

TechnoFeature™

Practice management and technology articles written by experts.

HotDocs Transformed – A Review of HotDocs 2006

By Seth G. Rowland, Esq.

TechnoScore: 4.6

1 = Lowest Possible Score; 5 = Highest Possible Score

www.hotdocs.com

INTRODUCTION: HOTDOCS AND ME

For the past eleven years I have dedicated my consulting practice to bringing “document assembly” to lawyers. As a “recovered lawyer” (University of Pennsylvania Law School ‘88 and veteran of Cravath, Swaine & Moore and Kramer Levin LLP), I see document assembly as central to the survival of the practice of law. As law becomes perceived as a commodity service, only by embedding knowledge in expert systems and practice automation can the lawyers of today deliver competitive high-quality legal services.

In the past decade, I have implemented document assembly solutions on a range of different document automation platforms. I have worked with HotDocs since version 4.2, serving at various times as a HotDocs Reseller, a HotDocs Consultant, a HotDocs Partner and now as a HotDocs Certified Independent Consultant (“CIC”). I have participated in several beta programs for HotDocs and posted hundreds of support e-mails to the HotDocs list since 2001. Members of my staff and I have logged several thousands of hours working with HotDocs and teaching others to use it. I can safely say that I know the product pretty well.

TEST LAB EQUIPMENT

In reviewing HotDocs 2006, I have tested it on an older Compaq Presario X1000 (1.5 Mhz Pentium M with 512 MB RAM) and a newer Dell Dimension 4600 (2.66 Mhz Pentium IV with 2 GB of RAM). Both systems run Windows XP, Service Pack 2, with Microsoft Word 2003.

WordPerfect users should note that LexisNexis continues to support WordPerfect templates. However, some of the new features (like Markup View) are only available for templates developed in Microsoft Word. Unfortunately for users of Mac OS X, no Macintosh version exists at present, although with the newer Macs running on Intel, you can dual boot into Windows XP and run HotDocs.

To check for compatibility issues between HotDocs 2005 templates and HotDocs 2006 templates, I installed Altiris Software Virtualization Solution (available from www.altiris.com). With Altiris, I could install each version as a separate application layer. This enabled me to switch between the two versions without rebooting.

TRANSFORMATION OF HOTDOCS

Starting with HotDocs 6 and culminating in HotDocs 2006, LexisNexis has completely rewritten the HotDocs engine. HotDocs has shifted from a powerful dialog-box based template-filler, into a true interview-driven application.

In HotDocs 5, the assembler would parse a template sequentially, and present a series of dialog-boxes, as needed, to complete the assembly. With HotDocs 6, the dialog-boxes were merged into a single interview with the ability to navigate from one dialog to another through an interview tree. This change began the shift from a template-driven system into a scripted interview-based application.

Over time, the developers added features to the interview. Via tabs, you could see a blank questionnaire and a fully completed questionnaire. In HotDocs 2005, they added a dynamic document preview. As the answers in the interview changed, the text in the preview changed. Variable text was

(Continued on next page)

marked in blue with hotlinks to actual questions. You could now use a comparison tool to visually compare the text under different scenarios. With the inclusion of the HotDocs database connection (and its enhancement), as well as the Time Matters connector, HotDocs entered the world of workflow, giving it the ability to easily pull in data locked in company and law firm databases.

HotDocs 2006 represents the culmination of this evolutionary process: easier, faster, more powerful, more flexible, and with a richer user experience. LexisNexis has also addressed the need for lawyers to better understand the templates through markup tools and coloring that hides the complexity of the systems.

PROS OF HOTDOCS 2006

Below you'll find just a few of the new features available with the latest release of HotDocs.

The Error Is Your Fault — And Now You Know Why

Perhaps the best new feature of HotDocs 2006 is that "IT WORKS." Stability issues plagued a number of the new features that emerged in HotDocs 6, but these have now been ironed out. HotDocs 2006 doesn't crash; it gives you an error message and takes you to the text of the template or the portion of the script that has caused the error. Most "crashes" result from template developer error (sorry folks, it is sometimes your fault). The problem was lack of information in a complex multi-template system; i.e. identifying the location of the error. The system will now halt the assembly and take you directly to the error.

In the template development environment, you can click a button to "Test Assemble" the document. In this Test Mode, you can use the Document Preview tab to test for errors under different scenarios and then use the editor to right-click on an offending variable or dialog to edit the script or text.

You Would Be Foolish Not to Use These Tools

Haven't you realized that "real" template developers

use toolbars? Seriously, template development is complex with myriad opportunities to make mistakes. In recognition of this, you now have a range of tools on the Word toolbar to assist in spotting potential errors in coding and adding codes:

- Markup View/Developer View

This feature enables the developer to hide all the "blue code" in a template and present it to the knowledge specialist with a simplified markup. Green square brackets denote the borders of conditional text. Blue italic text in square brackets denotes fields — very much like standard word merge-text fields. In this manner, you can edit the text of the template without the distracting conditional logic. And then, with a single click, you can convert the template back to developer view with all the HotDocs codes revealed. Making templates more readable is always a good thing.

