the window controller would be deleted before its asynchronous update.

File ▸ Update Agents Automatically and File ▸ Update Agents Now were disabled when a text window was the key window. (1712)

In a variety of situations, Tinderbox failed to set context variables such as **that, agent,** and **adornment**, which interfered with some complex expressions. Edicts failed to set **that** appropriately.

A subtle situation arose in documents with lots of complex rules, because the AgentManager and RuleManager sometimes changed the bindings for **that, agent, adornment** and and other keywords while we were in the middle of evaluating a rule or edict in the main thread in order to update a newly-selected note. This can be resolved either by evaluating the selected note's rule synchronously on the agent thread, which prevents agents from interfering, or by allowing the rule to make an immutable copy of the current action context at the start of evaluation.

The RuleManager and AgentManager have been rewritten extensively. This should increase stability, reduce the likelihood of incorrect results from actions, and also reduce the likelihood of crashing after Quit.

The keyword **agent** was not correctly resolved to the agent currently performing an action in situations where the argument could be either a list or a designator. For example, an attribute assignment $AgentPriority(agent)=-1 failed because an attribute assignment can take a list, while $AgentPriority($Path(agent)) succeeded because an attribute reference accepts a single designator. (1718)

Corrected crash when selecting a color scheme in Document Settings of recent backstage builds, due to overactive error handling. (1723)

Smart adornments failed to properly bind the designator **adornment** to the running adornment when evaluating their query.

After using the font panel to set a font such as $KeyAttributeFont from the Key Attributes table, Tinderbox failed to restore the firstResponder, which may be a new ValueCell that replaces the one we are editing. Doing so permits multiple changes to the key attributes font.

Nested designators did not always receive the correct evaluation context, and so dynamically-bound designators such is **that** or **agent** were not correctly evaluated in expressions like **sum(find(inside($Path(agent))), …)** (1728)

When an agent has a rule, the designator **agent** will be bound to the agent. Who an adornment has a rule, the designator **adornment** will be bound to the adornment. In both cases, one could simply use the designator **this**; the designators **agent** and **adornment** are only necessary in queries and actions where **this** is bound to the note being examined by the agent or the smart adornment. Still, binding the designators may make the intent of the rule clearer.

In map view, if a change to a key attribute caused a rule to move a note to move to a new container, the view was not properly updated.

Occasionally, **Close Document** saved a document with fewer open windows than it should have done. This may have occurred when an agent or rule marked the the document as changed, and needing to be autosaved, after the first window had been saved but before the last window had been closed. A new object, TbxDocumentUpdates, now monitors this situation and handles the asynchronous process of marshaling updates.

6.4.1

HIGHLIGHTS

GET INFO

Common Words, listed as "words" in the Get Info popover, now uses a new and better visualization method. Users can also choose a scope to examine: the selected note, the note and its descendants, all descendants of the note's container, or the entire document. The font used by Common Words is the NameFont of the selected note.

The Info pane of the Get Info popover now displays the note's position and dimensions.

In the Get Info popover, you may now tab from the pane list to the controls of the current pane.

Tab-key navigation in Agent pane of Get Info.

Tab-key navigation in Book pane of Get Info.

Tab-key navigation in Map pane of Get Info.

Tab-key navigation in URL pane of Get Info.

In the Attribute pane of Get Info, tab activates the search field rather than the category list.

IMPORT

HTML files, when dragged into Tinderbox, are formatted and imported as styled text.

Tinderbox now accepts .csv (comma-separated value) files as well as .tsv (tab-separated value) files. (1635)

MAPS

If Document Settings:Maps requests that Tinderbox should select the link destination after creating a link, then **Note ▸ Go Back** will now return the selection to the link source.

Expand Horizontal is less aggressive in expanding notes where $Height is so small that the title might be clipped even though it doesn't wrap.

Tinderbox attempted to draw names of adornments that were far too small to display them. Names are now hidden for adornments and notes if the available space is far from sufficient.

Note ▸ Expand View was incorrectly disabled for views whose parent is a top-level note.

OUTLINE

In outline view, the left-arrow key selects the parent of the selected note. If the outline view is focused on a container and that container is selected, the view is now expanded to focus on the parent container. If the outline view is expanded to include the entire document and the selected note is at the top level, left-arrow has no effect.

Outlines layout is now updated after ending an editing session by selecting a new item. Previously, the layout was only updated if the user pressed Return to end the editing session.

PRINT

Print should now respect paper choices in Page Setup.

TABS

If the Tab bar is hidden, moving the mouse near it will reveal the tab bar. Moving away hides the tab bar again, but only after a brief delay; this delay lets you use the breadcrumb bar if you wish. Previously, the immediate hiding of the tab bar meant that Tinderbox played “keep away,” snatching the breadcrumb bar away as soon as you approached.

TEXT

The Text Pane Selector allows you to view the note’s text, HTML, and to preview its exported HTML. Some Tinderbox projects don’t need to export HTML at all; **Window ▸ Hide Text Pane Selector** will hide this control. If the control is hidden, **Window ▸ Show Text Pane Selector** will restore it to its customary place.

The text pane can now be collapsed completely; previously, a very small residual width was always (barely) visible.

Selecting the text pane removes search highlighting. Previously, it was possibly to begin typing at the end of a highlighted passage and for the new text to be permanently highlighted.

TIMELINE

Timelines are more liberal in drawing links; they draw links among all descendants of the timeline container. Previous versions drew links only among immediate children of the container.

Some note titles were incorrectly hidden in Timeline view.

NEW ACTIONS

The new string operator **.find** returns the location of the first occurrence of a substring. If $MyString contains “We hold these truths to be self-evident”, $MyString.find("hold") will return **3**. If the string is not found, **.find** returns -1.

Intervals now provide **day, hour, minute,** and **second** operators for converting the interval to familiar units. For example, suppose $MyInterval is 12 hours. $MyInterval.day will be 0.5, $MyInterval.hour will be 12, and $MyInterval.minute will be 720.

Numeric computations now use greater accuracy, which should reduce rounding errors.(

MISCELLANEOUS

Outline export now respects $NameBold

Boolean attributes no longer interrupt the flow of keyboard editing of key attributes. To *change* the value of the selected checkbox from the keyboard, press the space bar.

While blind-typing to locate notes in the view pane, the delete key now deletes the most recently-typed character, not the selected note. The esc key cancels blind typing.

When first opening the document, if the text pane selector is initially hidden, hiding it is deferred slightly to allow you to see it sliding away, helping to confirm that it’s deliberately hidden (and suggesting that there’s a way to recover it!). This also avoids artifacts from trying to animate while the window is being assembled.

Sorting has been moved to the main thread, even when the underlying changes were made by the agent, rule, or edict thread.

Agent actions are now performed on the main thread; this should reduce or eliminate some intermittent crashes related to sorting.

The key attribute value cell no longer attempts to reduce inter-character spacing when the text won’t fit, working around an apparent bug in OS X 10.11.1.

On startup, defer hiding the text pane selector control until after the tabs have been initialized.

b181 could fail to open some files because Tinderbox tried to update text while the window was being laid out during tab initialization. The underlying problem was b181’s recalculation of layout geometry on selection, addressing issue 1603. This recalculation is now moved from all selection changes to the renamed: operator of the mapViewController.

6.4.0

HIGHLIGHTS

ATTRIBUTES

Attributes now offer Suggested values. Suggested values always appear in the value menu of key attributes, and are always offered for autocompletion, even if no notes currently use them. Suggested values may be added to attributes in the System Attribute and User Attribute inspectors.

A new read-only attribute **$NoteURL** exposes the tinderbox:// url used to access notes from other programs.

COMMON WORDS

The **Common Words** window returns in the form of a Get Info pane. The pane lists the most common words found in the entire document, omitting very common words like “the” and “and.” The size of each word in the image is proportional to its frequency.

EXPORT

Outline Export has been rewritten. Options are now available to export outlines as styled text (RTF or doc format) as well as plain text.

Text Export now allows subsection heads to be reduced in size for each level of indenting, providing finer control over export styling.

Pasting Tinderbox notes items into **DEVONthink Pro Office 2.8.8** or later provides DEVONthink with more information, including any $Tags assigned in Tinderbox as well as a link to the Tinderbox item.

FOOTNOTES

New Note menu commands allow you to add a **Footnote** either as a sibling of the current note or as a child of the Notes container for that note. The name of the newly-created note is taken from the selected word or phrase. A text link is created from the selected text to the footnote, and a basic link is created from the footnote to the currently-viewed note. After making a footnote, the selection shifts to the newly-created footnote; after writing the footnote, use the Navigate command to follow the basic link back to the note you were editing.

IMPORT

Selections copied from Delicious Monster’s library application, *Delicious Library*, are now pasted as reference item. Pertinent attributes are automatically filled.

Tinderbox accepts drags from the Calendar. If the document has an Event prototype, the newly-created note is assigned that prototype. $StartDate and $EndDate will be populated; the name of the note is the event title and the subtitle of the note is the event location.

LINKS

Double-click any link in the link browser to select its destination and dismiss the popover. If invoked from a view pane, the view will be scrolled or refocused if necessary to locate the newly-selected note.

New Note menu commands **Navigate** (⌘-Return) and **Go Back** (⌘-‘) are available. Navigate will follow the first basic link from the current note, selecting its destination. Go Back will return to the note most recently selected.

MAPS

A new option the **expand proportionately** has been added to Note menu, and the Map pane of document settings offers a new option to expand notes proportionately if the note name is long.

OUTLINES

In outlines, if a “leaf” note — a note with no children – is selected, collapsing that note (⌘⌥-left-arrow) will collapse its container and select the container. Formerly, “collapse” was disabled if the note had no children.

SHARING

Tinderbox now has a **Share** menu in the file menu. If some notes are selected, those notes will be shared with compatible programs. If no notes are selected, the Tinderbox document itself will be shared.

STAMPS

Stamps may be dragged in the Stamp inspector to reorder stamps in the stamp menu.

TEXT

Text: in the text pane, the **Indent** and **Unindent** commands (⌘-[and ⌘-]) indent and unindent the selected paragraphs.

NEW ACTIONS

Actions: new dot-operators **.count**, **.max** and **.min** return the size, maximum and minimum elements in a list or set. For example, $MyList.max returns the largest element in $MyList.

Actions: the new dot-operators **.beginsWith** and **.endsWith** return true if a string begins or ends with a specific substring. For example

```
$MyString.endsWith("cream")
```

is true if $MyString is “ice cream”. These operators search for literal strings, not regular expressions; if you need to search for a regular expression, use **.contains**.

Actions: the operators **linkToOriginal(), linkFromOriginal(), unlinkToOriginal(), and unlinkFromOriginal()** create and delete links. If either the source or destination of the action is an alias, the original note is used as the target rather than the alias.

MISCELLANEOUS

In **Outlines**, left-arrow now selects the parent of the selected note. Formerly, is selected the previous note in outline order, which up-arrow also does.

The Tinderbox inspector now reports the number of active agents along with the number of agents; this is useful if you have lots of inactive agents.

The font size of the note title displayed in the text pane is now reduced for long titles, allowing more of the title to be viewed.

AutoFetch was restricted to checking no more than once every five minutes. This makes debugging too hard. Now, the first update after opening a document will always perform an auto fetch.

