
Aaron Hardy
Whitman College

April 7, 2009
# Table Of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>3</td>
</tr>
<tr>
<td>BENEFITS OF PAPERLESS</td>
<td>4</td>
</tr>
<tr>
<td>DESCRIPTION OF PAPERLESS</td>
<td>5</td>
</tr>
<tr>
<td>REQUIREMENTS</td>
<td>6</td>
</tr>
<tr>
<td>HARDWARE</td>
<td>6</td>
</tr>
<tr>
<td>SOFTWARE</td>
<td>7</td>
</tr>
<tr>
<td>INSTALLATION</td>
<td>9</td>
</tr>
<tr>
<td>HARDWARE</td>
<td>6</td>
</tr>
<tr>
<td>SOFTWARE</td>
<td>7</td>
</tr>
<tr>
<td>ASSEMBLING A SPEECH</td>
<td>13</td>
</tr>
<tr>
<td>STEP ONE – OPEN a “Speech” Document</td>
<td>13</td>
</tr>
<tr>
<td>STEP TWO – OPEN FILES</td>
<td>13</td>
</tr>
<tr>
<td>STEP THREE – SEND blocks AND/OR cards TO Speech</td>
<td>13</td>
</tr>
<tr>
<td>STEP FOUR – ORGANIZE SPEECH</td>
<td>14</td>
</tr>
<tr>
<td>RECAP</td>
<td>14</td>
</tr>
<tr>
<td>SCREEN LAYOUT AND ORGANIZATION</td>
<td>15</td>
</tr>
<tr>
<td>PAPERLESS ORGANIZATION</td>
<td>17</td>
</tr>
<tr>
<td>FILE ORGANIZATION</td>
<td>17</td>
</tr>
<tr>
<td>DIGITAL TUB ORGANIZATION</td>
<td>18</td>
</tr>
<tr>
<td>BACKFILES</td>
<td>20</td>
</tr>
<tr>
<td>PRE-TOURNAMENT SETUP</td>
<td>22</td>
</tr>
<tr>
<td>IN-ROUND</td>
<td>23</td>
</tr>
<tr>
<td>PUBLIC RELATIONS</td>
<td>25</td>
</tr>
<tr>
<td>COMMON CONCERNS</td>
<td>27</td>
</tr>
<tr>
<td>THE DECISION TO SWITCH</td>
<td>27</td>
</tr>
<tr>
<td>PRIVACY/SECURITY</td>
<td>28</td>
</tr>
<tr>
<td>MACRO PROBLEMS</td>
<td>29</td>
</tr>
<tr>
<td>PRE-ROUND</td>
<td>32</td>
</tr>
<tr>
<td>IN-ROUND</td>
<td>32</td>
</tr>
<tr>
<td>CODE ANNOTATION</td>
<td>35</td>
</tr>
<tr>
<td>SENDTOspeech</td>
<td>35</td>
</tr>
<tr>
<td>SENDBLOCKToSPEECH</td>
<td>36</td>
</tr>
<tr>
<td>DELETEBLOCK</td>
<td>39</td>
</tr>
<tr>
<td>MOVEUP</td>
<td>41</td>
</tr>
<tr>
<td>MOVEDOWN</td>
<td>42</td>
</tr>
<tr>
<td>KNOWN ISSUES</td>
<td>44</td>
</tr>
<tr>
<td>FUTURE FEATURES/ADVANCED SUGGESTIONS</td>
<td>45</td>
</tr>
</tbody>
</table>
Introduction

As the policy coach at Whitman College, I switched our team to completely paperless in the fall of the 2008-2009 season. As of this writing, we have successfully completed an entire year of paperless with 7 teams traveling on the national college circuit. There have been a few hiccups and a lot of learning from our own mistakes, but on balance, our experience has been overwhelmingly positive. We have no intention of ever going back to paper, and based on the number of coaches and programs who have approached me about making the switch, I’m sure that carting around tubs will be an anachronism sooner rather than later.

This manual is intended as a brief “how-to” for programs considering the transition to paperless debate. While it is heavily geared towards the evidence intensive nature of college policy debate, the information here should be broadly applicable to both high school policy programs as well as other forms of debate, such as LD. Much of what is written here is an attempt at distilling the experiences at Whitman over the last year, to give other programs a leg-up at avoiding some of our early mistakes.

It’s my hope that our experiences (including this manual) can be the start of a collaborative process with any program that chooses to move forward with paperless. We’ve endeavored to make the Whitman system available to anyone who wants it in the hopes that the community at large can help reach a set of “best practices,” recommendations, and improvements on the original idea.

I remain indebted to several people who bear a public mention. First, Jim Hanson, on the cutting edge of debate technology for a decade before I started coaching – and a supremely supportive director through the transition. Second, the Denver debate team, who was completely paperless years before us and provided definitive proof of concept. Lastly, all my debaters, who acted as (mostly) willing guinea pigs, and trusted that this would work despite their many reservations.

A final comment for those still on the fence. Whitman had been bandying around the idea of going paperless for years before finally making the switch. Tests were run, many systems were tried out, and a laundry list of hypothesized problems were presented (including by me) as death-knells for the whole project. Ultimately, what made the difference in the success of our switch was not a radical improvement in technology or an advance of the underlying concept. Rather, it is attributable almost entirely to will power. Once the decision was taken that we would be debating paperless no matter what three weeks down the line, terminal obstacles simply became problems to solve, and the first tournament went off without a hitch. That Whitman Cohn/Straus could receive a first-round bid to the National Debate Tournament in their first year of our project serves, I hope, as a tangible demonstration that paperless can succeed broadly, even at the highest echelon of policy debate.

Any questions, suggested improvements, or criticism, about either paperless in general or this manual in particular should be directed to me at hardyat@whitman.edu. I would welcome feedback in any form.
Benefits of Paperless

The benefits of debating without the need to lug multiple 50+ pound tubs of evidence all over the country probably doesn’t bear much further explication. But, as we made the transition, we found ourselves continually unearthing new reasons we were glad we’d switched. Just to mention a few:

- **Cost savings** – Probably the number one factor informing our decision to switch at Whitman. Obviously, it saves all the money spent on paper, printing, copying, expandos, and other tub-related supplies. It also saves all the costs associated with checked baggage on airlines – with escalating fee structures, not an insignificant amount. Somewhat less obviously, it also saves money on the size of rental vehicles needed for to transport the average team. While obviously offset to a degree by the increased costs of the requisite technology (laptops, etc...), the net cost savings to our team just in the first year easily reach into the multiple thousands of dollars. In a time when many budgets across the country are at significant risk of being cut, paperless may soon become a necessity.

- **Ease of travel** – This should be obvious. Tubs weigh a lot, and airlines are evil.

- **At tournaments** – We have quickly found ourselves with a host of secondary benefits from the paperless transition. We have more prep time before rounds due to not moving tubs, we can more easily replicate standard work done in many different rounds, we get back to the hotel earlier because we don’t have to clean up, we can provide cite requests of every card read in a debate within minutes, it’s somewhat less likely my students lose their files...The list goes on.
Description of Paperless

For those who haven’t yet seen a team debate paperless, here’s a basic run-down of how it works.

All files are produced electronically using the same Word template, which incorporates both the normal formatting/organizing functions of a debate template, and a few added features specifically for paperless. Files are kept centrally organized in a digital “tub” comprised of folders, sub-folders, and individual files.

Each team carries three laptops. The debaters will each use a laptop to prepare speeches with, placing all cards that will be read for the upcoming speech into one Word document. This is accomplished rapidly by using a set of simple Word macros which facilitate both transferring blocks and cards between open documents, and organizing them into speech order.

Immediately prior to speaking, the debater will place their entire upcoming speech on a USB jump drive. This is first given to their partner, who copies it to their laptop to ensure a backup is available in the event of a tech failure.

It is then given to the opposing team. If the other team has their own laptop(s), they’re welcome to use them to view the file. If not, the paperless team uses their third backup laptop as a “viewing” computer for the other teams use for the whole debate. If for some reason the other team needs a second “viewing” computer, the paperless team can let the other team use one of their other laptops during their prep time.

The same will repeat for each speech (at least, those with cards). After the debate, the judge is obviously free to use either their own laptop or one of the paperless team’s to look at the evidence.

Are there differences between this and how a round proceeds under “normal” conditions with paper? Yes, although not as great as some have envisioned. More importantly, there’s now a large reservoir of experience to draw from which indicates one thing conclusively – it works. A list of commonly asked questions and concerns are included near the end of this manual, which should help give a more complete picture of how the debate transpires in actual practice.
Requirements

Running paperless on the Whitman system requires very little, and mostly stuff everyone already has. As the number of teams doing paperless expands and they develop their own systems independently, the specifications will probably change – but in its basic incarnation, paperless just requires a few laptops and a word processor.

