<p>| Hierarchy | 24 |
| Links | 24 |
| Attributes | 24 |
| To change an attribute’s value | 25 |
| To change an attribute’s default value | 26 |
| To create a new attribute | 26 |
| Attribute types | 26 |
| String | 26 |
| color | 26 |
| File | 27 |
| Boolean | 27 |
| Date | 27 |
| Interval | 28 |
| Number | 28 |
| List | 29 |
| Set | 29 |
| URL | 29 |
| Action (System use only) | 29 |
| Attributes and Agents | 30 |
| Exporting Attributes | 30 |
| Suggested Values | 30 |
| Key Attributes | 30 |
| Key Attribute display differences | 31 |
| Key Attributes | 32 |
| Displaying Key Attributes | 32 |
| Key Attribute value pop-up lists | 33 |
| Key Attribute value autocomplete | 33 |
| User Attributes | 33 |
| Aliases | 34 |
| Aliases and The Original Note | 35 |
| Intrinsic Attributes | 35 |
| Making Notes | 35 |</p>
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart Adornments</td>
<td>52</td>
</tr>
<tr>
<td>Badges</td>
<td>52</td>
</tr>
<tr>
<td>User badges</td>
<td>53</td>
</tr>
<tr>
<td>Cleanup</td>
<td>53</td>
</tr>
<tr>
<td>Containers</td>
<td>53</td>
</tr>
<tr>
<td>Display Expression</td>
<td>55</td>
</tr>
<tr>
<td>Hover Expression</td>
<td>55</td>
</tr>
<tr>
<td>Summary Tables</td>
<td>56</td>
</tr>
<tr>
<td>Attributes and the Summary Table</td>
<td>58</td>
</tr>
<tr>
<td>Histograms</td>
<td>58</td>
</tr>
<tr>
<td>Progress Bars</td>
<td>60</td>
</tr>
<tr>
<td><strong>Using Outlines</strong></td>
<td>60</td>
</tr>
<tr>
<td>The disclosure triangle</td>
<td>60</td>
</tr>
<tr>
<td>Editing a note title</td>
<td>61</td>
</tr>
<tr>
<td>Separators</td>
<td>62</td>
</tr>
<tr>
<td>The Outline Icon</td>
<td>62</td>
</tr>
<tr>
<td>The Outline Icon</td>
<td>62</td>
</tr>
<tr>
<td>Focusing On Notes</td>
<td>63</td>
</tr>
<tr>
<td>Identifying prototypes</td>
<td>63</td>
</tr>
<tr>
<td>Organizing Notes</td>
<td>63</td>
</tr>
<tr>
<td>Indenting notes in the outline</td>
<td>64</td>
</tr>
<tr>
<td>Checkboxes</td>
<td>64</td>
</tr>
<tr>
<td>Columns</td>
<td>64</td>
</tr>
<tr>
<td><strong>Using Charts</strong></td>
<td>65</td>
</tr>
<tr>
<td>The disclosure triangle</td>
<td>66</td>
</tr>
<tr>
<td>Focusing the View</td>
<td>66</td>
</tr>
<tr>
<td>Organizing Notes</td>
<td>66</td>
</tr>
<tr>
<td><strong>Using Timelines</strong></td>
<td>67</td>