The URL handler treated top-level items as an error. For example,

```
tinderbox://testing/?view=outline+select=1447138898;
```

should anchor the outline to the root, but instead failed to find a note named “” and posted an error.

Copy URL of a top-level note failed to export the / after host component.

```
tinderbox://testing/?view=outline+select=1447138898;
```

Prevent a crash when Tinderbox is reawakened after a flight in which the locale or time zone has changed.

Two new elements may be overridden in the optional configuration file, config/config.xml, located in the Support Folder. <TextFont> *font name* </TextFont> sets the default TextFont for newly-created documents, and <TextSize> *1-9* </TextSize> sets the relative text size from 1 (tiny) to 9 (huge).

Copy Note URL is now available on the Note menu as well as the item contextual menu.

Maps: stub links (links to and from notes in other containers) are longer and more easily visible.

File ▸ Update Agents Now will also perform any edicts.

Attribute inspectors: hide Suggested Values when type isn’t string, set, or list.

The Stamp Inspector disables its stamp name and action text fields when no stamp is selected.

Maps: the text thumbnail wrapped prematurely if the width of the map item was greater than the width of the note’s most recent text pane.

Avoided a startup crash when initializing agents, where changes were being processed on the wrong thread.

6.3.2

HIGHLIGHTS

ATTRIBUTES

A new built-in set attribute, $Tags, is available for free-form tagging of notes. The new Tags attribute is categorized under “References” because “General” is getting too full.

IMPORT

OPML import is now more tolerant of bad input, and no longer crashes if the file encoding is MacRoman.

Spreadsheet import again handles line breaks in quoted text fields correctly.

Spreadsheet import failed with certain complex, multiline text fields that contained encoded tabs or newlines inside escaped quotes.

INSPECTOR

Properties Inspector/Prototype pane: the menu failed to list all the prototypes if some prototypes share the same name, leading to erroneous selections when using the menu.

The fix of the Properties Inspector Prototype popup in b163 broke the map view prototype tab and the outline view prototype contextual menu. These work once more.

MAPS

Image adornments with transparent or translucent regions are no longer rendered opaque then the document is saved.

Arrowheads on linear links are again correctly rendered; their orientation had previously been arbitrary.

In maps, left-arrow and right-arrow now select the next and previous sibling of the selected note.

The plot inspector now allows you to enter a value for the target line.

XY Plots now draw a target line if a target is specified.

Changing some visible attributes of aliases, such as $NameStrike, failed to refresh the display immediately if the original note was absent from the current view.

ROADMAP

Roadmap is now available from the treemap contextual menu.

Torn-off Roadmap windows now automatically adjust the width of their columns, and also automatically elide titles if they exceed the window width.

TRAVEL

If Tinderbox remained open while the computer moved to a new locale, it would continue to use the old locale.

If Tinderbox remained open while the computer moved to a new time zone, it would continue to use the old time zone.

MISCELLANEOUS

Crash on wake, trying to update the layout of a ValueCellView in the key attributes table. Presumably, the selected note was an agent alias which was deleted while Tinderbox is in the background. Deleting changes the selection before the note is deleted, but that updated the text pane only if the text had been edited! Now, we update the text pane as well if the note has been hidden.

Crash when rendering preview containing ^similarTo^ in a document with active rules and agents, because the note was reindexed while rendering was under way. The Markup not makes a copy of the Lynx record.

Help: corrected export problems caused by missing double carets ^ in Additional Export Elements.

Took steps to prevent crashes on quit or when closing a document, if pending changes are enqueued when the document is closed.

In key attribute tables, the pull-down menu of value cells is now sorted case-insensitively.

It was possible to select a MapBackgroundColor document setting of “normal”, which (a) made no sense, since normal means “inherited or default” and we’re *setting* the default here, and (b) leaves us with the DebugColor, which is unsightly by design.

When edicts were updated while Tinderbox was in the background, the next edict update was scheduled for approximately five hours from the present, rather than one hour.

Fixed the help string for $Edition

Browse Links… is now available in the treemap contextual menu.

Text links created from an alias were not displayed, because the text is associated with the original note, not the alias. Now, creating a text link from an alias implicitly creates a link from the original rather than the alias, as the alias has no text to link to.

In some documents with active agents and rules, Tinderbox sometimes fails to save multiple window layouts whilst saving other information correctly. Modified autosave to ensure that documents are not marked as needing to be autosaved while in the process of being closed.

—

(For Developers) When one or more notes are copied to the pasteboard, Tinderbox now adds a new flavor **com.eastgate.tinderbox.metadata** for the convenience of other applications. The new flavor is a list of dictionaries, one for each selected note, which may have the following keys:

- Name: the note’s DisplayName (NSString)
- Tags: an array of NSStrings, possibly empty, containing the NSString values of each element of the note’s $Tags attribute
- Created: the note’s creation date (NSDate)
- Text: the note’s text (NSString)
- URL: the tinderbox:// url for the note

6.3.1

HIGHLIGHTS

Attributes: Intervals

A new attribute type, Interval, represents time intervals and durations. For example, the interval value

    01:25:40

represents a duration of one hour, twenty-five minutes, and forty seconds. The duration

    01:30

represents one minute and thirty seconds. The duration

    1h30

represents one hour and thirty minutes. A duration may be preceded by a number of days:

    1 day 01:00:00 (25 hours)

    2d2h30 (two days, two hours and 30 minutes)

An interval may have negative duration:

```
-5:30
```

Intervals may be added or subtracted from durations, multiplied or divided by constants or numeric attributes, and may be compared for equality using == and != or for magnitude using < and >. Interval may be added to dates. Note that subtracting two dates does not currently return an interval — rather, it returns the number of days between the two dates.

The interval “2d5” is now treated as 2 days, 5 hours. (1544)

The interval “1h30m10s” is now treated as one hour, thirty minutes and 10 seconds; previously the seconds field was ignored in this format. (1544)

The function interval(start,end) returns the interval between two dates. For example,

```
interval($Created,$Modified)
```

returns the interval between a note’s initial creation and its most recent modification.

The “unknown attribute” popover used when adding key attributes now lets you define interval attributes. (1542)

DEVONthink Integration

When any item is dragged from DEVONthink into Tinderbox, the newly-created note’s URL points to the DEVONthink database record for the original item. This has always been possible through the DEVONthink “Copy URL” command, but now it works seamlessly whenever you drag from DEVONthink. (Requires DEVONthink 2 or later)

Storyspace Support

Later this year, Eastgate expects to announce Storyspace 3, a new tool for writing hypertext narrative that builds on and extends the Storyspace legacy. Storyspace helped pioneer the craft of hypertext fiction, and Storyspace 3 will bring a wealth of new capabilities to this effort.

Tinderbox and Storyspace will share files, letting writers move seamlessly between them.

This release incorporates many changes in support of Storyspace, none of which should significantly affect Tinderbox. Several new System attributes have been added in the new Storyspace category, but these aren’t likely to provoke collisions.

Using Tinderbox From Other Applications

URLs that begin

```
tinderbox://
```

let other tools link to Tinderbox documents and specific tabs, views, and notes. The **tinderbox://** protocol has been substantially improved.

The protocol handler for tinderbox:// urls will now automatically open documents in the Recent Files list as well as documents that are already open.

The tinderbox:// protocol now accepts an additional command, select, which takes as an argument a semicolon-delimited list of IDs to be selected.

```
tinderbox://Notes/Issues?select=1429560859
```

Alternatively, the argument can be a URL-encoded name

```
tinderbox://Notes/Issues?select=Find%20Issues
```

If a single note is selected, Tinderbox will attempt to scroll it into view.

The contextual item menu for notes now contains an option to Copy Note URL which places the note’s tinderbox:// url on the clipboard. (1546)

Clarified fix 560, in which a contextual menu item places the URL of a document tab on the clipboard.

RevisedApplescriptManager to handle escaping more reliably, and corrected some mistakes in encoding tinderbox:// URLs.

ALSO OF NOTE

Actions

Using undefined colors — such assigning a note the $Color “Red” or “rde” when you meant “red” — previously had undefined behavior. We now return a specific arbitrary color (currently a brick red) for all undefined colors. (1538)

The dot operators .uppercase, .lowercase, and .captialize had too high an operator precedence, and so expressions like MyList.at(n).substr(0,3).lowercase required extra parenthesis to be parsed correctly. (1495)

Actions restoring the default or inherited value of a visible attribute

```
$Badge=;
```

incorrectly marked the attribute as changed even when the note’s value did not, in fact, change, potentially forcing necessary screen updates and interfering with scrolling.

Export

When exporting with $HTMLEntities false, some paragraphs ignored the note’s $HTMLEntities setting and export with entities anyway, causing difficulties when exporting East Asian languages (among others) to HTML. (1061, 1523, 1524)

New HTML markup elements ^firstSibling^ and ^lastSibling^ return the relative URL the the eldest and youngest sibling of this note, respectively.

Export as Text could display the export sheet in the inspector window if the inspector happened to have the keyboard focus. Tinderbox now disables Export As Text if the inspector has the focus. (1547)

Help

Corrected Help ▸ Feathering Your Nest ▸ Color Schemes

Many corrections to Tinderbox Help (thanks, Mark Anderson!)

Key Attributes

The formatting of dates in the Key Attribute table is now adjustable. By default, the date and time are shown using the system’s short formats for the current locale. Other formats may be chosen by changing the value of $KeyAttributeDateFormat. Suggested values include “L” and “l” to display only the date in long and short format respectively. (1421)

A new popup menu in the Text pane of Document settings lets you conveniently set the default format for dates in the Key Attribute Table. (1421)

In the key attribute table, the drop-down menu for set attributes now displays a checkmark next to items that are currently selected. (1519)

Typing in the key attributes picker prematurely ended an editing session if the text happened to include a text link or a smart link. Changing key attributes forced update rulers which toggled the text field, deselecting the token field and then reselecting it. (1525)

Maps

Tinderbox tabs should do a better job of keeping track of their scroll positions.

Avoid unsightly scrolling and unwanted boundary animation when dragging notes into a container. (1508)

Fixed several problems with the font button in Document Settings:Maps. (1539)

When right-clicking in the background of map views, Tinderbox displays the map coordinates of the click point as a disabled item in the menu.New Xcode beta.

Changing caption alignment in the caption pane of the Name inspector should now update the map view immediately. (1400)

Outlines

The “title bar” for agent icons is now heavier in outline view, allowing Tinderbox to better differentiate between agents that are running and those that are “off”. (1513)

Text

Ctrl-return now inserts a line break without also inserting a paragraph break. We avoid the conventional shift-return for this, on advice of the best authorities, because some typists press the shift key early in anticipation of capitalizing the first letter of a new paragraph. Ctrl-return should please Tinderbox users who work with poetry.