Since the entire system is built into a Microsoft Word template, a basic familiarity with how to use templates is assumed. If you need a refresher, Naveen Ramachandrappa’s guide is available at [http://debate.uga.edu/research_guide/howto.pdf](http://debate.uga.edu/research_guide/howto.pdf). The Whitman template differs from the type produced in his guide, but the basic concepts are all still applicable.

Hardware

- Laptops – Right now, we’re of the opinion that this is unworkable without 3 laptops per team. This is to facilitate sharing evidence with the other team via a “viewing” laptop. While sometimes opponents have their own computers that they would rather use, it’s certainly not a universal. Over the course of the year, the majority of our opponents have opted to use our third laptop. I am sometimes asked what specs the laptops needed should have – I would say that essentially any machine capable of running Word will be more than sufficient. We’ve run paperless on 5 year old team laptops without any problems, both Mac and PC. More specific info about software, operating systems, etc...is included below. But, if you’re looking to purchase some extra laptops for your team, the cheapest available will probably suffice.

- USB Flash Drives – Nothing fancy, any kind will do. Each team needs at least 2, although having a store of extras on hand is advisable. They’re easily lost, occasionally break, and are cheap enough to buy in bulk.

Optional Hardware

Everything else we use isn’t strictly required, but is recommended for contingency planning.

- A portable podium of some kind. Since they’re not bringing tubs to tournaments, the debaters found they didn’t have anything to put their computer on while they stood to speak. The best solution we’ve found so far is a portable telescoping podium. We bought seven of these: [http://www.pctabletote.com/](http://www.pctabletote.com/) They’re a bit pricey (~$50/per) and to be honest not all that durable, at least at the hands of my students. But, they’ve served their function well, and they’re fairly light weight.
• External Mouse – The process of assembling a speech is much more rapid with an external mouse than a clunky touchpad, and it can come in handy while using the computer to speak from. It’s even possibly to program some of the higher-end mice with multiple buttons to automate many of the common tasks used in paperless and essentially prep a speech with one hand. I’ll leave that up to the tech savvy to figure out on their own.

• Power supplies – Given the paucity of available outlets in most classrooms (especially in high schools), it’s a good idea to make sure each team is carrying a power strip/surge protector, a 3 prong plug adaptor, and a heavy duty extension cord.

• Paper backup – Not something that we do anymore, but for our very first tournament, each team carried a copy of the 1AC and 15 pages of negative evidence, just in case there was a really, really horrible meltdown. Notably, there was not, and none of my teams bother anymore because their comfort level is high enough with our system. It might be helpful to carry a few copies of your plan text for disclosure reasons, as my teams find themselves constantly writing it out for the other team. Recently, my affirmative teams have just written it up on the blackboard before the neg team arrives (10 minutes later, with tubs) to simplify disclosure.

Software

The whole system runs within a single Microsoft Word template. Theoretically, any computer capable of running Word 2003 or later should work on any operating system, although it might require some tweaks, especially when making the jump between Mac and PC, Word 2003 to Word 2007, or XP to Vista. We have successfully tested it under all of these configurations.

That said, the recommended setup is Windows XP with Word 2003. It’s the simplest, easiest to maintain, the fastest running, and it’s what the system was designed under. This is what we’ve standardized for all 21 of our laptops at Whitman, and it’s worked great, even when using Macs in dual boot mode. Since this level of standardization may not be possible at many institutions, just keep in mind that small changes may be necessary.
The Template

The most recent copy of Whitman’s template is available at http://www.whitman.edu/rhetoric/tech/Debate.dot. The macros needed for paperless could easily be ported and modified from this document to your own template, or you’re welcome to use ours and just change the header.

Operating System

It will work on any operating system, but we’ve found that paperless works significantly better in Windows than in Mac OS. This is primarily because the Mac version of Word is terrible – it’s a lobotomized version, it runs slowly, and it handles macros poorly. I would recommend using Mac machines as viewing computers, for which they work fine. Another option is to have people with Macs run Boot Camp or Parallels and dual boot to Windows when they need to use paperless. For more comments on using Macs with this system, see the Installation section below. I prefer XP to Vista, but haven’t really had any problems with Vista.

Microsoft Word

Word 2003 is strongly recommended. It’s the last version of Word that wasn’t poorly designed. Word 2007 comes in second – It’s slightly bulkier, the interface is nonsensical, and macro support can be irritating, but it works fine. Word for Mac should be avoided when possible, as anyone who has used it can probably attest.

The one recent version of Word that WILL NOT WORK is Word 2008 for Mac. Microsoft removed support for VBA macros, which renders paperless impossible. They’ve said support will be added back in for the next version of Word (2010, perhaps?) but for now, avoid this version.

Another comment about Word 2007 – if you’re going to use Word 2007, it’s recommended that you always save as a Word 97-2003 document (.doc) instead of a Word 2007 .docx file. First, this is because the system was designed in Word 2003, and compatibility with the .docx format isn’t tested very well. The macros won’t even run with a speech file that ends in .docx, rather than .doc (explained below). Second, it helps maintain compatibility with older files produced in previous versions of word – which probably describes at least a portion of most team’s backfiles. This system can almost assuredly be adapted for native use in .docx format – I would be grateful for any feedback people have with doing it this way.
Installation

The only step required is to install the paperless template in Word and ensure macros are enabled. The exact process for doing this varies somewhat between operating systems and versions of Word, but is usually as simple as putting the template file in Word’s templates folder and telling Word that it exists.

Important note – While you can keep a copy of the template anywhere on your computer and create blank documents based on it, it’s still recommended that you put a master copy in the Word template folder. This is so that any file which was previously created in the template can find it if necessary. After installing it, I would put a shortcut to the template on your desktop or quick-launch bar to make opening a new document based on the template quicker.

Upgrading or changing the template is then as simple as replacing the old version with the updated one in the Word template folder.

Another note – When navigating to the template folder, Windows by default hides the folder in explorer. If the “Application Data” (XP) or “AppData” (Vista) folder doesn’t appear, you need to change your folder view options to “show hidden files and folders.” If you’re unsure of how to do this, check your operating system help file.

For the following examples, c:\ is the windows partition of your hard drive and “username” is the name of your currently logged in user. If you use multiple user accounts, make sure to install the template for each of them.

Windows

Depending on which version of Word you have (2003 or 2007) and which operating system (XP or Vista), the Word templates folder will be stored at one of the two following locations:

C:\Documents and Settings\username\Application Data\Microsoft\Templates.
C:\Users\username\AppData\Roaming\Microsoft\Templates

In Word 2003, put a copy of the template in this folder, and it’s installed. Make sure your macro security settings are turned to “low” under Tools – Macro – Security.

In Word 2007, templates can be installed from anywhere using the Add-Ins manager. Click Word Options in the main task bar, select “Add-Ins,” then select “Templates” in the “Manage” drop down box and click Go. Click “Add” and navigate to the folder where you stored the template. To make sure macros are enabled, go to Word Options – Trust Center – Trust Center Settings.
Mac

To install a template, put a copy of it in the Microsoft Office – Template – My Templates folder. The template will then appear in the Word Project Gallery.

Running paperless on a Mac requires a few tweaks:

- To use any of the built in shortcut keys, you might need to change the Mac OS setting to use the function key. This is found under System Preferences – Keyboard & Mouse.

- When using Mac Word, the macros will sometimes exhibit strange behavior, such as cutting off the first letter of a Block Title when moving blocks between documents. This is because some versions of Mac Word treat Page Breaks differently than all other versions in Windows. The macros could easily be reprogrammed to take account of this, but it would likely mean that it would not be cross compatible with Windows. In general, the problems involved are aesthetic – the macros will probably still work.

Setting Up The Desktop

There are a variety of other tweaks you can make to your computer to make it easier to use paperless, especially in an in-round situation. A few suggestions follow:

- Move the Windows taskbar to the left side of the screen, rather than the bottom. Usually this is accomplished by right clicking on it and unchecking “Lock the Taskbar,” then dragging it to the left side of the screen. This is so that you can more easily see a large number of word documents open without the taskbar grouping them. The taskbar can then also be resized horizontally to take up more or less screen space.
• Add your digital tub folder as a toolbar to the taskbar. To do this, right click on the toolbar, select “Toolbars – New Toolbar” and then navigate to the folder which contains your digital tub. This will create a toolbar with immediate access to all of your files. It can be dragged to show icons, or collapsed so that clicking one arrow will bring up a directory listing of all the folders in your tub. You can even create multiple toolbars for different folders, such as one for backfiles, one for current files, and one for the affirmative. A helpful tip – holding down “shift” while clicking to open a file will keep the taskbar open, rather than closing it and forcing you to re-click through the levels of a tub. This is very helpful when opening more than one document at a time.