</tr>
<tr>
<td>Using Attribute Browsers</td>
<td>67</td>
</tr>
<tr>
<td><strong>Using Treemaps</strong></td>
<td>67</td>
</tr>
<tr>
<td>Focusing the View</td>
<td>68</td>
</tr>
<tr>
<td>Visualization and Treemaps</td>
<td>69</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Agents</td>
<td>69</td>
</tr>
<tr>
<td>Quick Links</td>
<td>71</td>
</tr>
<tr>
<td>Composites</td>
<td>71</td>
</tr>
<tr>
<td>Queries and Actions</td>
<td>73</td>
</tr>
<tr>
<td>Actions, Expressions, and Rules</td>
<td>73</td>
</tr>
<tr>
<td>Expressions</td>
<td>73</td>
</tr>
<tr>
<td>True and False</td>
<td>74</td>
</tr>
<tr>
<td>Actions</td>
<td>74</td>
</tr>
<tr>
<td>Assigning Values</td>
<td>75</td>
</tr>
<tr>
<td>Conditional Actions</td>
<td>75</td>
</tr>
<tr>
<td>Side Effects</td>
<td>75</td>
</tr>
<tr>
<td>Repeated Actions</td>
<td>76</td>
</tr>
<tr>
<td>Edicts</td>
<td>76</td>
</tr>
<tr>
<td>Attributes and Constants</td>
<td>76</td>
</tr>
<tr>
<td>Attribute References</td>
<td>76</td>
</tr>
<tr>
<td>Constants</td>
<td>77</td>
</tr>
<tr>
<td>Designators</td>
<td>77</td>
</tr>
<tr>
<td>Group Designators</td>
<td>79</td>
</tr>
<tr>
<td>find()</td>
<td>79</td>
</tr>
<tr>
<td>Text Patterns</td>
<td>79</td>
</tr>
<tr>
<td>Regular Expressions</td>
<td>80</td>
</tr>
<tr>
<td>Wildcard Characters</td>
<td>80</td>
</tr>
<tr>
<td>Character Ranges</td>
<td>80</td>
</tr>
<tr>
<td>Anchors</td>
<td>81</td>
</tr>
<tr>
<td>The \ [backslash] Character</td>
<td>81</td>
</tr>
<tr>
<td>Parentheses</td>
<td>81</td>
</tr>
<tr>
<td>Back References</td>
<td>82</td>
</tr>
<tr>
<td>Local Variables</td>
<td>82</td>
</tr>
<tr>
<td>String Operators</td>
<td>83</td>
</tr>
<tr>
<td>+ (concatenation)</td>
<td>83</td>
</tr>
<tr>
<td>* (repetition)</td>
<td>83</td>
</tr>
<tr>
<td>.at(&quot;keyword&quot;)</td>
<td>83</td>
</tr>
</tbody>
</table>
.capitalize 84
.beginsWith 84
.contains 84
.countOccurrencesOf 84
.empty 84
.endsWith 85
.find 85
[contains] 85
.json 85
.lowercase 85
.paragraphs(N) 85
.replace(pattern, replacement[, arguments]) 8685
.reverse 86
.size 86
.split("pattern") 86
.substr(start, length) 86
.toNumber 86
.tr(characters, replacements) 87
.uppercase 87
.words(N) 87
.escapeHTML(string) 87
.format(value, format-string) 87
.idEncode(string) 89
.urlEncode(string) 89