- HotDocs Outliner

Have you ever wanted to know which questions were ONLY asked if the Borrower was a corporation, or some other special criteria? With the HotDocs Outliner, you have a visual tree of all HotDocs variables in a template. If they appear in conditional text, the outliner will show all the parent conditions which determine the relevance of that particular variable. This feature really works; but you need to be in Developer View (see above) to use it.

- Apply Colors -> Nested

In HotDocs 2005, the concept of colors for conditional logic was introduced, along with the ability to number rules. In HotDocs 2006, this feature has been refined with the addition of a nested color schema. This means that by visual references (green is top level), you can tell all the parent conditions for a particular block of text. In plain English, you can now know "why" that darn paragraph did not show up in the template ... or the converse. As part of this feature, HotDocs will throw an error if there is a missing END IF or an extra END IF and bring you to that point in the template to fix it. HotDocs also has a "sequential" coloring scheme.

(Continued on next page)

While the colors are pretty, this feature is only marginally helpful in diagnosing errors in template design.

- Other Refinements to Toolbar Buttons

Match Fields and Label Fields are useful tools. The first enables you to click on a conditional expression and find the matching END IF (or converse). The Label Fields formalizes what has been known by developers for years — that you can provide documentation inside a HotDocs fillpoint which does not affect the automation by using a “//” followed by the comment. Label Fields now enables you to control the visibility of those comments as well as supply a sequential identifier for each block of conditional text.

Bake the Cake and Eat It Too with Span Tags

How often have you wanted to change a section of an automated document, while preserving the automation so that you could reassemble the document? With the addition of a SPAN instruction, developers can now control document editing by marking sections of a template as a SPAN. This enables users to edit that section of text at the Document Preview tab of the assembly window. You can then save changes made during assembly to the answer file so that you can reassemble the same document later and have your changes reapplied. Used judiciously, this is a great feature. It moves the editing process into the Document Interview. For some users it may appear counter-intuitive: to edit an automated document, you need to “assemble” it, click on “preview” and then make your changes. This is not yet the “holy grail” of document assembly, but it gets closer.

The Razzle-Dazzle of Dialog Elements

HotDocs now provides a new Dialog Element component that enables you to add additional text, hyperlinks, buttons, graphics, lines, and spacing to dialogs more easily. HotDocs has gotten rid of the ugly undeclared component known as “additional text.” What used to be done by special scripting of “additional text” now is done much more logically

with this new component. You should really care about this because it makes the “user” experience so much simpler.

What the Heck Are “Dot Codes”?

HotDocs 2006 introduces a new feature called dot codes, which enables you to 1) format text results derived from computation scripts, 2) insert special characters in plain text and template text, 3) format variable prompts and additional text, and 4) punctuate non-repeated lists of answers. This stuff is “cool.” If I had the last item, the punctuation codes, I would be much poorer today, for I have charged clients thousands of dollars to implement proper punctuation in automated agreements.

SOME AREAS OF CONTINUING WEAKNESS

Speeding Up Complex Interviews

With so much now riding on the interview that needs to be rendered dynamically, certain aspects of HotDocs has slowed down. There are pauses when you exit a question before the screen refreshes that can last a second or two as HotDocs crunches through the implications of each change. HotDocs has provided a solution, a button to “turn off dynamic interview.” This will substantially speed the performance since HotDocs will not refresh the interview tree until you exit the dialog.

Answer File and Document Management

HotDocs has historically been built around single template assembly off a shared client or matter answer file. If you don't have a document management system (“DMS”) like Hummingbird, iManage, Worldox, or Time Matters, you will have to manage documents with Windows Explorer. This works fine when producing one document at a time. However, when you use a single interview to produce 20 documents (e.g. a loan package), HotDocs does not give you control (in the Interview script) to define the name and location of the answer file based on questions in the interview or to provide an output folder and location for the documents being as-

(Continued on next page)

sembled. A clunky workaround for the ASSEMBLE command exists, but more explicit control in scripting would go a long way in giving HotDocs workflow functionality.

Autosave an Answer File

The HotDocs API (Application Programming Interface) is very powerful. In the hands of an experienced database programmer, miracles can happen. However, most template developers are not experienced programmers; they generally come from the ranks of lawyers, paralegals, and legal secretaries and are self-trained. I would love it if some of the functions of the API could be incorporated as functions in the component manager — in particular, the ability to force a “SaveAs” which incorporates a HotDocs Variable into an Interview Script.