• Turn off updates. It’s recommended that you temporarily turn off any program which will try to update itself automatically, including Windows Update, just for the duration of the round (or the tournament). This is to avoid the computer trying to reboot itself automatically, or popping up annoying reminders about updates in the middle of a speech.

• Turn off hibernation/standby/screen savers. You should set your computer to never go into sleep mode, hibernate, or turn the screen off, including when the lid is shut. This is usually accomplished by right-clicking on the desktop and selecting “Properties.” The power settings are included on the Screen Saver tab, under the “Power” button.

• Turn off any programs which could interfere with using the macros by utilizing “hot-keys.” Some programs running in the background of your computer may already have assigned certain keystrokes that are used by the paperless macro. A prominent example is certain NVidia graphics cards which assign hotkeys which rotate the screen. This can be turned off by right-clicking the desktop, selecting “Graphics Properties – Hot Keys” – and selecting “Disable Hot Keys.” Other programs may also exhibit similar behavior – if macros aren’t behaving correctly, check your computer for other hidden programs which might have hot-keys.
• Consider a separate user account for paperless. Since most people will not want their screen set up the same way for every day use as for paperless, consider adding a separate account in Windows used exclusively for debate. This will allow you to set up the desktop to your specifications without interfering with everyday work.

• Clear the desktop of non-essential items. Since the desktop is used to save Speech documents and copy them to the USB drive, it’s helpful to have it cleared of extraneous files. Useful things to keep on the desktop include: A folder with the most recent tub, a shortcut to the paperless Template, a pre-made blank Speech.doc, a folder for the current tournament, and a shortcut to your external USB drive. See the screenshot above for a sample layout.
Assembling a Speech

This is the real “nuts and bolts” of how to use paperless. At root, the idea is very simple. You take blocks and cards from a variety of files, and put them all in one master “Speech” document, where you further organize them into the order you expect to read them in your speech.

In reality, this process could be accomplished with nothing other than the built-in Cut and Paste functions in Word. However, this is far too time-consuming to be practicable, as the effort required to switch between word documents, select the exact text you want to copy, and then move to the correct place to paste it would eat up an inordinate amount of prep time.

This is where the paperless macros come in. Assembling a speech using this system is a simple four step process

Step One – Open a “Speech” Document

This is the most important step. Open a blank document based on the template, then save it to the desktop as “Speech.doc” Note that the S in Speech is capitalized. This file must be saved exactly as Speech.doc, because it’s how the macro knows which document to “send” all the blocks you select to. This document must be open for the rest of the steps to work – if it’s not, the macros will warn you.

Step Two – Open Files

Open any files you want to use from your digital tub. This doesn’t have to all be done at once – you can open and close files as you go, as long as your Speech document remains open, you can send things to it. Note that any files you want to use in this process must have the requisite paperless macros included – in other words, they must be based on the same template as Speech.doc, or have had the macros manually added.

Step Three – Send blocks and/or cards to Speech

This is accomplished through the use of two macros. In Word, these are named SendBlockToSpeech and SendToSpeech. Both of these macros take cards from your current document, and “beam” them into your open Speech document, pasting them in at the current cursor. The former will send one “block” at a time, the latter will send any amount of highlighted text, whether a single card, multiple pages, or even an entire file.

SendBlockToSpeech. By default, this is assigned to the hotkey Ctrl-Alt-→. The symbol stands for the “Right Arrow Key.” This macro will send everything between the previous and next manual page break. In Word, manual page breaks are added with Ctrl-Enter. The macro ignores “soft” page breaks which occur naturally in Word when it would print on the next page, and it ignores all styles and formatting – such as “Heading 1” block titles. Since the start of the next Block Title in your file usually coincides with a page break, this has the effect of sending one “block” at a time. The macro also doesn’t differentiate where your cursor is inside the block – it can be in a card, a tag, or at the beginning of the block title.
SendToSpeech. This is default assigned to Ctrl-Alt-←. This macro will send any text which you currently have highlighted. It doesn’t differentiate by size, so this macro allows you to send over a single card, a tag, or select multiple blocks at once. It also pastes into Speech at the current insertion point, which lets you quickly send a single card to the middle of another block, for example.

**Step Four – Organize Speech**

After you’ve sent as many cards or blocks as you want to Speech.doc, you just need to organize them as you want to have them for your speech. This is accomplished with two other macros, MoveUp and MoveDown. MoveUp will move an entire “block” above the previous block, while MoveDown will do the inverse.

**MoveUp**. This is assigned to Ctrl-Alt-↑. This will move everything from the previous Block Title to the next Block Title one “slot” up, above the previous Block Title. The macro ignores your cursor position inside the block, so it can be at the start of the Block Title or anywhere in-between. For the purposes of this macro, a “Block Title” means anything formatted as “Heading Level 1” by Word. Note that this is slightly different than the “Send” macros, which were based on page breaks, rather than headers. Also be aware that you cannot move a block “up” higher than the top of the document. Trying to do so will (at the moment) give you an error message.

**MoveDown**. This is assigned to Ctrl-Alt-↓. This does the exact inverse, but moves the block down.

There’s also another macro, DeleteBlock, which will delete the current block, between manual page breaks. By default, this is Ctrl-Alt-End. This allows you to quickly remove blocks which you determine in the process of organization that you’d rather not have in the speech document.

Taken together, these two macros let you quickly move blocks into whatever order you would like them for the speech.

**Recap**

There are only five macros needed to assemble a speech paperlessly.

- **Ctrl-Alt-→** – Sends over one “block” at a time, between manual page breaks.
- **Ctrl-Alt-←** – Sends over any amount of highlighted text.
- **Ctrl-Alt-↑** – Moves the current block up one position in the document.
- **Ctrl-Alt-↓** – Moves the current block down one position in the document.
- **Ctrl-Alt-End** – Deletes the current block, between manual page breaks.
Screen Layout And Organization

• Most of my debaters find that it helps to conceptualize your desktop like a desk which moves workflow from left to right. Starting with the taskbar along the left side, which keeps the open documents organized, try to use approximately the left half your screen for all the open files. Then, keep Speech open on the right side of your screen. Even if you can’t fit two windows open side by side on your desktop, try to leave a little bit that doesn’t overlap.

• Use “Normal” view, also known as “Draft” view in Word 2007. This removes the header, footer, and extraneous white space from your document view. It also makes it much easier to see manual page breaks, which are vital for the operation of the send macros. I would recommend using normal view essentially 100% of the time.

• Use the document map – Most of you will be familiar with the document map from creating files electronically. In paperless debate, its importance is elevated even further. It functions like an index, allowing you to see the entire file at once – and makes moving around within a file substantially faster. It also enables you to see where you’re moving blocks to while using MoveUp and MoveDown. I’d recommend leaving the doc map turned on in all documents, at all times. If you’re not familiar with the operation of the document map, it’s suggested that you read the template tutorial linked above.
• Use “Reading” view (Word 2003), or “Full Screen Reading” view (Word 2007) when actually giving your speech. This view will allow you to see one or two entire pages at a time, and quickly move through your document by using the arrow keys to move a page at a time, instead of scrolling or using Page Down. If only one page is visible, try shrinking the size of the document map horizontally until two pages appear.
Paperless Organization

One of the most important aspects of ensuring that you can use the paperless system hassle-free is effective file organization. Much like having dozens of reams of paper without indexes or block titles thrown at random into a tub would make debating impossible, a single folder on your hard drive with 200 Word documents cryptically labeled “updates.doc” or “politics.doc” wouldn’t fare much better.

Using digital evidence to its maximum advantage has two parts – individual file organization, and tub organization. I’ll give a few tips for each.

File organization

Paying meticulous attention to how your files are constructed in Word is important to ensuring smooth operation of the macros. This extends to making sure Block Titles are formatted properly, page breaks are in the right place, the document map accurately reflects the file’s contents, etc...

- Using Page Breaks. Most importantly, ensure that you are using them, every time. You should resist the temptation to just press “Enter” until the next page starts. This is true even if the previous page runs right to the bottom, and Word automatically inserts a soft page break. This is because the macros will be unable to function correctly without them. Ctrl-Enter will insert a manual page break, and should be done before the start of every new block title.

  Keep in mind you can also use this to your advantage. For example, you can safely ignore a card running just over a printed page, because as long as you haven’t inserted a manual page break, the macro will send over the entire card. Another example is a multi-page 1NC shell. Instead of including 3 or 4 duplicative block titles, it’s possible to include 3 or 4 pages under the same Block Title, inserting a manual page break only at the end of the shell. Then, when using SendBlockToSpeech, the entire shell can be sent with one key stroke.