Numeric Operators 89

Numeric Functions 90
Mathematical Operators 90

sum(), sum_if, avg and avg_if() 90

Converting Numbers to Strings 91

Color Operators 91

Color Components 92
Converting Colors to Strings 92
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converting Strings to Colors</td>
<td>92</td>
</tr>
<tr>
<td>Manipulating Colors</td>
<td>92</td>
</tr>
<tr>
<td><strong>Date Operators</strong></td>
<td>93</td>
</tr>
<tr>
<td>Date Properties</td>
<td>93</td>
</tr>
<tr>
<td>Date Operators</td>
<td>93</td>
</tr>
<tr>
<td>Date/time comparison operators</td>
<td>94</td>
</tr>
<tr>
<td>Locales</td>
<td>95</td>
</tr>
<tr>
<td><strong>Interval Operators</strong></td>
<td>95</td>
</tr>
<tr>
<td>Interval Operators</td>
<td>95</td>
</tr>
<tr>
<td><strong>List and Set Operators</strong></td>
<td>95</td>
</tr>
<tr>
<td>Sorting</td>
<td>97</td>
</tr>
<tr>
<td>List Operations</td>
<td>98</td>
</tr>
<tr>
<td>Building Lists</td>
<td>98</td>
</tr>
<tr>
<td><strong>Logical Operators</strong></td>
<td>99</td>
</tr>
<tr>
<td>Logical equality tests</td>
<td>100</td>
</tr>
<tr>
<td>Logical Group Operators</td>
<td>100</td>
</tr>
<tr>
<td><strong>Link Operators</strong></td>
<td>101</td>
</tr>
<tr>
<td>Link Operators</td>
<td>101</td>
</tr>
<tr>
<td>Finding Notes With Links</td>
<td>101</td>
</tr>
<tr>
<td>Making Links</td>
<td>101</td>
</tr>
<tr>
<td><strong>Eval() and Action()</strong></td>
<td>102</td>
</tr>
<tr>
<td>Notify, Twitter, Speak, and runCommand</td>
<td>103</td>
</tr>
<tr>
<td>Notify</td>
<td>103</td>
</tr>
<tr>
<td>Twitter</td>
<td>103</td>
</tr>
<tr>
<td>Speak</td>
<td>103</td>
</tr>
<tr>
<td>runCommand</td>
<td>103</td>
</tr>
<tr>
<td><strong>Locale()</strong></td>
<td>104</td>
</tr>
<tr>
<td>locale()</td>
<td>104</td>
</tr>
<tr>
<td>inheritsFrom</td>
<td>104</td>
</tr>
<tr>
<td>inheritsFrom()</td>
<td>104</td>
</tr>
<tr>
<td>hasLocalValue</td>
<td>105</td>
</tr>
<tr>
<td>hasLocalValue()</td>
<td>105</td>
</tr>
</tbody>
</table>

Tinderbox v7 Manual

7 of 206
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Export Operators And Groups</td>
<td>119</td>
</tr>
<tr>
<td>Feathering Your Nest</td>
<td>120</td>
</tr>
<tr>
<td>Favorites</td>
<td>120</td>
</tr>
<tr>
<td>User Badges</td>
<td>120</td>
</tr>
<tr>
<td>Character Badges</td>
<td>120</td>
</tr>
<tr>
<td>Badge Attributes</td>
<td>120</td>
</tr>
<tr>
<td>User Fills</td>
<td>121</td>
</tr>
<tr>
<td>A Distinct Look For Each Project</td>
<td>121</td>
</tr>
<tr>
<td>Color Schemes</td>
<td>121</td>
</tr>
<tr>
<td>Release Notes</td>
<td>121</td>
</tr>
<tr>
<td>7.2.0</td>
<td>122</td>
</tr>
<tr>
<td>HIGHLIGHTS</td>
<td>122</td>
</tr>
<tr>
<td>NOTABLE</td>
<td>123</td>
</tr>
<tr>
<td>MISCELLANEOUS</td>
<td>125</td>
</tr>
<tr>
<td>Inspector</td>
<td>127</td>
</tr>
<tr>
<td>Maps</td>
<td>127</td>
</tr>
<tr>
<td>Paste</td>
<td>127</td>
</tr>
<tr>
<td>7.1.0</td>
<td>128</td>
</tr>
<tr>
<td>NOTABLE</td>
<td>128</td>
</tr>
<tr>
<td>7.0.2</td>
<td>129</td>
</tr>
<tr>
<td>7.0.0</td>
<td>132</td>
</tr>
<tr>
<td>6.6.5</td>
<td>144</td>
</tr>
<tr>
<td>6.6.2</td>
<td>144</td>
</tr>
<tr>
<td>6.6.0</td>
<td>148</td>
</tr>
<tr>
<td>6.5.0</td>
<td>153</td>
</tr>
<tr>
<td>6.4.1</td>
<td>161</td>
</tr>
<tr>
<td>6.4.0</td>
<td>164</td>
</tr>
<tr>
<td>6.3.2</td>
<td>166</td>
</tr>
<tr>
<td>6.3.1</td>
<td>169</td>
</tr>
<tr>
<td>6.3.0</td>
<td>173</td>
</tr>
<tr>
<td>6.2.1</td>
<td>177</td>
</tr>
<tr>
<td>6.2.0</td>
<td>178</td>
</tr>
</tbody>
</table>
Help

What’s New In Tinderbox 7:

- **Composites**
- Beautiful built-in font families: Ideal Sans, Mercury, Ringside Condensed and Tungsten help make your documents more efficient. These state-of-the-art families make your work clearer and more legible – and more attractive as well!
- **Quick Links**
- and 175 other improvements you can see – and many more you can’t!