When editing the text pane, ⌘⌥-up-arrow and ⌘⌥-down arrow select the next and previous note, but now return the keyboard focus to the text pane. Previously, the view pane gained the focus. (1352)

Tinderbox no longer enforces $ParagraphSpacing if you override it while editing text. For example, if you are discussing a poem, you might want no additional spacing between lines of the poem, while still using $ParagraphSpacing to separate paragraphs. (1537 in part)

Automatic Link Detection (“Smart Links”), which makes URLs clickable (and blue) when typed in the text pane, is now disabled in template notes. (1489)

Forward-delete/fn-Delete when typed in the text pane was incorrectly intercepted and acted upon by the attribute browser view pane. (1536,1299)

Lengthy and complicated pastes were not inscribed in the document unless they were subsequently edited. If they were not inscribed, the text was also not marked as modified after pasting. (1548)

The text size of columns now adapts correctly to changes in the note’s font size and the view’s magnification. (1540)

Treemaps

Links may now be dragged from the parking spaces to notes in treemap view. (1514)

Workflow – Menus and Shortcuts

Added shortcut keys to item contextual menu’s entry for Roadmap ⇧⌥⌘-R (1516)

Document windows now remember whether or not the text ruler is visible. (1535)

The item contextual menu now includes Browse Links. (1549)

Browse Links is now disabled if the selected note has no links to browse.

MISCELLANEOUS

Edicts failed to run at hourly intervals, because the edict queue was suspended and not reactivated.

New (beta) version of development environment, with numerous changes to accommodate new SDKs.

Numerous corrections in Actions and Dashboards, and several fixes to the corresponding Tinderbox document.

Rewrote Network Updates::CheckForLatestVersion to allow a longer timeout and more reliable operations.

Fixed a crash when choosing a scheme from the color scheme picker. (1512)

To prevent accidents, Forward-delete is no longer active in attribute browser. (1522)

Explode now uses the prototype “/Prototypes/exploded notes” to contain exploded notes, creating a prototype if none exists. The default prototype simply adds $ChildCount as a key attribute. (1521)

Fixed an intermittent problem with layout in the Explode popover. (1537)

While closing a document, took steps to ensure that roadmap and get info windows do not attempt to use pointers to the now-deleted hypertext object. (1509)

Changes in network code to support El Capitan.

Addressed a possible crash when closing a window that is in the process of rendering an HTML preview. (1118)

Addressed a crash when creating Agents from the menu bar, arising because Tinderbox was updating the screen before the agent was properly baked

Tinderbox no longer performs Storyspace OnVisit actions when a note is selected. (1533)

Addressed a crash that could occur when the outline view pane was narrow — but not too narrow – because it proved impossible to measure the text rectangles.

Fixed a possible crash when editing Tinderbox Code Fields, when trying to find the names of attributes that begin with a given prefix.

6.3.0

HIGHLIGHTS

Treemaps

Treemaps are a new major view that reveal the structure of large and complex documents. Treemaps show the entire document, or any section of your document, as a set of nested boxes. The area of each box is typically proportional to the number of that note’s descendants, but may be proportional to any numerical attribute or expression. The color of each note may also reflect any attribute or expression.

Actions and Dashboards: A Walkthrough

An extensive new walkthrough, available on the Help menu, explore Tinderbox Actions and Dashboards.

Lookup tables

We sometimes need to some discrete values onto other values. For example, suppose we have some notes that record the state in which each customer resides. We want to know the sales region for the customer; each state is assigned to a region. One approach would be a set of if statements:

```
if($State=="AL") { $Region = "South";}
```

This can be tedious. Instead, we create a Set, List, or String of keyword/value pairs, separated by colons:

```
$RegionList="AL:South;AK:NorthWest;..."
```

We can now write

```
$Region=$RegionList.at($State);
```

If the set does not contain the key value, .at() looks for the special key default and returns its value; if there is no default value, .at() returns the empty string.

Lookup tables might be inherited from prototypes or kept in confederation notes:

```
$Region=$RegionList(/config/geography).at($State);
```

Lookup tables can specify several keywords that share a common value by separating the keyword with the pipe character |. For example, "Oliver|Micawber|Pip:Dickens;Palliser|Finn:Trollope". Lookup tables have been added to Help.

Actions

When expressions are coerced to numbers, the logical value **true** is now coerced to 1, and **false** is coerced to zero.

The currency format converters .format("$") and .format("$0") now use locales set by the locale() operator if it has been called; otherwise, they use the system's default locale.

Explode

The explode action is now applied *after* the text of the newly-created note is set, allowing the action to modify or depend on the exploded text. (1422)

Explode now remembers the most recently-used delimiter, which can be convenient when using complicated regular expression delimiters. (1423)

Export

Print has been added to the contextual menu for the HTML Preview pane. (1418)

Maps

Responsiveness of large and complex maps has been greatly improved. When dragging notes, Tinderbox is more aggressive in reducing detail in order stay responsive. In particular, links may be suppressed during the drag if link drawing is impairing responsiveness. (1128)

Progress Bars now accept an optional fourth argument, a target value.

bar(value,*min,max,target*)

The target represents a nominal or desired result. For example, we might write from 0 to 4000 words on any given day, but during NaNoWriMo we want to set the target at 1500 words/day.

The target is drawn in a dashed link alternating $PlotColor and either $Color or $Color2. (1468)

Fixed a crash when displaying a summary table if $Name contained an encoding error. (1478)

Outlines

Outlines update elements more smoothly while dragging the pane splitter; element resizing was being animated, which is unhelpful. (1451)

Right-clicking the background of an outline view (e.g. below the final item or in the left margin) now allows you to Create Separator in place of the disabled option to Create Adornment. (1385)

Tab bar

The tab bar may now be hidden (View ▸ Tabs ▸ Show Tabs). The hidden tab bar is "spring loaded": when the mouse moves near the top of the window the tab bar will be revealed and remain until the mouse moves away. (345)

The selected tab of most view types now has a small "info" button that permits you to set options for the tab; this replaces a variety of buttons and contextual menus formerly used to set background colors and other options.

Text

Format ▸ Style ▸ Reset Margins resets paragraphs in the selected range to use the standard margins and line spacing. (1419)

Editing a note with text links could previously have led the text links to shift position in the text. (1415)

When the key focus is on the text pane, the menu titles for Show Key Attributes/Hide Key Attributes/No Key Attributes are updated appropriated. (1425)

We now have a disclosure button for the key attributes table, as well as the show/hide menu command.

Text no longer scrolls to the top after paste. (1424)

In the text pane, Magnify and Shrink now change the size of the selected text, rather than scaling the view pane.

Automatic detection of URLs in note text is now enabled by default. It can be disabled using Edit ▸ Substitutions ▸ Smart Links. (1436) The status of automatic URL detection is now saved with the document and will be restored when the document is reopened.

Text: after pasting more than a few thousand words into a note, Tinderbox might fail to save the revised text unless it was subsequently edited because a formatting operation timed out. The formatting operation is faster now and the timeout more forgiving. (1438)

$SmartQuotes now controls both automatic quote substitution and automatic dash substitution. Double-hyphens in templates will no longer be converted to m-dashes, a convenience that plagued HTML comments. (1462)

To edit the anchor of a text link, option click in the link to set the insertion point without activating the link. Made a note of this in the Help file. (1396)

Miscellaneous

Using the search field's autocomplete menu in the Attributes pane of the Get Info popover incorrectly dismissed the popover, because the popover was transient and ought to be semi-transient. (1426)

If the parent node of a hoisted outline was selected and the user pressed return, the newly-created note was outside the scope of the view and therefore invisible. Instead, we now create a child of the hoisted parent, rather than a sibling. (1398,1413)

Map View: changes in 6.2 for more efficient map views rendered the map commands to Move Up/Down and Move To Front/Send to Back erratic. (1434)

Windows refused to close if a user badge was installed with a name that conflicts with a built-in badge. Tinderbox had code to resolve the conflict, but that code was faulty. (1435)

In the key attribute picker, slightly expanded the token field typing area to reduce the need to scroll. (1444)

The circular Timeline Info button has been replaced by an info widget in view tab. (1443)

The text label width of zero-duration events has been increased 60% to reduce unwanted wrapping of titles.

Select All is available in timeline (1441), chart view (1440), and treemap (1439).

When opening multiple windows, we did not always instantiate the windowController's window for secondary windows when reopening the file. Subsequent saving would eliminate the window. (1463)

Newly-created adornments were drawn with the wrong z-axis ordering, so they appeared beneath adornments that Tinderbox believed them to be in front of. (1458)

Adapted version and build strings to conform better to conventional usage.

Added documentation for local variables and the var statement. (1461)

The **Create Link** button was positioned incorrectly in its popover. (1466)

OutboundWebLinks export markup again works correctly. (1465)

Resizing notes and adornments sometimes failed to update the position of their text fields and shadows. (1464)

Cleaned up some obsolete constructors and factories in Node and NodeMaker.

Making a numerical attribute sequential now properly initializes the values of existing notes. (1363)

Edit ▸ Select All (and a number of other useful commands) were unavailable in treemap view because the treemap mouseDown handler didn't set the responder correctly. (1477)

View ▸ Treemap was disabled in Attribute Browser. (1476)

Tinderbox should remember scroll positions in maps and outlines more accurately.

Corrected a console warning arising when errors are discovered during a periodic scan from a background thread, causing Tinderbox to try to update a the layout of the error table in the wrong thread.

6.2.1

HIGHLIGHTS

Pie Plots

Pie charts have been improved in a variety of ways. (1353)

- Labels are now drawn atop pie chart segments; previously, some segments could be drawn atop labels from other segments.
- The colors for each segment of the pie chart are taken from the attribute $PlotColorList. The first color designates the color of the first segment, and remaining colors in the list are used in rotation until all segments have been drawn. If the list contains fewer than two colors, "black;white" are used inside.
- If the container is sufficiently wide, a legend is drawn to the right of the pie chart. Otherwise, pie segments are labeled in the pie chart.
- If $Direction is false, the first segment begins at the top of the pie and subsequent segments are added clockwise – as is customary in geography. If $Direction is false, the first segment begins as the right edge of the circle and segments are added counter-clockwise, as customary in mathematics.
- Pie segments are now separated by a dark gray line.
- If a container or agent has a pie chart, then the alias of that container or agent will also display the same pie chart, if space allows.

Renaming Notes

In-place editing (outline): clicking in the edit field incorrectly closes the edit session. This was a byproduct of fixing issue 1370; we now address that issue differently. (1414)

Actions

When formatting numbers format("$") and .format("$") apply conventional formatting for your local currency – for example, "$1,063.52"

When formatting numbers format("$0") and .format("$0") apply conventional formatting for your local currency, rounding to the nearest currency unit – for example, "$1064" .

Equality comparison == and != now work with Color attributes. Colors are equal if they designate the same color — for example, the colors "#FF0000" and "bright red" are equal in the default color scheme. (1408)

Equality comparison == and != now work with URL attributes. Note that URL comparisons are currently case-sensitive, even in the host: `http://Eastgate.com/` is not equal to `http://eastgate.com/` . This may change in the future. (1409)

Miscellaneous

Fixed an auto layout error in the text pane's title field, which could grow unreasonably if a note's title contains many paragraphs.

Moved the "smart quotes" checkbox in Document Settings: Text to the correct location.

The Rule and Edict panes of the inspector now indicate whether the rule or edict is inherited, or immediate. (1358)

When importing large spreadsheet tables, we no longer set the text of the import container to contain the full table, as this can take a long time without much benefit. (1217)

Corrected a crash when switching to a new space (?or perhaps awaking from sleep) while the Get Info popover is displayed, because on returning to view Tinderbox tries to refer to the no-longer-extant model for the Get Info category list.