- Use the Document Map as your index. My debaters rarely create an index on their files any more – instead, they pay more attention to making sure the document map is as organized as possible. Especially for files under 100 pages, it’s easy to see most or all of the files contents with a glance at the doc map. One of the most important aspects of keeping the doc map organized is the selective use of white space – created by formatting an empty line as a Block Title (Heading 1).
• Modularize your files. It’s helpful to walk a balance between creating files which are too large to be easily digested with the doc map, and files which are so small as to require dozens of individual word files. An example is writing an affirmative – where instead of creating a 1000 page aff file, you could create a file for answers to disads, a file for answers to CP’s, etc…and then keep them all in a separate folder, much like an expando.

**Digital tub organization**

There’s an infinite number of ways to organize your files on the computer – but here’s a basic run-down of how we’ve done things so far, which has worked fairly well for us. The entire tub is stored on a network server run by the college, which gives us access to it in our squadroom, as well as offsite using FTP. If a similar setup is impossible, there’s many other alternatives, such as designating one desktop as the master copy, purchasing a Network-Attached Storage device, or even just using a gmail account.

Our digital tub at Whitman is divided into four basic areas.

1) Archived Backfiles – this is a separate folder, organized by year. Since each season gets a separate folder, it’s easy to keep a record of each topic. The other three areas go in a separate folder for the current year, which gets moved to the backfiles at the end of the season.

2) Files sorted chronologically – This section has a subfolder for each tournament we attend. Each file that comes out before that tournament has a copy placed here, to ensure we have a record of when files were completed.

3) Files sorted by subject – This constitutes the main part of the digital tub. It has around 15 folders, almost all with subdivisions, into which every file that’s produced is sorted. This includes folders for “Disads,” “Critiques,” “Case Negs,” etc…Relevant backfiles are just sorted into the appropriate folders.

4) Private Tubs – this section has a separate folder for each team on our squad. It’s designed to be a place where each debater can put their own reorganized versions of files, highlighted copies of files, personal blocks, a completely revamped version of the main tub, or anything else they see fit to do with it. It’s also a place where they are encouraged to upload their “Speech” documents from each tournament, sorted by round, so that they have a record of each speech given over the course of a year.
Here’s a mock visual representation of the folder structure of our online tub. Indents represent a level of subfolders:

Backfiles
   2005 – China
   2006 – Courts
   2007 – Middle East
   etc...

2008 – Agriculture
   000 – Files By Tournament
      1 – Gonzaga
      2 – GSU
      3 – Kentucky
      etc...

Case Negs
   Biofuels
   CAFO’s
   Dairy
   etc...

CP’s
Case Negs
   Biofuels
   CAFO’s
   Dairy
   etc...

zzz – Private Tubs
   Whitman AA
   Whitman BB
   etc...

Some of these folders have only a few files, while others have dozens, organized with numbers in the file names to give an “expando-like” feel to the folder.

Here’s a hypothetical representation of a case neg folder:

Biofuels Neg – Case 1 – Solvency.doc
Biofuels Neg – Case 2.0 – Food Prices.doc
Biofuels Neg – Case 2.1 – Specific Food Prices Countries.doc
Biofuels Neg – Case 3 – Environment Advantages.doc
Biofuels Neg – Case 4 – Brazil Advantages.doc
Biofuels Neg – CP 1 – Tariff Only CP.doc
Biofuels Neg – CP 2 – Japan Rice CP.doc
Biofuels Neg – DA 1 – Cellulosic DA.doc
Biofuels Neg – DA 2 – Natural Gas DA.doc

I also maintain a team Gmail account. This is useful for rapidly distributing files produced at a tournament, files which miss the cutoff time to be uploaded to the server, etc...
Backfiles

One of the major questions people pose about making the transition to paperless is what becomes of old backfiles. Whitman is fortunate to have 12 years of backfiles in a backwards compatible template that was built on for the development of paperless...but I sense that this is the exception, rather than the norm.

There are two scenarios for backfile integration – Paper/PDF backfiles, and backfiles produced using a template (or several templates).

In the first instance, the short version is that you’re out of luck. While it’s possible to cut and paste pars of PDF or TIF files into Word, it doesn’t work very well, and can’t easily be integrated in with the rest of the paperless system. Paper backfiles are, well, on paper...Fortunately, I sense that the majority of all files used in debates are produced during the current year – so as long as the switch to paperless is made in-between seasons, the impact of the former problems should be minimized.

If, on the other hand, you have electronic backfiles which were produced using another Word template, they should integrate relatively hassle-free. There’s two ways to port your current files to paperless.

1) The backward way. Theoretically, you could transfer the needed macros to your old template, potentially make some tweaks to the code to account for formatting changes, and have it work. On the other hand, this would require transferring the macros back into any different template that another member of your team used, and then dealing with the programmatic difficulties involved when collating blocks and formatting differences from a variety of different sources. I’d recommend option b.

2) The forward way. Design a standardized template (or steal ours) that will be used by every member of the team, and make sure it has all necessary macros. Then, paste any relevant backfiles into the new template and resave them in the new format.

The real key to whether this will work effectively is whether the templates you’re trying to combine into the new format are compatible. The most relevant question is whether the old templates formatted Block Titles as Heading 1. As long as they did, it will likely paste in with minimal hassles, maintain the integrity of the document structure, and work more or less without a hitch.

We’ve successfully tested the macros on a variety of other templates. It’s been possible to get them working, sometimes with a few small changes to the macro code. In general, the primary conversion difficulties are only aesthetic – for example, pasting old blocks into the new template will make document elements (such as tags) occasionally appear in the wrong font. This is easily fixed by someone with a degree of Word savvy, using either the Styles and Formatting tools, or even just a Find and Replace of the offending styles.
Even when the aesthetics are left unfixed, the basic functionality of paperless is usually intact – transferring to Speech.doc and moving blocks up and down. The primary thing which I’ve seen not work correctly is MoveUp and MoveDown, especially when the backfile involves strangely formatted Block Titles or section headings. Then, it can occasionally move the wrong block, or not move them at all. There’s usually a workaround if you play with the file for a bit.
Pre-tournament Setup

This is just a brief pre-tournament checklist to make sure you’re ready to debate paperless. Mostly, it’s just a distillation of the preceding advice.

- Drills, drills, drills – Most of the difficulties I’ve seen my debaters have with paperless so far have been the exclusive product of a lack of practice. While I firmly believe that debating without tubs speeds up almost every aspect of the debating process, including prep time, there’s a definite element of relearning involved. While memorizing the hotkeys and understanding the basic elements only takes 5 minutes, becoming proficient enough with the system to deal with contingencies or be extremely quick takes hours. My debaters said their comfort level increased significantly after their second full practice debate – but they have still needed lots of work as the season progressed on things like efficient USB transferring, file organization, and keeping a consistent workflow on the desktop. While this is a significant time investment, it’s useful to remember the amount of time saved vs. hand-labeling manila folders and printing expando indexes.

- Make sure you have enough laptops – As mentioned above, I think each team needs a minimum of three laptops. It’s probably also a good idea for the team at large to have some backups, as debaters seem to be kryptonite to technology.

- Check that each computer works – my debaters seem to have a knack for ensuring that any computer purchased the night will be infected with a virus, rife with spyware, and running too slowly to effectively open multiple word documents before the morning arrives.

- Clean off the desktop and USB drives. Since the desktop is used as a work space where Speech files are saved and copied, digital tubs are kept, etc...It’s a good idea to clear everything else off, at least for the duration of a tournament. If there’s an inordinate amount of clutter, it can just be temporarily stored in a new folder. It’s also a good idea to create a folder for each tournament, so you can store (and label) Speech documents and update files as they are produced. It’s also a good idea to start with a clean USB drive before each round – this both ensures there’s not confusion over which file to open, and prevents the accidental spread of either viruses or previous files.

- Make sure each laptop has the most recent version of the digital tub. While archived backfiles can be put on each computer at the beginning of the season and left relatively untouched, the master copy of the digital tub is constantly in flux. It’s a good idea to have a complete copy on a USB drive before leaving for the tournament, just in case one (all) of the debaters forget to update their local copies.