Also fresh: **Treemaps, Edicts, Simple Export**, improved **Badges**, faster map performance, **lookup tables**.

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 Treemaps
- Agents
- Quick Links
- Composites
- Queries and Actions
- Fetching Information From The Web
- Import
- Export
- Feathering Your Nest
- Release Notes
- License, Copyright and Colophon
- Recent Changes
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:

<table>
<thead>
<tr>
<th>attribute</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Among Others</td>
</tr>
<tr>
<td>Subtitle</td>
<td>Jo Walton</td>
</tr>
<tr>
<td>Color</td>
<td>poppy</td>
</tr>
<tr>
<td>Color2</td>
<td>light poppy</td>
</tr>
<tr>
<td>BorderColor</td>
<td>red</td>
</tr>
<tr>
<td>Caption</td>
<td>Hugo award winner</td>
</tr>
</tbody>
</table>

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, and Width.

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.
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 make also apply an action to notes when they are added to 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 an 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 Key Attributes table shows some the key attributes of the selected note. If the note has no key attributes, the table isn’t shown. The + button lets you add new key 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
6, 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.
- HTML 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.

**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. Key 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.
- **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 My 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 now include font settings and other attribute defaults as well as color definitions.

### Settings and Preferences

Tinderbox 5 had two distinct kinds of preferences. Document Preferences (⌘-8) provided access to a host of settings for the current document. Tinderbox Preferences (⌘-, ) provided access to settings that changed behavior throughout the application; in practice, most of these settings controlled default values for new documents. The two different kinds of Preferences were a frequent source of confusion.

In Tinderbox Six, settings for the current document are renamed Document Settings. The Document Settings window remains in its customary place in the Edit menu, and retains its old shortcut ⌘-8.

Tinderbox Preferences (⌘-, ) is not reserved for settings that affect all documents. In Tinderbox 6.0, 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
- Key Attributes
- User Attributes
- Aliases
- Intrinsic Attributes
- 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. 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. $HideKeyAttributes 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 \texttt{hh:mm:ss} –

\texttt{03:16}

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

\texttt{3h30}

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

\texttt{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 Key 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 key 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
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 . For example, if we have stored the location of an image in user attribute ImageName, we might export:

```html
<img src="">
```

Suggested Values

When working with key 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.

Key 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, key attributes might include author, title, and ISBN number.

A note’s key attributes are displayed at the top of its text pane, and at the top of the text window.
You choose the key attributes by setting the value of the attribute $KeyAttributes — just enter the names of the attributes you’d like to make key attributes, separating them with semi-colons.

To add an attribute to the note’s key attributes,

- Press the Key Attributes Button (labelled “+”) in the upper right-hand corner of the text pane
- Add the key attributes you prefer

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

Notes often inherit key attributes from their prototype.

**Key Attribute display differences**

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 key attribute:
• String data types appear as a simple text field.
• Boolean data type displays 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.
• 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.
• 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.

### Key Attributes

Tinderbox keeps track of hundreds of attributes for each note, but only few attributes are likely to be of chief interest to you. If you are making notes about some books, for example, key attributes might include author, title, and ISBN number.

A note’s key attributes are displayed at the top of its text window. You can choose the key attributes by setting the value of the attribute named **KeyAttributes** — just enter the names of the attributes you’d like to make key attributes, separating them with semi-colons.