Added some precautions to guard against Tinderbox agent updates attempting to proceed after the document has been closed.

Shortened the timeout on Simplenote authorization requests from 30 to 15sec.

In outline view, Note ▸ Create Note created new notes at the top level of the view, even if the selected note was more deeply nested. Note ▸ Create Child Note has also been corrected.

Console messages were displayed after undoing note creation in outline view, warning that Tinderbox could not create a LayoutInfo. We no longer ask it to create the LayoutInfo for the now-hidden note. (1411)

$HTMLExportPath had incorrect values for notes not at the top level. (1412)

After dismissing the find bar, Tinderbox returns the focus to the currently selected note. (1404)

Searching in the view pane's Find bar now temporarily highlights text in the text pane. Highly experimental. (1406 in part)

Fixed a crash when trying to import text files encoded with UTF16 rather than UTF8. (1416)

CeresReader could sometimes leave notes read from a document marked as "newlyCreated", and this could sometimes lead Tinderbox to unexpected behaviors such as spontaneously opening Get Info when switching tabs, if the selection in the destination tab happened to be an agent.

6.2.0

HIGHLIGHTS

- Import and Export are simpler, yet also provide more flexibility. A variety of new Export tools let you export your work to familiar formats instantly, while the flexible template-based approach remains for demanding export needs.
- Map performance has been substantially improved.
- A new feature, **Edicts**, allows you to establish low-priority rules for automatic housekeeping.
- Badges can be larger, and can be emoji or other characters.
- Text can contain lists and tables.

This release is recommended for all Tinderbox Six users.

Actions

The designator **child[-1]** designates a note's youngest child. **child[-2]** designates the elder sibling of the youngest child. (1292)

runCommand passed MacRoman to the command line arguments, not utf8 (1301)

In agents, **that** is now bound to the examined note both when the query is evaluated and also when the action is performed. Previously, **that** was not bound for the action.

A new attribute $ImageCount in the Textual category reflects the number of images in the note's text.

Date formats: new format code U formats the data in Unix style (seconds since January 1, 1970) (1319)

Explode: the explode popover adds an Action field which lets you enter an action to be applied to each exploded note.

Added a new attribute group, Sandbox, to hold several attributes useful for testing and experimentation: **MyString, MyNumber, MyList, MySet, MyBoolean, MyColor,** and **MyDate.**(1331)

The attribute **$Bend** replaces the former attribute $LeafBend. The former attribute continues to work as before but the old name is deprecated. (1355)

Convenience

In outlines and other views, ctrl-Return creates a new note as the elder sibling of the selected note.

Tinderbox now recognizes additional abbreviations of days of the week. For example, in English the four-letter abbreviation "Thur" is now recognized for "Thursday." (1238)

During long exports, both ⌘-period and Esc cancel the operation. MapItemController has been intercepting the cancellation. (1326,1327)

The tooltip for the text pane note title field is the full path to the note. If the note is an alias, the tool tip shows both the path to the alias and the path to the original note. (1350)

Better behavior for page-up/down (fn-up-arrow and fn-down-arrow) in outlines. (1306)

Support for home/end keys (⇧-fn-up-arrow and ⇧-fn-down-arrow) in outlines.

Edicts

A new action attribute, Edict, lets you specify housekeeping actions that do not need to be updated frequently. Edicts are performed soon after a document is opened, and then are updated at intervals of approximately one hour. Edicts are especially useful for chores like archiving obsolete notes which only need to be performed occasionally. (1214)

A new intrinsic Boolean attribute, EdictDisabled, allows you to disable Edicts associated with a note. Often, EdictDisabled will be used when a prototype has an edict to be inherited by its instances, but you don't want the edict to apply to the prototype itself.

A new pane in the Action Inspector permits you to view, edit, and enable edicts.

The Agents and Rules pane of the Tinderbox Inspector now reports the edict count and the interval since the most recent edict update. (1393)

Export

Export has been extensively revised to provide simpler access to common export formats.

Export Templates: a folder "templates" is now created in the Application Support folder to hold export templates that are to be shared among files. Shared templates are stored as text files in this folder, and may be added to any document by selecting the file from File ▸ Built-In Templates. If a template with that name already exists, the template's text is replaced with the text of the template file. (1300)

Extensive changes to export, which may now be a bit faster. Export is now performed in the background; in principle, you can continue working during export.

Backed out changes to make export more massively parallel, because the export context (including binding of this) is shared by the hypertext and so isn''t thread safe. Added an issue to explore this.

Export now uses a progress indicator built into the main window. Enjoy this; it was a bear.

Export commands moved to a submenu of the file menu.

Export ▸ As OPML exports the entire document as an OPML file, adding the built-in OPML templates if necessary.

Export ▸ As Outline exports the entire document to a text file, presenting the display name of each note in a tab-indented outline.

Export ▸ As Rich Text exports the entire document as an rtf file, including titles (treated as headings) and the text of each exported note (exported as

styled text).

Export ▸ As Rich Text (no titles) omits note titles.

Export ▸ As .doc File exports the entire document in Microsoft Word format, including (treated as headings) and the text of each exported note (exported as styled text).

HTML Export can be cancelled (at last!) by pressing ⌘-period or Esc while the keyboard focus is in the map pane.

Export Selected Note has been rewritten to use the standard save dialog.

A new export method Export ▸ Attribute Browser exports each attribute browser category and the notes it contains.

$HTMLFileNameLowerCase now defaults to false; in previous Tinderbox Six releases it defaulted to true.

Export commands for .doc export are now enabled from the text pane as well as the view pane. (1318)

RTF/Word Export: changed titles from 24pt to 18pt.

Added an option sheet for export, permitting us to consolidate RTF, doc, and text export into a single menu choice.

The new option sheet for Export now allows you to choose to export the entire document or only the selected notes. (1317)

Export: OPML export is now an option in the Text export sheet, rather than its own menu choice.

Export: added an option to export plain text.

Export: added an option to export to Scrivener.

Export: title size is adjustable.

The standard Save panel displayed by the Export Inspector's Export To button can now create folders.

The label of the Export To button is now correctly placed.

Fixed links in Tinderbox Help:Export

HTML Export: conditional evaluation of the existence of notes, e.g. `^if(nextSibling)^ .... ^endIf^` always returned `true`; it now returns `false` if the designated note doesn't exist.

HTML Export: conditional evaluation of the existence of notes will now work with designators. `^if(nextSibling)^` will be `true` if this note has a `nextSibling`, `false` otherwise.

Single note export now sets the initial export location to the note's expected export folder. Formerly, the export location was set only of the file already existed.(1325)

Smart Dashes and Smart Quotes are both implicitly disabled (again) in templates.

Import

Tinderbox now imports .webarchive files. On import, the $URL of the imported note is set to the URL of the archive's main resource, and Tinderbox takes a rough stab at extracting the styled text to the imported note. (1307)

XML-parseable text files, such as many html page templates, failed to import because they are mistaken for malformed OPML. Now, files with text or txt extensions will never be checked for parseability. (1303)

AutoFetch now imports styled text extracted from the target URL, rather than importing raw HTML.

If Tinderbox is asked to parse an invalid OPML file, it leaves an error notice in the text of the imported note. (1302)

.md, .mmd, and .markdown files are now imported as text.

Tinderbox now accepts file drops of .tsv files (tab-separated values) (1332)

When Tinderbox supports tab-separated tables (including drags from spreadsheets), cell values that are surrounded by straight quotes have the quotes trimmed. (1332)

Inspectors

Quickstamp: the search field now offers autocompletion (1377) and updates the current value on selection when a completion is chosen (1243).

Applying a Quickstamp immediately updates the key attributes table, allowing faster feedback. (1254)

In the User Attributes pane of the Document Inspector, if you try to name or rename an attribute with a name beginning with '$', the '$' is ignored. Previously, this was treated as an error. (1298)

The Tinderbox Inspector's Agents and Rules pane now indicates that automatic agent updates have been disabled.(1321)

The System Attribute pane of the Document Inspector now indicates when the selected attribute is intrinsic. (1357)

Name and Caption inspector: alignment terms are no longer capitalized. (1373)

The missing "font" type now appears in its place in the User attribute type list. It is disabled, as only system attributes can usefully hold fonts. (1369)

Closing a document no longer closes the Inspector if some other document remains open. (1367)

The Text Inspector's Text panel now correctly reflects the note's paragraph spacing; formerly, it was off by one. (1371)

Maps

Substantial improvements in scrolling speed, both for mouseWheel scrolling and grabby hand scrolling. Links in complex maps are automatically hidden during scrolling to improve performance.

The tendency of the map view to "jump" after creating new notes in sparsely-populated views has been reduced.

Breadcrumb bar appearance and removal is animated, breadcrumb titles are positioned more neatly.

Link labels were truncated at scales after increasing the magnification of the map, because the cached link widths were note cleared.

Labels for curved links were placed as if the links were linear; the new label placements should be somewhat better.

Agents now have a summary table widget at all times, even if they have no children. Formerly, agents with no children did not display the table widget. (1256)

Emoji and other unicode characters may now be used as badges. In place of the badge name, set $Badge to the unicode character to be displayed in the badge area. Best results will generally be obtained with $BadgeSize of 32 or greater.

Shrink To Fit was far too conservative in using the available width and height of the note. (1330)

If a smart adornment has a subtitle, the layout algorithm now allows space so the subtitle will not be completely obscured.

Better selection of link pads when note edges nearly overlaps, especially when large notes are closely spaced.

Somewhat better map printing.

Smart adornments allow a little extra space beneath the title so that notes do not crowd title descenders.

Maps: when clicking in the interior of containers, Tinderbox failed to make proper allowance for the Title bar, which displaced the click target for the note from the note's image. (927)

Outlines

In outlines, ⇧-PageUp (⇧-fn-upArrow) moves to the first item and ⇧-PageDown (⇧-fn-downArrow) moves to the last item.

Double-clicking in the background beneath an outline adds a new note as the youngest child of the parent, not the eldest.

Improved drawing of agent icon in outlines. (1375)

Text Pane

Added Table… and List… commands to the Format ▸ Text menu, providing panels to insert tables and lists in the text pane.(1379)

Using Check Spelling While Typing in the text pane should now update both $NoSpelling and the current text behavior immediately. (1281)

Tinderbox text panes now save and respect line spacing, paragraph spacing, list styles and tabs set in the ruler. (1245, 1271)

When $ParagraphSpacing is used, half of the spacing is now applied before the paragraph and half is applied after the paragraph. Previously, the entire spacing was applied after the paragraph. (1251)

Text: Tinderbox 5 documents opened with a fixed right margin, which is unhelpful in Tinderbox 6.

The text pane's contextual menu displayed inappropriate link-editing options when right-clicking on a text link. These options no longer appear, and are replaced by Browse Links. (1362)

Text again observes $LineSpacing and $ParagraphSpacing throughout the text, overriding local changes from the ruler.

Scrolling the text pane and then activating it with a click no longer scrolls the text pane to the start. (This was a side-effect of the ruler update workaround.)

Miscellaneous

(Internal) DirectoryList, which is used by the badge picker, built-in templates and built-in prototypes, now automatically skips all files with the .xml extension. The special file manifest.xml is read, parsed, and may be used to provide configuration information about the directory.