- Make sure each computer is set up for paperless – toolbars created, hibernation/screen saver turned off, updates turned off, screen rotation turned off, etc...
In-round

After practicing with the whole paperless process a few times, I’ve found most debaters are quickly comfortable enough with it to not dread the idea of trying it in a competitive setting. That said, here are a few suggestions (some duplicative with earlier tips) to make the in-round process go smoothly. Take special note of the admonition below to save files to the hard drive before copying them to the USB drive. In roughly chronological order:

- Explain to your judge and opponents that you’ll be debating paperless, and what that might entail for them. Explain the viewing computer, the USB jumping process, and any other logistical issues. Since the whole notion of paperless debate is still relatively new for many teams and judges, some are bound to have questions or concerns about the impact on evidence sharing, length of round, etc...

- Make sure you know if the other team wants to use their own computers to view your evidence, or would prefer to use the viewing laptop – they take time to plug in, set up, etc...so don’t wait until the round has already started.

- If you have concerns about the other team “stealing” your evidence instead of just looking at it on the USB drive, talk to them about it in advance. You can also ask the other team not to “look ahead” in the document while you’re giving your speech if this is something which concerns you.

- Plug in and set up – when you get to the debate, immediately get both computers plugged in so there’s no battery problems. Figure out if you need extension cords, power strips, etc...Also, set up your laptop stand in advance if you need it for a podium.

- Get a USB drive plugged in to the computer in advance, with a folder open to quickly facilitate copying to it from the desktop. This is important to facilitate rapidly moving files – it can take a long time for a computer to recognize a drive, open a folder, etc...

- Both debaters should get a “Speech.doc” open on their computers. It should be saved on the desktop. Remember the capital S. You can also use the top page of your speech document or a blank “notepad” document to jot down coaching notes.

- Minimize the number of open Word files. If you’re clearly done with a file, close it. Word can pretty easily handle a large number of open documents, but the more you push it, the more likely it is to freeze, or become unbearably slow. This is especially true right before you’re about to speak – the most important time for your computer to not have tech problems.

- Once you’re done prepping your speech, save it. Then, “Save As” and rename it on the desktop as “1NCSpeech.doc” or whatever your speech is. You should then drag and drop the file to the jumpdrive. Very important note: Do NOT save the file directly from Word on to the jumpdrive. If you do, Word sometimes gets very angry when you remove the drive. The only tech problems we’ve had so far are related to making this mistake.
• Work out in advance with your judge whether you can “stop prep” before doing the USB transfer, or whether they consider that prep time. Trying not to irritate the judge is generally good practice, never more so than when doing something alien to “normal” debate.

• You should first hand the jumpdrive to your partner, who should copy it to their desktop and open it on their computer as a backup. Then you should set up on your podium, give the roadmap, etc... while your partner hands it to the other team or sets up the viewing laptop for them.

• It’s helpful to maintain a consistent naming convention for all your speech docs, as well as a consistent organizational scheme on your desktop and jumpdrive. Since every speech starts out as “Speech.doc” it would quickly become impossible to keep them straight unless they’re given more accurate names and organization.

• Use “Reading Layout” during your speech – as described above, this makes scrolling through a document much easier, using only the arrow keys. Note: If you accidentally click inside the text of a card, it will disable moving between pages with the arrow keys in favor of moving through the text. To fix this, press the Escape key once. Alternately, Page Up and Page Down should always work to advance pages.

• After the debate, you can politely remind your opponent and/or judge to delete their copies of your speech documents if anyone transferred them instead of leaving them on the USB drive.

• Keep every speech you give. This makes writing blocks, sharing work and intel, etc...much easier.

• What do I do if I have a catastrophic crash? – If you’ve done everything right, there won’t be a problem. If something really insane happens, like in a speech, then you should be able to quickly switch to your partners laptop if you’ve done the backup process correctly. In an absolute disaster, you should beg for mercy from the judge while you figure out what went wrong. Hopefully, they’ll be nice…
In the very beginning of our paperless transition, Whitman proposed two “community norms” we thought would help facilitate fairness during paperless debates, as well as alleviate some of the concerns our debaters had. Neither norm was objected to by any of the people we debated over the course of the year, and they seem to have been reasonably accommodated by most if not all of our opponents.

I list them again here as no more than an ongoing request – it’s certainly the case that practices will evolve along with the more widespread utilization of paperless debating, and these types of norms will likely take care of themselves, in time. More importantly, we feel it’s the burden of the team pushing a new practice (paperless) to bear the brunt of the responsibility for accommodation should anyone disagree.

Nonetheless, we feel the following practices would be best for competitive equity:

- The opposing team should, to as reasonable a degree as possible, minimize “looking ahead” in the speech document to try and gain a competitive advantage by figuring out what will be read later in the speech. This is especially applicable in rounds where something such as a new affirmative is being read. While obviously only so practicable, we feel that an honest attempt is still better than nothing.

- Opposing teams or judges who opt to transfer the “speech” document to their personal computers should delete them at the conclusion of the debate. We feel that taking evidence wholesale is the equivalent of taking a paper file. We’d hope the majority of the community would agree that stealing files crosses the line, especially given the easy availability of cites.

Two other issues bear mentioning in relation to paperless teams interacting with the non-paperless world, especially judges.

- Prep Time – Some judges have expressed concern that the process of jumping files, setting up computers, etc...takes too much time. In particular, they seem to be frustrated that it appears as if the paperless team is “stealing prep” while waiting for something such as a Word document to open on the viewing computer. While a legitimate concern, I think it is misplaced, for several reasons. First, after a season of debating with seven paperless teams, I can say that I’ve noticed zero difference in the average length of time it takes to conclude our debates vs. rounds involving only paper. Secondly, I would say that paperless more frequently saves time, by eliminating the “stolen prep” involved in giving each teams evidence back to each other, searching under desks for piles of misplaced 2NC cards, or looking for the lost CP text. My hunch is that this time is significantly greater in the world of paper, but judges are used to it taking place, while they are not used to the time involved in jumping files.
Tech problems – While we have yet to have any truly terminal tech problems in the middle of a round, it’s probably inevitable that it will happen at some point, if not to us than to some other debate team making the transition. To a certain extent, this can’t be avoided – but it’s probably worth thinking through how the judge should deal with it. If a few debates a year have to conclude 5 minutes later while a debater gets one free “reboot,” it seems worth the myriad other benefits it brings to the debate community. In general, I would just hope for patience on the part of the judges and debaters when this problem inevitably arises.
Common Concerns

When told they would be debating paperless at their next tournament, my debaters immediately raised a host of questions and concerns, most of which seemingly took the form of “what if’s.” This section will hopefully answer the most commonly asked of these. Much of the advice given here is repeated elsewhere, but this is an attempt to present it in a form more easily accessible when something goes wrong.

The Decision To Switch

What do I do if my computer crashes?

This is far and away the most frequently voiced concern. As mentioned earlier, 14 debaters at Whitman debated an entire year without a crash – but it is probably inevitable at some point. First, keep in mind that it’s important to minimize the chance of a crash by practicing good preventative care on your computer. Ensuring your operating system is up to date, that you’re running anti-virus software, and that the machine is physically well taken care of will go a long ways towards avoiding any problems.

That said, if it does happen, there are several backups in place. Since each debater puts their speech on a jump drive and gives it to both their partner and their opponents, there should be at least 2 other computers looking at the current speech at any given time. After doing contingency drills with my debaters, they can swap out a crashed laptop in no more than a few seconds. If a computer crashes before the speech, a reboot will usually solve the problem – and if the debater has been saving regularly, not much work should be lost.

This concern is also not really unique to paperless anymore – many teams flow on their laptops, or frequently read a card or two during a debate. While not as catastrophic as the previous examples, the debate community at large will have to eventually develop a set of norms surrounding how judges and competitors deal with the occasional crash.

Can’t we switch half-way and still use some paper?

It’s obviously possible to develop a debating method that falls somewhere in-between fully paperless and relying entirely on tubs. Several suggestions have been made, such as printing evidence before each speech, printing the evidence for the judge after the round, carrying only the most frequently used files in one tub, carrying everything in tubs except a few backfiles, etc...

At Whitman, we toyed with all of these ideas, but ultimately decided that they defeated much of the purpose for us. Since the primary motivations for our switch were to eliminate baggage costs, printing costs, etc...and simplify the process of traveling with a large number of teams, creating another headache by making each team carry a printer, for example, wouldn’t have met our needs. While I’m sure that some teams will effectively implement a hybrid system, we have no regrets about our decision to remain completely paperless.
Why is support for Mac Word spotty?

Blame Microsoft. The short version is that Mac Word seems to treat certain elements of the document differently than Windows versions, especially when accessed programmatically in VBA. In particular, where Windows versions use manual page breaks (^m in VBA code terms), these sometimes appear to Mac Word as section breaks. This means that the paperless macros, which rely on finding manual page breaks, sometimes don’t operate correctly.