To add an attribute to the note’s key attributes,

- select the key attribute from the Key Attributes menu in the text window sidebar

OR

- drag the key attribute from the Attributes palette and drop it into the text window

OR

- select the note and press Enter to view the note’s Rename dialog. Choose key attributes from the Key Attributes popup menu, and press OK.

Drag key attributes to up or down to rearrange them, or drag them out of the key attributes table into the text area of the note to remove them.

### Displaying Key Attributes
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 key attribute:

- Action, File and String data types appear as a simple edit box
- Boolean data type displays as a tick-box (ticked = true).
- Color data types has a color chip to the left of the edit box, which displays the color currently set. The value in the box may be a named Tinderbox color or hexadecimal color value.
- Date type attributes are show as the day, month and year plus hours and minutes, formatted according to the user's OS' System Preference for 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.
- Number data types are displayed as exponential numbers if very large or very small (0.000001 as 1e-06). If there are large numbers of decimal places, these may be truncated in the display (though not the underlying data).
- List and Set data types are shown as a single string with semicolons delimiting individual values.
- URL data type shows a globe icon before to the text edit box. If the icon is clicked, the currently displayed URL is opened in the user's default web browser.

**Key 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.

**Key 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.

**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 user attributes you have already defined. The inspector lets you choose the 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, $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.
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

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.

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
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 information 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.

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.

**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.
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 **HTML 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.

| 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.

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.

**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.

![Among Others](image)

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

![Among Others](image)

- 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.

**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 is often convenient to lock the position of large adornments. The attribute $\texttt{Lock}$ 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.

**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, simply 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.

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 $\texttt{InteriorScale}$. Typically, $\texttt{InteriorScale}$ 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 $\texttt{⌘}$ 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).

In this container, the title bar has been enlarged to permit Tinderbox to display a second line of
the title (the exact depth of the titlebar is set/stored in $TitleHeight).
If the title bars is sufficiently tall, it may also display the container’s text or summary table.

<table>
<thead>
<tr>
<th>Display Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>In maps and other views, Tinderbox typically displays the note’s title ($Name) to identify the note.</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>The attribute $DisplayExpression lets you change the note’s label without changing the value of its $Name.</td>
</tr>
<tr>
<td>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 $Name.</td>
</tr>
<tr>
<td>For example:</td>
</tr>
<tr>
<td><strong>DisplayExpression</strong></td>
</tr>
<tr>
<td>$Name+&quot; (+$WordCount+&quot; words)&quot;</td>
</tr>
<tr>
<td>$Name+&quot;:&quot;+format($DueDate,&quot;l&quot;)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hover Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
<tr>
<td>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.</td>
</tr>
<tr>
<td>To add or change the hover expression, open the Text inspector, and switch to the Hover pane</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>HoverExpression</th>
<th>typical display</th>
</tr>
</thead>
<tbody>
<tr>
<td>$WordCount+&quot; words&quot;</td>
<td>7500 words</td>
</tr>
<tr>
<td></td>
<td>7500 words</td>
</tr>
<tr>
<td>$WordCount+&quot; words\n\n+Text&quot;</td>
<td>Now is the winter of our discontent…</td>
</tr>
</tbody>
</table>

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.
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 Bar Example](image)

### 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.

![Outline Tree Example](image)
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 “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, its 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 display 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.

### Focusing On Notes

Focusing or “hoisting” a note focuses the view on it and its descendants. Expanding the view returns the focus to 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.

### Organizing 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.

**Checkboxes**

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**.

**Columns**

Select **Use Columns** from the View menu to display columns in the current tab, it is possible to enhance the view by adding one or more columns of system or user attribute data to the right of the title ($Name) data. This can be very useful for visual data review, spreadsheet-style.

To add a new column, click on the + button in the column header.
Tinderbox will add a new placeholder column. Double-click that column to select or change what is displayed in that column.

To hide columns in the current tab, uncheck View>Use Columns, or click the close button in the column header.

Note: the left-most column is always the Name (or DisplayName) of the note.