(Internal) The optional manifest.xml file may be added to user badge directories as well as built-in badge directories. It has the form:

```
<badges>
 <monochromatic> 1 </monochromatic>
 <oversize> 0 </oversize>
</badges>
```

If monochromatic is not zero, choosing a badge from this family will set $BadgeMonochrome to true and the badge will be drawn differently against a dark background color. If oversize is not zero, $BadgeSize will be set to its value.

HTML Pane: If a note has no template, and the HTML pane is visible but the divider is dragged to hide it, the divider width formerly popped back to reveal the pulldown menu that allows the user to select a template. The layout had been coaxed to permit the pane to remain completely hidden. (1257)

In some cases, deleting all the text of a note, deselecting, and then reselecting the note left the text pane typing attributes set for Helvetica 12. The typing attributes in this case are now restored to the note's default text font. (1304)

A specific note – originally imported from email – crashed when attempting to render it into HTML. (1308)

Console messages "CoreAnimation: warning, deleted thread with uncommitted CATransaction…" were traced to attempting to mark the document as edited from a secondary threat. TbxDocument's setModified now uses the main thread.

TbxDocument's setModified: no longer launches an asynchronous task if the task won't change the then-current value of isDocumentEdited.

When importing an OPML file that isn't valid OPML, Tinderbox now identifies the line where the parser failed in the text of the imported note.

Avoid crashing when a selecting a malformed color scheme in Document Settings:Colors.

Rewrote tbxMapValet's restoreItemOrder for more efficiency, allowing us to maintain the z-ordering of items even during scrolling.

When stamps are applied to a large number of notes, a progress indicator appears.

Views: when editing a view title, pressing ESC cancels the editing session. If the note was previously untitled, it is deleted; otherwise, the note's title returns to its previous title.

Get Info should again be available in Mavericks when an item is selected and the item has the focus. (1329)

Corrected right margin handling when saving.

Browse Links: when examining links from an alias, Browse Links now list text links from the original note, since aliases share that original's text and text links. (1328)

In Document Settings ▸ Maps, the Interior Scale slider displays its current value, and updates that value as it is moved. (738)

The subtle gradients of the bezel border are suppressed when printing and copying images, since Quartz renders translucent gradients incorrectly when drawing offscreen.

Format ▸ Style ▸ Black (^⌘-5) now restores the color to the note's text color if the text color is not black.

Outline text heights were measured incorrectly for notes where $OutlineTextSize was not 100% (1294)

Clarified what "Link to Selected Text" does by setting its title to reflect the selected text. If the text selection is "Arkansas" and the note Arkansas exists, the menu title becomes Link to "Arkansas".

Legacy documents with a border bevel of "normal" are now treated as having a border bevel of "automatic" in the Appearance inspector.

Links are refreshed after drag-scrolling a map view, since they may have been suppressed during drag-scrolling for better performance.

Fixed a distracting animation artifact in maps when snapping a note to a guide changed both the mapView bounds and the position of the dragged note.

New compiler and SDK release, requiring various updates to remove deprecated idioms.

Corrected layout of Get Info URL pane. (1374)

If, after editing the name of a note in outline, you click into the text pane to close the edit session and begin revising the text, the text pane's title field

was not updated to reflect the changed name. (1370)

Roadmap: shortcut changed to ⌘⌥⇧-R

Adjusted background color appearance in roadmap tables.

Corrected a problem when creating macros, in which the newly-named macro would not be selected for editing (1372)

Menu item: Window ▸ Toggle Full Screen (1364)

Agent with a cleanup action of “none” or “” retain their map layout. Previously, the layout lost when the file was reloaded. (1378)

Revise mouseUp in expand widget to ensure that the scrollable viewport grows after the expand. We failed to cancel the potential drag before updating the view frame, causing scrolling to sometimes appear “sticky”. (1381)

Browse links: links to aliases are now italicized. (1382)

Badge picker: Extra “blank” items arising from system files hidden from the finder are no longer displayed. (1383)

Edit ▸ Make Alias creates an alias as the sibling of the selected note. Previously, the alias was created as a child of the parent note of the view. (1389)

The HintView’s illustrative tab arrow has been moved leftward to avoid hitting the splitter bar. (1392)

Explode has been revised to reliably delete delimiters. (1394)

Browse Links handles multiline text link anchors more gracefully. (1397)

6.1.3

HIGHLIGHTS

This release is recommended for all Tinderbox Six users.

What’s New?

A new item in the TinderboxSix menu displays highlights of this version.

The What’s New window will also be displayed at startup for registered users the first time they install a new version.

Getting Started With Tinderbox

A tutorial walkthrough for beginning users, found in the Help menu.

Badges

The badge picker has been completely rewritten to support larger badge families.

New attribute $BadgeMonochrome. In maps, if $BadgeMonochrome is true and if $Color is dark, the badge will be drawn in sourceOut mode instead of sourceOver. This is useful when using black badges on a dark map background. (1226)

$BadgeSize is a numeric attribute which sets the size of the badge in map view. If zero (the default), the default badge size is used. The Badge Picker automatically sets $BadgeSize to 32 when selecting badges from the Avatar family; an even-larger badge may sometimes be desirable.

The Badge Picker’s pane selector adapts somewhat more reasonably if there are numerous user families of badges (1268)

Text

Highlighting selected text always used yellow, even if the red or blue highlighter was selected. (1216)

Dog ears sometimes appeared in notes and containers which had formerly held text, but from which all the text had been deleted.

Changing dates in the key attribute table would often set the selection to the first row, not the next row, because the need to update the value cell forced the table to be rebuilt, losing the selection. (1218)

Paste styled text into an empty note, then delete the text. The typing style correctly adopts the style of the pasted text. However, in past versions, Format ▸ Style ▸ Standard Font did not correctly reset the typing style to the standard font if the note was empty. It does, now. (1205)

Documents can be Saved and Closed from the File menu when the current window is a text window. (1232)

Timeline

Timeline links are drawn once again. (1173,1206)

The minimum width of timeline items has been reduced, improving the appearance of short-duration events with short display names.

Miscellaneous

Text windows are now saved with the document and reopened when the document is reopened. (1212)

A new designator construction, child[n], allows you to designate the nth child of a note. Adornments are ignored in choosing the available children. The eldest child is designated as child[0] or, as before, simply child. If a note has no children or if the nth child does not exist, the result is noSuchNote and the value returned will be the default value or be undefined. (1227)

The label for case-sensitivity in the agent inspector now reflects the actual functionality. (1207)

Thanks to a glitch in the constructor for HTMLExportTemplate, Tinderbox might think that a link context (set up by link-listing constructs such as ^basicLinks^) existed when none does, resulting in a crash. (1223)

Addressed a crash when scrolling the results table in a torn-off Find window.

Fixed a crash in Attribute Browser, which could in some circumstances be forced to sort simultaneously in two different threads.(1222)

Smart adornments always issues multiple change messages, even if they did not, in fact, make changes.

Stamp inspector: "+" should save pending changes (1221)

When adding a built-in template to a document which has no default template, the newly-added page template becomes the document’s default template. (1213)

Fixed a crash when closing text windows. (1230)

Automatically draw icons from the Icons and Symbols family in light gray against a dark map color, and dark gray against a light map element.

Rewrote the Badge Picker because (a) the old picker used the newly-deprecated NSMatrix, and (b) the old picker couldn’t accommodate the large repertoire of the new Symbols family. The change also permits us to update the badge immediately, rather than when the popover is dismissed, and permits arrow key navigation in the Badge Picker. (6, 991)

Document Settings: the Simplenote account and password fields could appear not to be saved if new information was typed and then the window was closed without selecting a different text field or pressing return. (1235)

Typically, HTMLExportExtension is either empty or begins with a period (for example, ".html"). If the period is omitted, Tinderbox will now supply it. (1231)

Sequential attributes now auto-increment correctly. (1219)

Attribute Browser: if multiple tabs held an attribute browser, changing the attribute category in one tab also changed the category in other tabs. Attribute browser tabs are now fully independent. (1200)

Link Types Inspector: corrected handling of the color well.

Renamed tab contextual menu item New Window to reflect what it actually does. (1098,1186)

In the action inspector's query pane, we now provide prompt feedback for the number of notes the agent has retrieved. (1167)

Fixed a crash when pressing when the cursor focus is on a string or set-valued key attribute, but the attribute's text field is not active. (The pulldown menu was returning an unexpected null value because no choice was made). (1237)

If HTPath was called for a synthetic note or one that doesn't exist, it could crash because it used an obsolete test (node==nil) and was not prepared for a null object (NotANode*). (1264)

Updated the OPML and Scrivener templates to normalize handling of $HTMLExportExtension.

Help for Get Info and Inspector.

In a specific document, creating a new agent by right-clicking the background and selecting Create Agent… intermittently did nothing or created an agent in the wrong container. The problem arose because the parent note was taken from the current selection; if that selection was an agent alias, then no agent could be created. The parent note is not chosen appropriately. (1269)

Similarly, creating notes by pasting text into map view could create in an unexpected place if the selected note was not in the map view.

Removed a half-dozen additional Law of Demeter violations that involve node->GetHypertext()->; these can run into trouble if called on a NotANode.

Corrected the (Node*) constructors of some classes like AttributeInfo which inherit both from HypertextFacade and NodeFacade. (1270)

Additional help, especially for export.

6.1.1

HIGHLIGHTS

This release includes fixes intended to make Tinderbox more reliable, correcting a number of recently-reported issues.

This release is recommended for all Tinderbox Six users.

Actions and Rules

Action values have not been thread safe; if one thread changes an action while another thread is using the same action, Tinderbox might crash. To avoid this, we add a recycling pool for actions and only delete actions in the agent thread. This should now avoid risk of the agent thread using invalidated, deleted actions. (1114)

A document with many inherited rules could crash on update, because the new, thread-safe approach to attribute fetching failed to handle cached actions correctly. To avoid this, we no longer cache compiled rules; this wastes some work, but the since rules are no longer evaluated on the main thread the performance hit may be tolerable. (1141)

Fixed a crash in a rule when applying nsort to a list of items that are not, in fact, numeric.(1180)

nsort() sorted incorrectly if the list contained floating point numbers. (1188)

Modified format() and .format() so the localized formats "l" and "L" respect local handling for separators. For example, in the US the result of 10500.format("l") will be "10,500"

Agents

Turning an agent on or off in the Action Inspector now updates the agent's outline icon immediately to reflect whether it is running or not. Previously, the icon might not be redrawn until the selection changed. (1127)

Attribute Browser

Attribute Browser: category labels are formatted more legibly. For example: "801-890" rather than "8.1e+2-8.9e+2" (1168)

Fixed a crash when switching to attribute browser from other views, if the browser attempted to set up categories before the browser attribute control had been initialized. (1172)

Removed animation when switching between attribute browser and another view, as the crossfade was unsightly with outlines and had terrible performance. (1199)

Dates

Days of week: the date parser now recognizes the day of the week in the current locale, not only in English. Recognized forms include the full day (Sunday), the short day (Sun), and the very short day (S). Note that the very short day is ambiguous in many languages, including English.

Month names: the date parser's super-lenient formulation, which recognizes partial dates like "Jan 15, "now deals with month names and month abbreviations in the current locale.