This is certainly a fixable bug, but hasn’t yet reached the level of top priority. Given that Mac Word is an inferior program to the Windows versions in many other ways, motivation to accommodate this problem has been low.

Privacy/Security

Won’t people steal your Speech documents and keep all your cards?

Probably. First, we choose to have a more optimistic view of the debate community, and assume until proven otherwise that our opponents will ask us for cites rather than wholesale take our evidence. While I’m sure that some of the people we debate will lack scruples, I prefer to believe that this practice is not widespread.

More importantly, even if it happens frequently, it doesn’t confer much of a competitive advantage to the team who chooses to do so. In an era of massive caselists and prolific cite requests, access to any piece of evidence read in a debate is already a matter of a few minutes work. I think that this concern also buys into the myth that “not letting your opponents see your evidence” somehow confers an important strategic benefit. In reality, most debates are won because of superior technique, argumentative capabilities, or ethos – not because the other team didn’t have a chance to digest your evidence. I’ve frequently told my debaters that if we prepare effectively, we should be able to give the other team access to all our files for an hour prior to the debate and not have it affect our chances to win. Either way, I can’t isolate a single debate that any of my 7 teams lost this year due to “file stealing.”

What happens if I break a new (advantage/disad/etc...) and don’t want to give it to the other team before my speech?

There are several solutions to this problem. First, take care with labeling the block titles in your Speech document something less descriptive than the argument name, such as “New Advantage.” Combined with asking the other team not to “scroll ahead” since you’ll be breaking something new should go a long way towards restoring the strategic benefit of a few extra minutes of “surprise.” Secondly, you could save the new argument to a separate word document on the jump drive, and ask the opponent to only open that file once you reach that part of your speech. Finally, you could theoretically bring a paper copy of just your new argument.
Much like the concern about stolen Speech files, I think this falls under the category of “scarier in theory than in practice.” This ranked near the top of my debaters concerns before doing paperless – now, they no longer even bother with any of the aforementioned “solutions” (except perhaps changing block titles). It became clear to them that the extra minute in which the other team doesn’t know what your new advantage is just doesn’t have much tangible bearing on the debate.

Won’t people read ahead in the Speech document and gain a competitive advantage?

This concern is pretty much the same as the previous two – not as frequent or as big of a deal as people seem to expect. If anything, this works in the other direction – our experience has borne out that opponents who have tried to read ahead have been much more likely to stop flowing, miss arguments, or even waste speech time answering arguments which were never made.

Doesn’t sharing USB drives so widely present a virus risk?

Yes. Ensuring that each computer is up to date with current anti-virus and spyware software is an important element of ensuring paperless is as safe as possible. It’s also recommended that the USB drives used for paperless be used exclusively for that task – if they are kept clean and wiped before each debate (or at least before each tournament), the virus risk can be minimized. I’m fairly certain none of our computers has contracted a virus this year as a result of paperless – and in this age of widespread file sharing, other risks seem much higher.

Macro Problems

My macros aren’t working at all.

Most of the time, this is because your macro security settings are set too high. See the Installation section for more specific information on how to enable macros.

If this fails to fix the problem, ensure that the file you’re using was made using the correct template – especially when converting old files for use with paperless, it’s easy to forget to use the template and paste them into a regular Word document.

Another problem might be that Word has not correctly assigned hotkeys to each macro – select Tools – Customize – Keyboard (in Word 2003), and then make sure each Macro is assigned to a unique key combination. If this still fails to fix the problem, make sure that there are no other programs interfering with the use of the macro hotkeys, such as graphics software. Macs in particular should try using the Function key in addition to the hotkeys. Check the setting in System Preferences – Keyboard & Mouse to use the F keys as standard function keys.
You should also check to make sure that you have the paperless template installed in the Word templates folder – if you are using a file which doesn’t have the macros built in, it might be looking for the wrong file.

Lastly, if you’re using Word 2007, make sure that you’ve saved your files as Word 97-2003 Documents. A frequent error is to save them as Word 2007 (.docx) files, or even Word 97-2003 templates (.dot) files. This is especially important for Speech.doc – if it’s inadvertently saved as Speech.docx, for example, the paperless macros won’t be able to find it.

I was using a macro, and then got an error message. It says “Microsoft Visual Basic – Run Time Error 4198, Command Failed” and gives me the option to End, Debug, or Help.

Almost every way you can “break” the macros has been error trapped in the code, and you’ll get a message telling you what to do. When you get this message, it just means that you’ve done something the macro didn’t know how to handle. Fortunately, you haven’t hurt anything – it just means that whatever you just tried to do didn’t complete properly.

If you click “End” or press “E”, you’ll be returned to Word and can try and figure out what you did wrong. If you press “Debug” on accident, you’ll be sent into the Word VBA debugger, with a lot of cryptic looking code instead of your Word document. To exit this, just close the window and click “OK” when told this will stop the debugger.

Most likely, this is a known bug. Probably, you tried to move a block above the top of the document. If it was something else, please let me know if you can replicate it, and I’ll see what I can do to fix it.

I emailed a file to another team member, and the macros stopped working.

Some email programs or online mail services have been found to strip all macros from Word files when sending them as an attachment, presumably as a security “feature.” If you find this happening to you, try sending the Word document in a zip file, or with a temporarily modified file extension, such as File.dco instead of File.doc.
I pressed the macro hotkey, and my screen suddenly rotated 90 degrees.

This occurs on certain laptops using a particular graphics card software package. To get your screen back to normal, press Ctrl-Alt-↑. Then, right click on your desktop, select “Graphics Options – Hotkeys” and select “Disable hotkeys.”

I saved my document as Speech, but it still doesn’t work in Word 2007.

Odds are good you accidentally saved it as a Word 2007 .docx file. If your operating system is set to “Hide file extensions for known file types” this might not immediately be apparent. Try resaving as a Word 97-2003 file and see if it fixes the problem. It also might be helpful to set your operating system to always show file extensions, to more easily catch when something is saved incorrectly.

How do I change the macro hotkeys?

You can set Word to use any key combination you choose for each macro in lieu of the default hotkeys. Naveen’s template introduction linked near the beginning of this document provides directions on how to do this.

Why do the “Send” macros use page breaks, rather than block titles.

Two reasons – first, I think it gives a finer degree of control over how much information is sent to the Speech at a time. As referenced in the Assembling a Speech section, this enables the debater to send over multiple page shells, for example, instead of tediously clicking through an entire section, or wasting time highlighting multiple blocks. Secondly, VBA likes page breaks more than it likes Styles – programmatically, they’re easier to control. This is especially true in the instance where you’re trying to use files from multiple templates. Styles might alter between documents, but page breaks pretty much stay the same.

Move Up/Move Down don’t work correctly – I keep getting an error message, or it just seems to be moving page breaks around.

First, check to see if you’re attempting to use MoveUp on a block that is already at the top of the document. This is a known bug, and will give you an error message.

If not, then this is likely caused by improperly formatted blocks in your source documents. This arises most frequently when attempting to use backfiles which weren’t formatted in the paperless template. MoveUp and MoveDown can then sometimes have trouble identifying the proper start and end points for the blocks. It may also have to do with page breaks being in the wrong place, or even absent altogether.
Lastly, this can arise when the document in question contains blank pages used as headers for organization – known as “hats” or “fold-downs.” The macros are capable of using these with no difficulties as long as they are formatted as Heading 1 block titles, similar to the real block titles. However, strangely formatted Hats from other templates have a tendency to confuse it.

The solution is usually to check that page breaks are all in the right places, and then to just use cut and paste manually when all else fails.

**Pre-Round**

**How do we integrate backfiles produced in other templates?**

This is largely discussed above in the section on File Organization – the short version is that you should be able to adapt the macros to work in most templates. Either you can integrate the macros back into your old files so that you can send them as-is to a Speech document, or you can cut and paste files from the old templates into a new template document. This will likely give rise to some small aesthetic problems, such as incorrect font sizes – but in general, paperless functionality should remain intact.

**Isn’t highlighting files harder?**

To a certain degree, yes. Our template has a built-in highlighter function which is fairly rapid with practice. It is, however, slightly slower than doing it by hand. I think that this is made up for by the benefits of only needing to highlight any card once – it can then be copied to other files, or complete files can be shared by the team, saving duplicative highlighting efforts.

**In-Round**

**What about prepping before the block? There’s only one viewing computer, but both negative debaters might need to see evidence.**

In reality, this hasn’t presented much problem – most opponents have been happy with one viewing computer, as the 1NR has a sufficient amount of prep time during the 2NC to look at any evidence they need. They can also prep without looking at the cards for at least a few minutes – there’s usually analytics to write out, evidence to pull, etc...