More Sophisticated Event Management

When a programmer talks about events, she is not talking about parties. HotDocs supports only two types of events: a global on change and a local on click. The on change happens when you change a variable. Every time you change a variable, it is checked against all scripts in the system. If there is a match, that script fires and things happen in your interview. By contrast, the on click event happens ONLY when you click a button. It would be better to have a middle ground, where you can tie an event directly to a single variable and have it fired “on exit” from that variable.

Rationalizing Complex Dialogs with SubTabs and More Variables in a Row

Computer monitors provide only so much screen “real estate.” For dialogs with lots of variables, the screen fills up with questions (and the infernal scrollbar emerges). This happens because HotDocs only enables you to put three variables in a row. This made sense when we had VGA screens with 640 x 480 resolution. But I run two 19-inch LCD panels with 1280 x 1024 resolution and my three variables in a row dialogs look ridiculous. The ability to layout additional variables would go a long way in making the dialogs more user-friendly. HotDocs could add

a new dialog element called a “SubTab”. By default, all variables in a dialog would be on the primary tab. However, once you inserted a SubTab, all variables that followed would reside on the SubTab. Tabs would have titles and run across the top of the dialog. So rather than scrolling (to see what you can see), you could click on the tab button.

WISH LIST FOR THE NEXT VERSION

Developers always want “more.” So I’ve compiled a speculative wish list for LexisNexis to consider.

Component File Plugins

HotDocs enables the developer to use “pointed component files.” In so doing, the developer can avoid “rebuilding” dialogs and variables for each template. With HotDocs 2006, the “pointing component file” can have a single computation component that contains a special interview script that calls on the master component file. This is the first step to making HotDocs more modular. Ideally, you’d have a master component file that at runtime can pull in elements from multiple other component files. This would enable you to develop special-purpose CMP files that function essentially like plug-ins to extend the power of the program.

Custom Functions

HotDocs has a powerful scripting language, with a number of special functions for text, number, and data manipulation. There are times, however, when there is no function available for the task at hand. The current approach is to create repetitive computations. So if a special format is required for fractional shares, a new computation script is created for “each” variable that needs to be converted into a fractional share. With a function, you can pass an argument to it, e.g. <fncFractions(MyVariable)> and then reuse that computation for handling other variables.

Style Sheets and Library Level Customizations

With HotDocs 2006 you can change the appearance of the HotDocs interview, selecting colors, font type, and font size. This approach works much like

(Continued on next page)

a Web-based cascading style sheet, enabling the same interview to be rendered differently depending on the settings in the styler. If the style settings could be stored at the HotDocs Library or the Component File level, content providers (and office managers) could provide a distinctive look and feel for each form-set (e.g. Red is for Real Estate and Blue is for Corporate).

EXPERIENCE WITH TECH SUPPORT

Technical Support for HotDocs falls into two categories: (1) application support, and (2) template development support. LexisNexis provides very effective application support: installation assistance, compatibility issues, support for links to word processors and document management systems. If the program won't load, or the PDF writer won't work, LexisNexis support technicians will work with you until it does.

However, LexisNexis draws the line at template development support. Any issues regarding the proper design or functioning of templates, development of interviews, formatting, etc. is handled either through paid consulting services offered by LexisNexis; through a searchable and extensive HotDocs knowledgebase; or through the HotDocs Listserver.

The HotDocs List (with over a thousand registered members) is a unique place for freewheeling discussion of ideas, problems, and best-practices. Unlike other lists, HotDocs developers (both those working for LexisNexis and those working independently) actively monitor this list and review and respond to posts by users. Many discussions started on-list move to off-list calls and exchanges and real solutions. The response time for posts can be measured

in minutes, whether it is from a developer or a fellow user with a similar issue.

CONCLUSION

If you take your law practice seriously and plan to stay in the law business over the next decade, you should seriously look into document assembly, and more specifically HotDocs. It is a truly great product that can dramatically enhance your productivity and profitability. Buying the software, however, is only the first step. Like my membership at New York Health and Racket Club, merely paying dues doesn't make you healthy and fit.

Copyright 2006 Seth Rowland, Esq. All rights reserved.

ABOUT THE AUTHOR

Recovered attorney Seth Rowland was named TechnoLawyer Consultant of the Year in 2002 for his contributions to TechnoLawyer on the subject of document assembly and law practice automation, and more importantly, his service to law firms around the country. He is a nationally known technologist whose company has helped many law firms and content providers build document assembly applications for both internal use and for resale. Please feel free to visit his [blog](#) for the latest on document assembly or the [video tours page](#) to see what such a system can look like. Basha Systems currently offers document assembly consulting services in HotDocs, DealBuilder, and Perfectus.

Contact Seth:

E: sgr@bashasys.com

T: 914-827-9173

About TechnoFeature

Published on Tuesdays, *TechnoFeature* is a weekly newsletter containing in-depth articles written by leading legal technology and practice management experts, many of whom have become "household names" in the legal profession. Most of these articles are TechnoLawyer exclusives, but we also scour regional legal publications for superb articles that you probably missed the first time around.