The super-lenient date parser now recognizes dates like "Jun 2014" as well as "Jun 5"

Unparseable dates are now treated as "never". (1137)

Export

File>Export Note asks for export folder, defaulting to the current export folder. (1110)

Export Selected Note is now enabled when the text pane has the focus; it was previously available only when the view pane had the focus. (1175)

Links

Attempting to make a link from the text link parking space by clicking and typing the name of the destination crashed .(1143)

Revised link parking space help messages, and hide link parking space help controls when they are inapplicable. (1144)

Maps

When pasting notes in maps, if the map background has been clicked, the pasted note is placed at the location of the most recent click.

When pasting a group of notes into a map, the relative geometry of the pasted notes is preserved. (983,1076)

Added "Paste" to map background contextual menu. (796,1077)

Menu command for Paste is no longer enabled when nothing can be pasted.

Images on the clipboard may now be pasted into maps as image adornments. Image files may also be dragged. Images on the clipboard may also be pasted into outlines or charts; in this case, a new note named “pasted image” is created and the image is placed in its text.

Double-click on the note name no longer hoists the view in maps, outlines, or charts. Instead, the title is opened for editing. Triple-click opens the title for editing and selects the entire title. Double-clicking the interior continues to hoist the view in maps, and double-clicking the icon hoists the view in outlines. (1148)

Summary tables no longer coerce integers to show two decimal points. This old behavior might be desirable for currency, but decimal points can still be forced using .format(2).

Adornments inside agents could not be pasted or duplicated. (1197)

Text excerpts in maps now respect $MapBodyTextSize. (1190)

Summary tables incorrectly handled tables in which an interior cell is empty or contains only white space. For example, if a line was

> **Lincoln | | 1865**

then “1865” would be drawn in the second column, not the third. (1198)

Menus

The item contextual menu now provides an opportunity to open a Text Window. (1132)

The Find results table now has a contextual menu, and offers an opportunity to open a Text Window (1132) in both popover and freestanding window implementations. This is particularly tricky because the menu is owned by the table, not the cell view, and so must be prepared through the NSMenuDelegate protocol.

Outlines

Outlines: link widgets are larger (1140).

Selected notes are no longer expanded when switching tabs.(1162)

Roadmap

Roadmap: Improved layout of torn-off roadmap, so title remains centered and lists expand to fill the available space. (1158)

Roadmap no longer lists links to notes deleted in this session. These links are retained internally while the deletion might be undone, but should not be displayed. (1170)

After scrolling a long list of link sources or link destinations in roadmap, some notes which have no badge were listed with another note’s badge. (1195)

Simplenote

Simplenote syncing had been inadvertently disabled. It’s back now. (1171)

Text and Key Attributes

Key attribute tables and Get Info>Attributes now accept Format>Text>InsertDate and Format>Text>Insert Date and Time with equivalent effects. (1142)

File attributes in key attribute table: if no value is specified, clicking on the folder icon lets you choose a file. Previously, the folder icon was disabled. (1129)

Format>Style>Standard Font now adjusts the typing attributes. This accommodates the scenario where the user pastes unwanted text in a new style, deletes the text, and then applies the standard font: because there is now selection, previously this failed to reset the typing style which the unwanted paste had altered. (1125)

If you try to add a key attribute that is not defined as an attribute, a popover allows you to create a new attribute on the fly.

Key Attributes: the popover for unknown key attributes now deletes attributes that don’t exist if those attributes are not created (1151).

Key attributes: the unknown attribute popover now includes "file" and "boolean" types (1152).

Key attributes: autocomplete interference with names that such as DestState, which have prefixes that are shared by only one existing attribute, should no longer prose difficulties (1154)

Text windows: when a text window is activated, its cursor position remains where it was when the window was deactivated. If the text has been shortened in another view, so that the cursor position no longer exists, the cursor is placed at the start of the text. Formerly, the cursor was always placed at the start of the text. (1133)

Read-only attributes no longer enable “Use Inherited Value” in the key attribute table. (1177)

Selecting a new note and clicking in the text pane could sometimes cause the text to scroll to an arbitrary position. (1201)

The “unknown attributes” popover now supports creating URL attributes. (1202)

Miscellaneous

The positioning of the “+” control in the tab bar has been corrected.

The Color Palettes picker now includes a palette that reflects the current Tinderbox palette.

The Registration Window provides a link from the reminder text to the order page (1155)

The initial position of the inspector has been moved, allowing more space in the upper right-hand corner for notifications. (1165)

When a new document is opened, its first window becomes the key window. (1161)

$ID was not correctly populated for newly-created notes. (1174)

Yosemite triggers layout debugging pane in its standard file windows; disable the debugging pane in release builds. (1173)

Popovers again detach correctly under Yosemite. (1178)

Crashes when closing popovers that ought to be capable of being torn off under Yosemite should no longer occur. (1178)

The Stamp Inspector failed to reset its text fields after switching documents if the nth stamp was previously selected and the new document had at least n stamps. (1182)

Switching tabs to a tab using the Preview pane, and then switching back to any map pane, resulted in a host of update problems. If the preview completion block failed to run, then change updates would be permanently locked. Even if the preview completion block did run successfully, the update was not thread safe.

Fixed a typo in Help:Anatomy of a Note

6.1.0

SUMMARY

This release includes fixes intended to make Tinderbox more reliable, correcting a number of recently-reported issues. Most significantly, Tinderbox now saves the state of all windows that were open when the document was closed.

In addition, this release prepares for the OS X 10.9.5 update and for the forthcoming OS X 10.10 Yosemite.

Tinderbox Help has been expanded substantially.

This release is recommended for all Tinderbox Six users.

Aliases

Traditionally, Tinderbox draws alias titles with an italic font. This is problematic for note titles written in languages that do not typically use italics. A new document setting (Outline pane) allows you to underline alias titles in order to more clearly identify them. (1062)

Editing an alias now marks the original as modified. (1056)

Attribute Browser

Attribute Browser: crash when switching to User attribute family when there are no user attributes. (1071)

Attribute Browser: addressed an intermittent crash when sorting the attribute browser in the background.

Attribute Browser: the disclosure triangle animation no longer stalls. (1044)

The attribute browser now reflects $NameStrike and the title's strike-through style.(1104)

Badges

The Badge Picker now sets the badge of all selected notes, not only the note whose badge was clicked. (1047)

Built-in Templates and Prototypes

When creating built-in templates, the Templates container's OnAdd action now sets $IsTemplate=true. (1064)

Built-in Prototypes: added an OnAdd action to the prototype for HTML Template. (1068)

Built-In Prototypes were not created if the document had another note with the intended name, even in a different container. We now create a new prototype as expected in the Prototypes container.

Charts

Expand All is enabled in Chart View. (1055)

Export

Renamed File>Export to HTML… to File>Export Document (1101).

Added a new menu item File>Export Selected Note(s) that exports only the selected notes. (1101)

If a note is marked as a template, automatic substitution of dashes and automatic text replacement are disabled. (1031)

Get Info

Get Info>Agents: editing a query or action without pressing return, and then changing the priority, discarded changes to the action or query. (1046)

Get Info: changing colors is now undoable (1086)

Corrected scrolling of long values in value cells of Get Info's Attribute pane. (1091)

Get Info adjusts the popover width to suit the selected pane, but failed to adapt the width of the initially-selected pane because the pane was selected before the popover had been created. (1096)

The Get Info popover is not transient rather than semi-transient, and is more readily dismissed. (1120)

Inspector

In the Text Inspector, fields for Display Expression and Hover Expression now offer autocompletion.

Maps

Potentially improve performance when dragging in a map view if an attribute browser is open in another window. (1032)

Maps: the link widget is larger, and its click target is larger yet, to make it more easily discoverable and less finicky to use. (1082)

Maps: after pasting in a map, scroll to the pasted note.(1075)

Maps: shift-option-drag in the background should marquee **extend** the current selection rather than replacing it. (1079)

Shift-click and cmd-click on the map background no longer deselect the selected notes. Simply clicking the map background clears the selection; the modifier keys suggest that the user wants to do something else — and perhaps simply clicked the background by accident. (1078)

In map view, ⇧-click now extends the selection, exactly like ⌘-click. In outline and chart views, ⇧-click continues to range-select from the current selection to the clicked note. (1080)

When a map or outline has the focus, it displays a subtle shadow along its left edge. Some people dislike the shadow; it is now controlled by the boolean attribute $MapBackgroundShadow. (1081,1084)

Restored the crosshair cursor when pressing the option key for marquee selection. (1083)

Outlines

In outlines, clicking on the collapse/expand widget of a note no longer selects that note. This makes it easier to manipulate an outline while editing a note's text. (956)

Expand widget: ⌘-click expands the clicked note and its younger siblings. Formerly, only younger siblings were expanded. (1069)

Expand widget: option-click expands the clicked note and its descendants. (1069)

Revised Create Separator. When no note is selected, the separator is created as the parent's youngest child. (1085)

Preferences

Tinderbox Preferences is now available even when no document is open. It is also no longer possible to open two instances of the Tinderbox Preferences window. (1073)

Registration and Updates

The Registration pane now shows both the version you're using and the latest version, and provides a link for downloading the latest version. (1063)

Text

Format>Style>Standard Size no longer removes bold and italic styles. (946)

Format>Style>Standard Font now sets the font of the current selection to $TextFont and the size throughout the current selection to the default size. Other styles will remain unchanged, provided that they can be rendered in $TextFont. (29, 39, 837)

Text windows now display Key Attributes. (1072)

Key attributes: after editing a value and pressing , the focus moves to the next row. (1074)

From the text pane, ⌘⌥-up-arrow and ⌘⌥-down-arrow move the selection in the view pane up or down, equivalent to up-arrow and down-arrow in the view pane. (1000)

View>Magnify, View>Shrink, and View>Standard Scale are now enabled when the text pane has the focus (1093)

The Text Window command will now open a text window for all selected notes, unless more than nine notes are selected. (1094)

Text acquired a fixed trailing margin, which was frequently undesirable. (1121)

Timeline

Added "Change Settings…" to the timeline background's contextual menu. (1053)

Windows

Tinderbox now saves the tab state of each window separately, and restores the tab state when loading the window.

Miscellaneous

Tinderbox no longer asks whether you want to save empty documents.

Abbreviated dates of the form 06/2014 are recognized and interpreted as the first of the month. Note that 6/14 is interpreted as June 14 of the current year, not as June 2014. (1099)

In Get Info attribute browser and in key attribute tables, autocomplete is no longer limited to choosing among the first 99 values. (1102)

Fixed a possible crash when setting a badge.

The bottom of the label "Description" in the Document inspector User Attributes pane was truncated. Correcting this led to a cascade of layout issues in the document inspector.

Revised the XML format to prepare for saving and restoring the state of multiple windows.

Fixed an intermittent crash when opening a text window, especially when the text window was empty, arising from activating a window without properly setting up the saved insertion point. Added insertion-point caching for text windows, so the selection is preserved when the text window is reactivated. (1065)

Fixed a potential crash when restoring the selection of a text window if the text were edited in another window while the text window was deactivated.