For opponents who have insisted that having two computers is important, we’ve had two solutions, both of which have worked – either they have used their own computer as a second viewing laptop, or the 2AC has offered their computer as a stand-in until the 2NC is done prepping. Since the affirmative is usually backflowing until the 2NC is ready to speak, and the 2AC doesn’t have to give another speech which requires pulling evidence, their need for a laptop is minimal for those few minutes.
I just like paper – tactile feel and organization are important to me.

This would be my number one problem with paperless as a debater – there is admittedly no easy substitute for spreading out lots of files, or the feel of holding a stack of cards in hand. Partly, I think this is just an issue of familiarity and training – my younger students have a much easier time with this than my older debaters. It’s fairly clear from observation that my debaters are much more organized using the computer – the inability to “lose” cards during a debate is a boon.

How do I mark a card while giving a speech?

Most of my students just use their mouse to click at the part of the card where they stopped reading, then press Enter a few times to make clear where they stopped. If it’s only one or two cards, this can even be done after the speech. Remember, if you’re in reading view and you edit a card text, you’ll need to press Escape to re-enable the arrow keys.

My debaters have also taken to saying “marked at xxxxxx” when marking to let the other team know where they stopped if they’re following along. In practice, this system has presented no problems that I’m aware of.

What if I send over more cards than I’m going to read to my Speech document?

This is no different than giving the other team a block with multiple cards but only reading the top one. Some responsibility is on the other team to flow the speaker and pay attention to which cards are read – or to clarify in cross-ex if there’s any confusion. This also provides an incentive for the opposing team to follow along in real time, rather than skip ahead. The speaker can also opt to keep a file or two open on their computer to read a few extra cards should they have time – and then jump those last couple cards to the other team during cross-ex.

Reading View suddenly stopped working, what do I do?

Most likely, you’ve accidentally clicked inside a card, switching Word into edit mode. To re-enable navigation via the arrow keys, press Escape once.

I sent something to my Speech document, and now it looks weird (bigger font, out of order, etc...)

Keep in mind that both “Send” macros will send your selection to the current insert point in the Speech document. Odds are good that you accidentally had the cursor in a Block Title or other area of formatted text – and Word attempted to apply that formatting to everything you sent. Try pressing Undo and then resending the blocks to the bottom.
Doesn’t paperless make cite requests super easy?

Yes. There’s even a macro built in to our template. Put the cursor inside any card text and press Ctrl-Q. This will automatically reduce the card to just first and last sentences, ready to be sent as a cite.

I used MoveUp (or MoveDown) and now my view is in Normal/Draft view instead of my previous view (usually Page Layout).

The macro is set to do this – Normal view is better for working in the template. If this frustrates you, the macros could be modified to return to the view you had before using them.
Code Annotation

All of the paperless macros were coded by hand using Microsoft Visual Basic for Applications. This section includes the complete code for each of the five macros, annotated with commentary to better explain how they’re written. I make no claim that these are written as efficiently as they could be, nor that they are particularly elegant – I’m sure someone with more coding talent than me could offer suggestions on how to improve them. My hope is that this section can at least provide a starting point for teams which need to tweak the macros to make them work better with their preexisting template, to sort out bugs, or to add functionality in the future.

Lines which begin with ' are comments.

SendToSpeech

This macro sends any amount of highlighted text to an open Speech.doc file.

Sub SendToSpeech()

'Variable declaration section.
Dim CurrentDocument
Dim Doc

'Make sure Speech.doc is open – this is accomplished by cycling through each open document 'and checking the name against Speech.doc. This is where you would change the name if you 'wanted to use a different name than Speech.doc – such as Speech.docx to make it compatible 'with Word 2007. If Speech.doc isn’t open, it displays a message box reminding the user, then 'exits.

For Each Doc In Application.Documents
    If Doc.Name = "Speech.doc" Then Found = True
Next Doc
If Found <> True Then
    MsgBox "Speech.doc is not open"
    Exit Sub
End If

'Set CurrentDocument name, to remember which document you’re looking at
CurrentDocument = ActiveDocument.Name

'Copy selection
Selection.Copy

'Find and activate Speech document
Documents("Speech.doc").Activate

'Paste and insert hard return. Hard return is added to ensure that the next thing pasted in doesn’t
SendBlockToSpeech

This macro sends the current Block to Speech.doc, between manual page breaks.

Sub SendBlockToSpeech()

'Variable Declaration Section.  BlockStart and BlockEnd are used to remember the character
'position of the block being sent.  CurrentStart and CurrentEnd are used to return the user to the
'exact same place in the document after the copying process.  PageBreak is used as a flag to
determine whether to insert a page break after copying.
Dim BlockStart As Long
Dim BlockEnd As Long
Dim CurrentDocument
Dim CurrentStart
Dim CurrentEnd
Dim PageBreak

'Make sure Speech.doc is open – this is accomplished by cycling through each open document
'and checking the name against Speech.doc.  This is where you would change the name if you
'wanted to use a different name than Speech.doc – such as Speech.docx to make it compatible
'with Word 2007.  If Speech.doc isn’t open, it displays a message box reminding the user, then
'exits.
For Each Doc In Application.Documents
    If Doc.Name = "Speech.doc" Then Found = True
Next Doc
If Found <> True Then
    MsgBox "Speech.doc is not open"
    Exit Sub
End If

'Save Current Selection, so it can be reset at the end.
CurrentStart = Selection.Start
CurrentEnd = Selection.End

'Set CurrentDocument name, to remember which document you’re looking at
CurrentDocument = ActiveDocument.Name
'Clear Selection, to avoid conflicts with the following Find function
Selection.Collapse

'Find the start of the previous block. ^m is code for a Manual Page Break. If you wanted to
'modify the macro to define a “block” in some other fashion, this is where you would change it.
'.Forward is set to False to search backwards to the previous page break. .Wrap is set to
'wdFindStop to tell Word to stop after finding the first occurrence, rather than continuing to
'search the whole document.
Selection.Find.ClearFormatting
With Selection.Find
  .Text = "^m"
  .Replacement.Text = ""
  .Forward = False
  .Wrap = wdFindStop
  .Format = True
  .MatchCase = False
  .MatchWholeWord = False
  .MatchWildcards = False
  .MatchSoundsLike = False
  .MatchAllWordForms = False
End With
Selection.Find.Execute

'If no page break, set start to beginning of document. This assumes that the lack of a page break
'means you’re on the first block of a file.
If Selection.Find.Found = False Then
  BlockStart = ActiveDocument.Range.Start
Else
  Selection.MoveRight Unit:=wdCharacter, Count:=1
  BlockStart = Selection.Start
End If

'Find the start of the next block, in order to determine where the current block ends. The only
'difference with the previous Find function is that .Forward is now set to True.
Selection.Find.ClearFormatting
With Selection.Find
  .Text = "^m"
  .Replacement.Text = ""

Forward = True
Wrap = wdFindStop
Format = True
MatchCase = False
MatchWholeWord = False
MatchWildcards = False
MatchSoundsLike = False
MatchAllWordForms = False
End With
Selection.Find.Execute

'Check that the next block was found
If Selection.Find.Found = False Then

'If it wasn’t, set BlockEnd to the end of the document and set the page break flag. This assumes 'that the lack of a page break means you’re on the final block in the file. It also ensures that a 'page break gets added to the end of the file when being copied to Speech – this helps keep the 'file organized when sending over large numbers of blocks.
BlockEnd = ActiveDocument.Range.End
PageBreak = True

'If a page break was found, then move to the start of the next block and set the end position. The 'first line moves one character to the right, just like before.
Else
Selection.MoveRight Unit:=wdCharacter, Count:=1
BlockEnd = Selection.Start
End If

'Copy the block – sets the selection to equal the found BlockStart and BlockEnd, then copies it.
Selection.Start = BlockStart
Selection.End = BlockEnd
Selection.Copy

'Find and activate the Speech document
Documents("Speech.doc").Activate

'Paste block
Selection.Paste

'If copying the last page, insert a hard return and page break. This flag was set during the find 'process to ensure each block starts on its own page.
If PageBreak = True Then
Selection.TypeParagraph
Selection.InsertBreak Type:=wdPageBreak
End If
'Return focus and return to original selection
Documents(CurrentDocument).Activate
Selection.Start = CurrentStart
Selection.End = CurrentEnd

End Sub

DeleteBlock

This deletes the current block, between page breaks.