**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 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.

![Treemap Popover](image)

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.

**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.

### 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.

### 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.

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 it is now adjacent.

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.

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 \texttt{my:} is equivalent to \texttt{compositeFor(this)}. For example,

\begin{verbatim}
my:role(author)
\end{verbatim}

returns the notes with the “author” role in this note’s composite.

\textbf{Queries and Actions}

\textbf{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 \textit{expression} is simply something that has a value. For example, the expression \texttt{0.05*$Price} might calculate the sales tax on an item based on its price.

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

Actions may be performed at a variety of times.

- An \texttt{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 \texttt{Rule} is performed at frequent intervals, and affects the note that possesses the rule.
- An \texttt{Edict} is performed after a document is opened, and at infrequent intervals while the document remains open.

\textbf{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.

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"}

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.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>this</td>
<td>The note being examined by an agent, the note being added to an adornment, or the note whose rule is being run</td>
</tr>
<tr>
<td>next</td>
<td>The note that follows this note in outline order</td>
</tr>
<tr>
<td>previous</td>
<td>The note that precedes this note in outline order</td>
</tr>
<tr>
<td>prevSibling</td>
<td>The next older sibling of this note</td>
</tr>
<tr>
<td>nextSibling</td>
<td>The next younger sibling of this note</td>
</tr>
<tr>
<td>firstSibling</td>
<td>The first sibling of this note</td>
</tr>
<tr>
<td>lastSibling</td>
<td>The last sibling of this note</td>
</tr>
<tr>
<td>parent</td>
<td>The parent of this note</td>
</tr>
<tr>
<td>grandparent</td>
<td>The parent of the parent of this note</td>
</tr>
<tr>
<td>child</td>
<td>The first child of this note</td>
</tr>
<tr>
<td>child[n]</td>
<td>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.</td>
</tr>
<tr>
<td>lastChild</td>
<td>The last child of this note</td>
</tr>
<tr>
<td>randomChild</td>
<td>A randomly selected child of this note</td>
</tr>
<tr>
<td>current</td>
<td>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.</td>
</tr>
<tr>
<td>cover</td>
<td>The first note in the document.</td>
</tr>
<tr>
<td>source</td>
<td>The link source (used only in link expressions)</td>
</tr>
<tr>
<td>destination</td>
<td>The link destination (used only in link expressions)</td>
</tr>
<tr>
<td>agent</td>
<td>Available only in agent queries and agent actions; refers to the agent that is currently examining the note.</td>
</tr>
<tr>
<td>adornment</td>
<td>Available only in adornment queries and adornment actions; refers to the smart adornment that is currently examining or performing an action on the note.</td>
</tr>
</tbody>
</table>
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.

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

**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
- **all** all notes in the document

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

**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 XaaaaaaaaY.

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-Za-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,
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(^ 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.