The RuleManager now checks $RuleDisabled before running a note's rule. (1059)

Revised handling of setViewController patching of the responder chain to accommodate both traditional and Yosemite behaviors.

Simplified drawing of tabs.(1024)

Disable assertions in release builds.

Break potential dependency cycles in many popover classes where the class was the popover's owner *and* its delegate.

Document inspector: the height of the text field for default values was insufficient.

Fixed a potential crash in the parking space help dialog if you try to make a link when the parking space is empty.

Explode omitted the final character(s) of the final exploded note, if **delete delimiter** was checked and if the final note did not end with a delimiter. (1087)

Stamps were sometimes disabled when more than one note was selected because the text pane inappropriately disabled the stamp. (1092)

Many menu items were incorrectly disabled in attribute browser.(1095)

Many additions and improvements in Tinderbox Help.

Built with Xcode 6.

6.0.4

Tinderbox 6.0.3 sometimes failed to load specific files because it tried to scroll to the text's insertion point when the text had no insertion point. This release is otherwise identical to 6.0.3.

6.0.3

SUMMARY

This release includes fixes intended to make Tinderbox more reliable, chiefly newly-reported issues. Most significantly, a new sort option has been added to the Attribute Browser control panel, allowing you to sort notes within attribute browser categories. In addition, this release prepares for the forthcoming OS X 10.9.5 update.

This release is recommended for all Tinderbox Six users.

Attribute Browser

Add controls for sorting to attribute browser. (1042)

Previous Tab and Next Tab are recognized in Attribute Browser. (1016)

Refactored Attribute Browser and Quickstamp to extract a TinderboxAttributePicker to coordinate category and attribute name popups. Repaired broken test if the picker is not initialized because the inspector controller doesn't call awakeFromNib.

Map View

Identical note titles did not always appear identically in maps on non-retina displays, perhaps because MapItemView frames were not required to fall on integer boundaries. SRect::AsCGRect and SRect::AsNSRect now return integral rects, which should improve matters. (1013)

bar() and vbar() patterns are now drawn for containers and agents. (972)

One can no longer paste notes inside agents. (987)

In map view, link stubs, which indicate links to or from notes not present in this level of the map, were sometimes omitted. (1006)

We now permit maps to zoom in (Focus View) on notes without children; this can be useful if you want to begin adding children to a newly-made container. Confusion that formerly arose should be lessened by the prominent breadcrumb bar. (1003)

In maps, up-arrow/move up now selects the former parent after moving up a level in the map hierarchy.

Smart adornments could crash on open with recursive calls to MoveOff because they used integral rectangles to compute overlap (see b87).

Adornment actions are now longer applied if the note was already on the adornment before the drag was started. (964)

Clicking on a locked adornment now selected the adornment. No adornment handles appear, as a locked adornment cannot be resized, but the lock icon is now shown in order to permit the user to unlock the adornment. (1043)

Outline View

The parent of a hoisted outline is not available as a link target; previously, attempts to link to the parent node were rejected. (975)

In outlines, items with large font sizes received far too much leading. (1014)

Extra vertical spacing frequently was applied in outlines with columns, because the effective text rectangle was 32px narrower than the actual text rectangle. (1018)

Text Editing

Changing text color or highlighting failed to take effect if no text was selected, because Tinderbox failed to update NSTextView's typingAttributes. (1010)

Strike through is now enabled when the selection is empty.

Activating a text window moved the scroll position to the top while leaving the insertion point unchanged. Instead, the insertion point is now moved into view. (1039)

HTML Export

Increased the limit for repeated ^ includes from 10 to 90.

HTML export: /help/release_notes/6_0_3.html crashed. (960)

Menus and Shortcuts

File>New Window doesn't need ellipsis. (967)

In the tab contextual menu Rename "New Window" as “Move Tab To New Window” (984)

Get Info… is available in the attribute browser. (1011)

Edit Background… is now enabled in Chart as well as Map and Outline view. (1004)

The shortcut keys for Expand and Collapse are now ⌘⌥-left-arrow and ⌘⌥-right-arrow. The option key is added to avoid a conflict with conventional start-of-line and end-of-line key bindings. (999)

View menu acquires a Tab submenu with New Tab, Next, and Previous. Next and Previous Tab commands have shortcut keys ⌘⌥-[and ⌘⌥-]. (994)

Other changes

Date strings prior to November 1883 were sometimes converted to a day one day later than intended. This may continue, depending on your local time of day, pending discovery of the proper way to handle time zones before the introduction of time zones (of which, it seems, time zone objects are aware).

Fixed a potential crash when Tinderbox switches to the foreground or background whilst updating the document inspector.

In the Agent pane of Get Info, the right-most part of the priority pulldown control could not be clicked, because is was covered by the transparent portion of the adjacent text field. Adjusted constraints to avoid this. (963)

In Document Settings, the Simplenote password is now a secure text field. (980)

The default value for the Creator preference in newly-created documents is now the current OS X account name. (981)

$ReferenceURL is now a URL attribute, not a string. (990)

Certain malformed regular expressions could cause a crash when used in agent queries, actions, or attribute browser queries (1015)

Built-in Prototypes: changed the hover expression to use $Date.format() in place of format($Date,…)

Build and sign on Mavericks in preparation for OSX 10.9.5.

Double-clicking a note in the Similar Notes list of Get Info will select that note. (1044)

Explode: using $$$$ or **** for a delimiter yields regex matches of length zero, which hangs the chunkier. (1049)

Explode: title is first sentence doesn't stop at first sentence but continues through the whole note, except that the first note is titled correctly. (1050)

Timeline adornments no longer enforce a minimum width. (1035)

6.0.2

SUMMARY

This release includes fixes intended to make Tinderbox more reliable, chiefly newly-reported issues that had not been seen backstage. This release is recommended for all Tinderbox Six users.

Text editing

Added a new checkbox **Smart Quotes** to the Text pane of Document Settings. $SmartQuotes is now a preference attribute using this setting as the document default. (898)

Text highlighting is again available (Format>Style>Highlight menu) (928)

Cleaner handling of ⌘-left arrow and ⌘-right arrow, which are start and end-of-line in text pane but collapse/expand in right pane. If the text pane is active and the current selection can collapse or expand, it will be collapsed and expanded. If the current selection cannot collapse or expand or is already collapse or expanded, the arrow keys have their natural meaning. (938)

Map View

Select All followed by Expand Vertically failed to expand some notes that had never been displayed in the view, because their bounds rectangle was not set. We now set the bounds rectangle as needed. (908)

Clicking the lock of a locked adornment unlocks it; previously, the locked adornment couldn't be clicked and so clicking the lock had no effect. (921)

Outline View

Indent and outdent notes in outlines was slow when ~500 notes or more were selected. Speeded this up by a about two orders of magnitude by avoiding unnecessary screen updates and skipping an O(n^2) collision-avoidance algorithm that didn't work particularly well in any case. (932)

Notes that scroll into the bottom of the outline after an expand widget is collapsed were not being drawn because they were failing to receive updateValues. (947)

Get Info

The search field in Get Info:Attributes was not functioning properly because a the search field in the nib was not connected properly. (913)

The search field in Get Info will again select the intended attribute when typed or selected. (926)

Inspector

Fixed problems in inspector panes caused by misinterpreting the null hypertext. (923)

More inspector fixes. (923)

A crash could occur when setting the sort order of a smart adornment in the Action inspector, if the action inspector's **selectedCell** method returns nil or a null object. (939)

A crash could occur when opening documents if the inspector is hidden but the current inspector pane is the document inspector. (941)

Explode

Explode's **delete delimiter** option now deletes the delimiter. (924)

Other changes

New icon (thanks Robert Black, strategy cartoonist)

Text links for aliases include *both* the text links for the original *and* any text links that have been added to the alias. Note that creating a text link in the text of an alias does not add the link to the original note, even though both alias and original share the same text. (905)

For some reason, a note named "à" crashed Tinderbox because the name string couldn't be turned into an NSString. This is bizarre, but a guard has been added to catch it. (924) Perplexing problems with notes beginning with, or sometimes containing, à and £ turned out to be an error in Parser::Trim, which was not properly unicode-aware in searching for non-breaking spaces and which, in consequence, improperly trimmed notes that began or ended with certain Unicode code points. (936)

Some legacy files grew to enormous size in Tinderbox Six, and increased in size whenever the file was saved. This problem turns out to have been caused by erroneous handling of file paths. Vestiges of MacRoman support led Tinderbox to specious encoding errors in some cases; these typically had no effect but in some cases could return incorrect and very long paths. (934)

Notes were sometimes marked as modified in the course of reading them from the document file, if the note had text or styled text. (940)

Corrected a layout error in chart view (943)

Corrected failure to create an agent if no note was currently selected. (952)

6.0.1

SUMMARY

This release includes numerous fixes intended to make Tinderbox more reliable. This release is recommended for all Tinderbox Six users.

VIEWS

Added "Badge…" to the item contextual menu. (897)

Roadmap is now available when right-clicking an item in maps or other views. (863)

The tooltip for table cells in roadmap and link browser is the path of the destination node for outbound links, the path of the source node for inbound links, and the url for web links. (895)

The outline icon now indicates the presence of links. If the note has inboundhelp links, a small arrow is drawn to the left of the note. If the note has outbound links, a small arrow is drawn to the right of the note. The result may be too busy, or may prove a performance bottleneck. Comments welcome. (862)

Default and fallback fonts in several contexts changed from Lucida Grande to Helvetica Neue. (881)

Notes created by pressing [Return] in map view with a note currently selected are now placed next to, and vertically aligned with, the selected note. (793)

In maps, links outside the selected neighborhood are now more opaque and typically darker. (861)

In outline view, if no note is selected, down-arrow and up-arrow both select the first note in the view. (855)

When holding down the option key in map view or chart view, the cursor changes to the cross cursor when not hovering over an item to indicate that marquee selection is active. (811)

Separators with children no longer appear in map view (834), and children of separators no longer appear in chart view (835).

ATTRIBUTE LISTS AND KEY ATTRIBUTES

File attributes: the contextual menu for the folder now includes "Show In Finder." (58)

File attributes now recognize "~" (tilde) as a shorthand for the user's home folder. (866)

The File button in the key attributes table now has a contextual menu allowing you to open the file (if one has been chosen) or to select a file to open (832).

The Get Info attributes browser now has URL and file buttons.

MENUS

New menu item Stamps>Quickstamp with shortcut ⌘-2. (882)

Several menu commands are newly accessible when the text pane has the focus. These include View>Timeline, View>Roadmap,View>Expand, View>Collapse, View>ExpandAll, View>CollapseAll, View>New Tab, View>Use Columns, View>Use Checkboxes, View>Hide Key Attributes. (886,887)

Browse Links now sorts text links by their position in the text, making it easier to locate a specific link in a long list. (888)

Rearranged View menu to group views together and separate popovers. Roadmap gains a shortcut: ^⌥⌘-R. (850)

Format>Text>Show Ruler now changes to Hide Ruler when the ruler is shown. (838)

Edit>Document Settings now has a shortcut key ⌘-8 (836)

File>Update Now renamed Update Agents Now.

Reorganized File menu to place prototypes and templates together.

TEXT

Text formatting respects $Tabs. (841)

The trailing (typically right-hand) margin of the text is automatically set to match the available space, correcting difficulties that arose when pasting text with an explicit right margin. (883)

Option-click in a text link to set the insertion point without following the link. (885)

For greater clarity, View>Hide Key Attributes now toggles between “Hide Key Attributes” and “Show Key Attributes”.

ACTIONS

The **runCommand()** command has been extended to accept an optional third argument specifying the intended working directory.

runCommand(command,input,directory)

creates a shell and runs the designated command. The input, if any, is passed to the command’s standard input. If *directory* is specified, it sets the working directory. Otherwise, the working directory is the user’s home folder ~. (831)

EXPORT

HTML Preview: Edit>Select All (⌘-A) works as expected, as do other familiar editing commands. Copy should reliably copy the previewed HTML. (845,846)

Export Inspector: the Export Folder button now selects the current export folder; previously its initial selection was unspecified.(848)

DETAILS

The limit of 200 notes/band is raised to 500 notes, for the benefit of Kandahar.

The value of $ID of some aliases was set to zero; it now correctly reflects the internal ID. (899)

Caption placeholder text was transiently visible. (900)

Inspectors now update values when switching to a different hypertext, even if you were editing a value in the previous hypertext.

Fixed a possible cause for reported conflicts with dictation software. (902)

If a note had a rule forcing a screen refresh, the edited text was reset when the rule was run. (893)

Occasional problems relating to PColor’s conversion of NSColor to the correct color space may not be understood, but should be resolved.

Fixed a crash when TextPaneListener called DocumentHasChanged from its work queue rather than the main thread. (874)

Fixed a crash in Exploder when omitting text from exploded notes, if there was in fact no text to omit. (878)

After importing OPML, notes might initially expand in outlines without their proper names, instead displaying “Label”. (877)

Resolved the “sticky map anomaly.” The crucial unspoken factor here was that the sticky map anomaly only occurs when the view pane is inactive, typically because the text pane is active. (876)

Corrected the Haight theme to use the intended linen background.

Corrected the Standard theme to use no background fill.

Tinderbox Six has been counting aliases of agents as agents in the Tinderbox Inspector’s count of agents. Aliases will no longer be included in this count.

Mysterious deletions: untitled notes created in map view were deleted when switching to outline view and then back to map view, even if those notes had descendants that should have prevented the automatic deletion. (890)

Link parking space autocompletion lists no longer include aliases. Aliases are less common link targets, and their presence makes it hard to know which of several identically-named notes is the intended link destination. (884)

Email feedback on help now identifies the source page correctly.

Avoided a hang when opening documents, caused by obsolete thread synchronization.

Cleaned up tracking of the item being dragged, improving performance slightly when dragging in map view and *possibly* addressing reports of intermittent failure to respond to map view clicks.

We now read OPML files and accept OPML file drops on the dock. OPML files open as Untitled. (204, 795)

Adornment actions were not triggered when notes were dragged onto adornments (869)

Documents that make heavy use of computed values such as $DisplayName, especially for sorting, ran into a variety of hangs and crashes tied ultimately to the caching mechanism. The fetcher returns a pointer to a Value object; for a computed object, the fetcher caches this object and deletes it on the next fetch so it doesn’t leak. In the past, the sort comparator cloned the first Value to compare it to the second Value, since the second fetch deletes the cached value. But now, with multiple threads, we need to clone *both* values right away, lest another thread call the fetcher again and delete the second value. (858, 871, 872)

Throttled agents significantly to reduce computational load in typical documents. Please advise if your agents are updating too slowly.

Double-clicking on a locked adornment created *two* notes with identical positions. (870)

Leading and trailing spaces are now ignored in registration code processing. (860)

Select All (⌘-A) now selects the HTML Preview text when the HTML Preview pane is visible.

Fixed a hang, evident in aTbRef, caused by rebuilding the index simultaneously inside and outside the indexing queue.

Fixed exorbitant CPU use caused by launching thousands of tiny asynchronous tasks to deal with the Value recycling queue. The underlying problem isn’t that values need to be recycled, but rather that some computer attributes weren’t properly synchronizing in their ValueFor methods, making it possible for two threads to simultaneously dispose of the same value.

In outlines, Create Agent was wrongly disabled when the selected note is an agent. (859)

Remove some obsolete outlet references in nibs; these cause problems in Yosemite.

Rename the instance variable “shadow” to “shadowHelper”, as this causes a name collision in Yosemite (and perhaps elsewhere).

Modified map dragging to reduce jitters when resizing the boundaries of the map.

When creating a new note in the outline by pressing [Return], the view scrolls if necessary to ensure the note is visible.

Possible fix for the sticky map view problem (which has not yet been reproduced) (852)

$HTMLExportTemplate is no longer a preference attribute and no longer a file attribute; instead, it's a string attribute.

Agents can no longer be created inside agents. (808)

In map view, the view bounds is now updated correctly after a moving multiple notes. (827)

Tinderbox would crash if, while editing an outline column, you clicked the ⊗ button to close columns. (843)

Revised indexing again in order to avoid occasional crashes, and to simplify the code by replacing locks with a serial queue.

Modified sorting so all sorting is done on the same queue – currently the main queue. Occasionally, the user interface and the agent thread would try to sort the same container simultaneously, with bad results.

In the key attributes table and in Get Info:Attributes, a pulldown menu is available when editing strings, lists and sets. Making a choice from this menu now cancels the current editing session; previously, the current editing session overrode any choice from the menu. (849)

We now include the Tinderbox 5 Help manual as an aid to new Tinderbox users who otherwise would lack documentation.

Pasting certain phrases in Chinese into a map or outline caused a crash because the title-choosing method returned invalid UTF8. (822)

Improved typing performance by avoiding sort updates in response to changed in text. This should also alleviate a crash arising from conflict between updating sort order on both the agent and the main thread.

Resolved a very tricky crash in a document where an agent changed the $Container of reference notes. Dragging a Bookends reference into the document correctly created a reference note and its item view; the agent would then run but the reference view was not correctly removed from the map. It turned out that the newly-created note was still marked as being dragged, so it was never recycled.

While editing note title in place, double-clicks failed to select the clicked-on word. (830)

Some backstage team members were inadvertently omitted from the About box.

Switching from outline view to another view, and then back to outline, turned off Show Columns even if Show Columns had been on. (839)

Adding a built-in prototype relied on the container's OnAdd action to set $IsPrototype. We now explicitly set IsPrototype in case the user omitted or changed the action. (842)

If $OutlineBackgroundColor is set, separators could overpaint part of the border that lies beneath the text control. (824)

Expand widgets for separators which have children now observe the correct left margin. (829)

Corrected Chart view layout problems; the layout policy was using the wrong font size. (728,791,840)

In outline columns, requesting 2 decimal places always gives two decimal places, even if the value is an integer. Previously, we did not display trailing zeroes. (825)

$Text.replace("\n",";") did not work as expected because Tinderbox retained some obsolete code for dealing with old-style Macintosh line endings in text.

Clarified the registration status message when the upgrade period will soon expire.

Added some material to Help.

Links to and from aliases were sometimes not created when the document was loaded. Alias resolution is asynchronous, and in some situations the alias resolution was not yet complete when links were being loaded. We now ensure that all aliases have been resolved first.

Fixed a hang in attribute browser when view numeric attributes and choosing reverse sort order. (800)

Pasting a multiple selection, all of which have prototypes, sometimes failed to set some prototypes because the paste triggered an asynchronous index rebuild that interfered with locating the prototype. Instead, we now rebuild synchronously (804)

Tab drawing failed cryptically when trying to truncate long tab labels that which (a) don't fit the allotted space, (b) contain multi-byte characters, and (c) have the misfortune, when the multi-byte sequence is improperly truncated, to yield invalid UTF8. (806)

Simplenote synchronization crashed in a complex document (syncing 600 notes, 50,000 words) when seeking the second authorization token, because the second pass deleted the existing importer and created a new importer while tasks from the first importer were still pending. (814)

Backdrop watermark should now be updated as soon as the registration code is entered. (810)

If a map has more than fifty links, drag-scrolling the map results in consistently misplaced links. Complex maps suppress animation, which forces periodic redraws; absent those redraws, we need to ensure a redraw after adding a new item view as it scrolls onto the screen. (816)

If a file has an XML error and cannot be parsed, Tinderbox reports the unparseable line to assist trouble-shooting. (802)

Clicking on the autocomplete menu of the Get Info action pane no longer dismisses the popover (818)

Plots could crash if Tinderbox attempted to plot zero or 1 points. (819)

TextPaneListener could crash, though very infrequently, trying to copy the text pane's textStorage when the main queue was redrawing the pane. (820)

Various corrections and improvements to Help.

License, Copyright and Colophon

This help document was created and edited with Tinderbox. Other especially helpful tools included BBedit, Acorn, Skitch, Xcode, AppCode, DropDMG, and Transmit.

Tinderbox™ Software License Agreement

Eastgate Systems, Inc., grants you a non-exclusive license to use this copy of the program on the following terms:

YOU MAY:

I) Use the program on any one computer;

II) Install the program on a second computer you use provided all copies of the program are used directly and exclusively by yourself.

II) allow anyone else to use the program, so long as there is never more than one user per licensed program at any time;

III) make copies of the program in machine-readable form, but only for archival purposes, and only so long as all proprietary notices are reproduced on each copy.

YOU MAY NOT:

I) Modify, translate, reverse engineer, decompile, disassemble, create derivative works based upon, or copy (save for archival purposes) the program or the accompanying documentation;

II) rent, transfer or grant any rights in the program or accompanying documentation in any form to anyone else without the prior written consent of Eastgate Systems, Inc.;

III) remove any proprietary notices, labels, or marks on the program and accompanying documentation;

IV) use this program, or permit this program to be used, on more than one computer at any one time.

Non-compliance with any of the above restrictions will terminate this license.

This license is not a sale. Title and copyrights to the program and accompanying documentation and any copy remain with Eastgate Systems, Inc.

Household License:

You and up to four (4) other persons who occupy the same household may use the program on their respective computers. "Household" means a person or persons who share the same home, apartment, condominium or dormitory suite, but shall also extend to student household members who are primary residents of the household but who reside at a separate on-campus location.

Limited Warranty and Disclaimer

This product and associated files are provided without warranty of any kind, express or implied, including without limitation the warranties that it is free of defects, virus free, able to operate on an uninterrupted basis, merchantable, fit for a particular purpose, or non-infringing. The entire risk as to the quality and performance of the product is borne by licensee. Should the product prove defective in any respect, licensee and not licensor assumes the entire cost of any service and repair.

This disclaimer of warranty constitutes an essential part of this agreement. No use of the product is authorized except under this disclaimer.

The liability of Eastgate Systems, Inc., shall be limited to the replacement of the product or the refund of the purchase price. This is the entire liability of Eastgate Systems, Inc., and your exclusive remedy. Save for the above express limited warranty, Eastgate Systems, Inc., makes no warranties or conditions express, implied, statutory or in any communication with you.

This agreement is the entire agreement. If any provision of this agreement is held invalid, the remainder of this agreement shall continue in full force and effect.

Eastgate Systems Inc.
134 Main Street
WatertownMA 02472 USA

Tel: (800) 562-1638 (617) 924-9044

Email: info@eastgate.com

Web: http://www.eastgate.com/