### 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.

```javascript
var x,y; ...
```

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

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

### 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

```javascript
$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:

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

rather than the long and error-prone

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

---

Tinderbox v7 Manual

83 of 206
.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().

.contains

String.contains("pattern") - 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 matching sub-expressions are bound in order to $1...$99.

.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.

.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 .startsWith().

.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

.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

.json

Returns the string in JSON-encoded form. Characters that have special meaning in JSON are escaped with backslashes.

.lowercase

Returns a copy of the string, transforming all capital letters to lower case.
See also .capitalize and .uppercase.

.paragraphs(N)

Returns first N paragraphs of the string.

.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.

.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"

.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(?): world

If start is negative, the substring begins -start characters from the end of the string.

"Hello, world".substr(-5): world

.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()`.

`.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 '&lt;' and '&' is replaced by '&amp;'.

`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

\[
\text{format}($MyNum,2,7) \text{ is "}3.14\text{"}
\]
\[
\text{format}($MyNum,2) \text{ is } 3.14
\]
\[
\text{format}($MyNum,0) \text{ is } 3
\]

An option fourth argument specifies the padding character to be used if needed. for example

\[
\text{format}($MyNum,2,7,0) \text{ is "}0003.14\text{"}
\]

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.)
I: 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: 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($KeyAttributes","")

converts key 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,""

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

")

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.
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 its argument.
atan(N) computes arctangent, and returns its result in radians.
ceil(N) returns the first integer that is larger than N.
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.
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.
round(N) rounds the value of its argument to the nearest integer.
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.
sum(group,expression) computes the sum of a series of expressions. For example,

\[
\text{sum(child,}$WordCount)$
\]

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

\[
\text{sum(children(/agents/books),}$TextLength)$
\]
**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 “$1000” in the US.
**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 be 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 allows a color’s hue to be read or set using a value in the 0-360 range representing 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.weekday (1-7) Monday = 1, Sunday = 7. 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.)
I: 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: 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.

\[
date(2004,7,23,16,45) \text{ is 23 July 2004 4:45pm}
\]
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 apart, the result is 0 and not 1 as might be expected.

- **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 months 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 years that elapsed between date1 and date2, as a number. If date2 is the earlier date, the result is negative.

### 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

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, \( 4 \times MyInterval \) 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(start, end)** returns the interval between two dates, **start** and **end**.

### 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. 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 or 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 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.

**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.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.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. 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.empty** - true if the List or Set has no elements.

**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/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.size** - the number of values in the the list. 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

\[
\text{collect(children(/agents/books),$Name)}
\]

*expression* can be any expression, but is typically an attribute. For example,

\[
\text{collect(children,$Name)}
\]

\[
\text{collect(children,$Width * $Height)}
\]

**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

\[
\text{collect_if(children,$Status="Important",$Name)}
\]

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

\[
\text{collect_if( all, query, $Path)}
\]

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 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")
Link Operators

link[[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.

Making Links
Four actions create or remove links between notes.

```plaintext
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

```plaintext
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.

---

### Eval() and Action()

The `eval()` function evaluates an string and returns the result. For example,

```plaintext
eval("$Price*1.05")
```

returns the value of price augmented by 5%.

The `action()` function performs an action stored in a string.

```plaintext
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.
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.

Twitter

twitter(screen name, status) will post a new status to your Twitter account. The screen name must have been registered in System Preferences:Internet Accounts.

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.

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
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:

```plaintext
runCommand($CommandValue)
```

### 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

```plaintext
locale("en_GB")
```

sets the current locale to “British English”. Locale codes begin with a two-letter language code, followed by an 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

```plaintext
locale();
```

You may also save the old locale in an attribute for subsequent use. For example:

```plaintext
$OldLocale=local("en_GB");
```

```plaintext
... do various things ...
```

```plaintext
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.

### 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

```plaintext
inheritsFrom(/Plant)
```
is true for both Flower and Plant.

\[ \text{inheritsFrom}(/\text{Flower}) \]

is true for Rose, but false for Plant. You can also write

\[ \text{inheritsFrom}(\text{which}, \text{prototype}) \]

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!

**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 key 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 key 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

**Export**

<table>
<thead>
<tr>
<th>Simple Export</th>
</tr>
</thead>
</table>

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 Text screenshot](image)

- **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, 3 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 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,
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: `<a href="^nextSibling^">Next</a>`.
^indent(n,c) exports n instances of the string c. If c is omitted, exports n tab characters.

^lowerCase(...) and ^toUpperCase(...) 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 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)^`
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 HTML 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:
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 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
- TODO: currently we do this in Document Settings, but these may move to a new Defaults or Settings view.

### 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
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 text of the imported note.

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 $Completed.
$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 now always operate on the original note, even when **this** is an alias. Aliases share their text with the original notes.

The export element Tinderbox Help again returns the current document’s name.

An agent gathers a list of notes, and wants to use to create a list of those notes. This seems useful, but if the agent doesn’t export a copy of its children, would export links to pages that don’t exist. A new solution: if 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 , help/release_notes/7_2_0.html, and .

**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(7.2.0):: text tries to include itself 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 ratio of the original image.

When dropping multiple images into a map, the image adornments 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**

failed to initially bind **that** to the note referred to by **this**, and so some expressions of the form 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 Inspector 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

\[
\text{\$MyString=exportedString(this, "/myTemplate")}
\]

The second argument may also supply the template inline:

\[
\text{\$MyString=exportedString(this, "0")}
\]

\[
\text{\$MyString=exportedString(this, \$MyTemplateString)}
\]

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** submenu 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)spli comparatively view:(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):count` return 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):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 `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:

```plaintext
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

/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 tag is superseded by , 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 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 operator failed to evaluate expressions the refer to notes with non-ASCII names, such as , 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.
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 if $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.

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.

ALSO 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
help/release_notes/6_6_2.html 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 well 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 synthetic 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.

---

**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, 3 will be replaced
with the value of the width of the exported note.

^if(expression)…. ^endif and … ^else … 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”.

**ALSO 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

```markdown
inheritsFrom(/Plant)
```
is true for both Flower and Plant.

\[
\text{inheritsFrom(\text{Flower})}
\]
is true for Rose, but false for Plant. You can also write

\[
\text{inheritsFrom(\text{which, prototype})}
\]

which returns true if the note \text{which} inherits from the note \text{prototype}.

The operators \text{sum()} and \text{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 \text{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 \text{.count}.

\$InboundLinkCount and \$OutboundLinkCount now report separate values for the alias and original note. (1769)

**EXPORT**

\$HTMLQuoteHTML and the **Quote HTML** checkbox in the HTML Inspector were not always observed in text markup.

Changing the Quote HTML checkbox in the HTML 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)::template not found, or template includes itself. 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 $\text{BadgeSize} > 32$ 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.

**ALSO 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 calculations 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

```plaintext
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 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.

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 backslashes 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 possible 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 
\[
\text{sum(find(inside($Path(agent))), \ldots )}
\] (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, TbxCentralDocumentUpdates, now monitors this situation and handles the asynchronous process of marshaling updates.
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. `font name` sets the default TextFont for newly-created documents and `1-9` 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.
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

- Prototypes

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 .capitalize 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 map 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:

```perl
if($State=="AL") { $Region = "South";
```

This can be tedious. Instead, we create a Set, List, or String of keyword/value pairs, separated by colons:

```perl
$RegionList="AL:South;AK:NorthWest;..."
```

We can now write

```perl
$Region=$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:

```perl
$Region=$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 to 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 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 counterclockwise, 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 ExportAttribute 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 HTML 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 ( …. ) 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. 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.
Breadcrumbs 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 not 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:

```
1 0
```

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.
Explode has been revised to reliably delete delimiters. (1394)
Browse Links handles multiline text link anchors more gracefully. (1397)

6.1.3

SUMMARY
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 ) 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 usec 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

SUMMARY

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.

**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 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)
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 inbound help 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)

HTML 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

© Copyright 2014-2015 by Eastgate Systems, Inc. All Rights Reserved.

Tinderbox is © Copyright 2002-2014 by Eastgate Systems, Inc. All Rights Reserved.

Tinderbox™ is a trademark of Eastgate Systems, Inc. Storyspace™ and CIVILIZED SOFTWARE™ are also trademarks of Eastgate Systems, Inc. All other trademarks are the property of their respective owners and are used for informational and illustrative purposes only.

This help document was created and edited with Tinderbox Six. Other especially helpful tools included BBedit, Acorn, Skitch, Xcode, AppCode, DropDMG, Transmit, and CSSEdit.

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.

---

**Markdown Copyright (c) 2004 by John Gruber. All rights reserved.**

---

Eastgate Systems Inc.
134 Main Street
WatertownMA 02472 USA
Tel: (800) 562-1638 (617) 924-9051
Fax: (617) 924-9051
Email: info@eastgate.com
Web: http://www.eastgate.com/