Sub DeleteBlock()

'Variable Declaration Section. ExtraDelete is used to ensure Word deletes the right number of 'characters, as explained below.
Dim BlockStart As Long
Dim BlockEnd As Long
Dim ExtraDelete

'Find the start of the previous block
Selection.Find.ClearFormatting
With Selection.Find
   .Text = "^m"
   .Replacement.Text = ""
   .Forward = False
   .Wrap = wdFindStop
   .Format = True
   .MatchCase = False
   .MatchWholeWord = False
   .MatchWildcards = False
   .MatchSoundsLike = False
   .MatchAllWordForms = False
End With
Selection.Find.Execute

'If at the top of the document, set start as beginning of document
If Selection.Find.Found = False Then
   BlockStart = ActiveDocument.Range.Start
   BlockEnd = BlockStart

'If not at the top, move the cursor to the start of the block title and set the start position. The first 'line of code moves the cursor one character to the right to ensure that it’s on the block title rather 'than the page break. It also sets ExtraDelete as True, to ensure that Word deletes the right 'number of characters – it has a tendency to leave one character of the block title if this isn’t 'done.
Else
    Selection.MoveRight Unit:=wdCharacter, Count:=1
    BlockStart = Selection.Start
    ExtraDelete = True
End If

'Find the start of the next block
Selection.Find.ClearFormatting
With Selection.Find
    .Text = "^m"
    .Replacement.Text = ""
    .Forward = True
    .Wrap = wdFindStop
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute

'Check a page break was found.
If Selection.Find.Found = False Then
    'If it wasn’t, set the end point to the end of the document
    BlockEnd = ActiveDocument.Range.End
Else
    'If it was move to the start of the next block and set the end point.
    Selection.MoveRight Unit:=wdCharacter, Count:=1
    BlockEnd = Selection.Start
End If

'If not deleting the first page, add an extra “delete” to eliminate the extra character.
If ExtraDelete = True Then
    Selection.Delete
End If
End Sub

**MoveUp**

This moves the current block up one slot in the document hierarchy. It makes use of the Word Outline view, which is based on the assigned Heading levels. Any errors are likely caused by problems in your document structure.

Sub MoveUp()

'Remember current location
Dim CurrentLocation
CurrentLocation = Selection.Start

'Collapse outline. This step switches the view to the Master Outline view, then sets it to only display Heading 1. This step is a prerequisite to the rest of the macro – failure to do this means Word will frequently move the wrong amount of text. One byproduct of this section of code is that the view is automatically set to Word’s Normal View on completion of the macro. If desired, you could add extra code to determine the current view and restore it.
ActiveWindow.ActivePane.View.Type = wdMasterView
ActiveWindow.View.ShowHeading 1
ActiveWindow.ActivePane.View.Type = wdNormalView

'Return to current location
Selection.Start = CurrentLocation

'Find the start of the previous block
Selection.Find.ClearFormatting
With Selection.Find
    .Text = "^m"
    .Replacement.Text = ""
    .Forward = False
    .Wrap = wdFindStop
    .Format = True
    .MatchCase = False
    .MatchWholeWord = False
    .MatchWildcards = False
    .MatchSoundsLike = False
    .MatchAllWordForms = False
End With
Selection.Find.Execute

'If at top of document, exit. This is the section of code that needs to be modified to fix the "moving above the top” bug referenced in the text.
If Selection.Find.Found = False Then Exit Sub

'Move the block up.  First, move the cursor one character to the right to place it on the Block
'Title, rather than the page break.  Then use Word’s Relocate function to move the section up.
Selection.MoveRight Unit:=wdCharacter, Count:=1
Selection.Range.Relocate wdRelocateUp

'Move cursor to the start of the block title for easier viewing
Selection.HomeKey Unit:=wdLine
Selection.MoveRight Unit:=wdCharacter, Count:=1

End Sub

**MoveDown**

This moves the current block down one slot in the document hierarchy.  It is structurally
identical to the MoveUp macro, so code annotations are not repeated.

Sub MoveDown()

'Remember current location
Dim CurrentLocation
CurrentLocation = Selection.Start

'Collapse outline
ActiveWindow.ActivePane.View.Type = wdMasterView
ActiveWindow.View.ShowHeading 1
ActiveWindow.ActivePane.View.Type = wdNormalView

'Return to current location
Selection.Start = CurrentLocation

'Search for bottom of page to error trap
Selection.Find.ClearFormatting
With Selection.Find
  .Text = "^m"
  .Replacement.Text = ""
  .Forward = True
  .Wrap = wdFindStop
  .Format = True
  .MatchCase = False
  .MatchWholeWord = False
  .MatchWildcards = False
  .MatchSoundsLike = False
  .MatchAllWordForms = False

End With
Selection.Find.Execute

'If at bottom of document, exit
If Selection.Find.Found = False Then Exit Sub

'Return to current location
Selection.Start = CurrentLocation
Selection.Collapse

'Find the start of the previous block
Selection.Find.ClearFormatting
With Selection.Find
  .Text = "^m"
  .Replacement.Text = ""
  .Forward = False
  .Wrap = wdFindStop
  .Format = True
  .MatchCase = False
  .MatchWholeWord = False
  .MatchWildcards = False
  .MatchSoundsLike = False
  .MatchAllWordForms = False
End With
Selection.Find.Execute

'If on first page, just move to the beginning of the document
If Selection.Find.Found = False Then
  Selection.Start = ActiveDocument.Range.Start
  Selection.Collapse
End If

'Move the block down
Selection.MoveRight Unit:=wdCharacter, Count:=1
Selection.Range.Relocate wdRelocateDown

'Move to start of block title
Selection.HomeKey Unit:=wdLine
Selection.MoveRight Unit:=wdCharacter, Count:=1

End Sub
Known Issues

Most of the macro code is error trapped to ensure stability and avoid annoying error messages – in general, it’s pretty hard to “break” the system. That said, there are a few known “bugs” or other computer related issues which potentially interfere.

- **MoveUp beyond top of screen.** When attempting to move the top block in a document higher than possible, i.e. beyond the first position in the document hierarchy, the macro will sometimes crash and give a debug error. This is easily avoided.

- **Moving the very bottom block in a document to the very top will combine two blocks between page breaks.** This occurs when the final block on a page doesn’t have a manual page break at the end, but simply stops at the end of the document. When moving this block to the top, it will leave the 2nd Block Title in the document stranded without a page break. For now, the easiest fix is to simply add one with Ctrl-Enter.

- **Problems with converting backfiles or combining files from multiple templates – these problems are referenced extensively in the text. Usually, there’s a simple workaround.**

- **Screen rotation when using hotkeys – known conflict with certain graphics card software – see the Common Concerns section for instructions on how to turn this off.**

- **Word 2008 support on Mac – Microsoft removed VBA support from this version of Word, seemingly for no reason – nothing can be done about this, so using an earlier version of Word is necessary.**

- **Conflicts with the Microsoft Language Bar.** This is perhaps the strangest behavior we’ve seen while using paperless. Some versions of Office come with the Microsoft Language Bar, which includes a rudimentary Speech-to-Text tool. This can be turned on inadvertently while using Word, which will then attempt to convert speech picked up by the computers built-in microphone into text in your open document. If you find random words and fragments of sentences appearing seemingly out of nowhere in your Word documents, try disabling the Language Bar. Somewhat embarrassingly, this took two days and four people to figure out.
Future Features/Advanced Suggestions

The bulk of the work on these macros was done in a couple hours a few weeks before our first tournament. Since time was of the essence, no attempt was made to make them perfect, or to include any advanced functionality – it just needed to work, and it needed to work immediately in order to leave time for drills/debugging etc...As we’ve debated with it over the course of the season, lots of ideas have come up for ways to improve the basic system, features to add, etc...This is a partial list – I’d be happy to get feedback on other ways people think this could be made better.

- More effective use of Outline View. One thing my debaters have started playing with is using Words Master Outline view to select which blocks they want to send to Speech. One particularly promising trick is the ability to select multiple blocks at a time using the Ctrl key. While you have to be careful about where you click to avoid inadvertently selecting two blocks at once, this enables you to send multiple blocks from different parts of the file all at once. It helps to click on the far left of the Outline view, near the “plus” sign. Even better, they will appear in the Speech document in the order they were selected, rather than the order in the original document.

- Programmable mouse integration. Since many higher-end mice come with multiple buttons, including the ability to assign each one to a keystroke, it’s possible to program a mouse to use all of the paperless macros using one hand.

- Use of AutoHotKey to speed desktop setup. AutoHotKey is a free program designed to quickly automate tasks in Windows, such as opening programs, reorganizing windows, etc...Essentially, it adds macro support for all Windows applications. Many possibilities exist, such as assigning one keystroke or desktop shortcut to automatically open a Speech doc and all your affirmative files, then organize them on the desktop in a standard layout.

- MoveUp/MoveDown macros for cards instead of blocks. One frequent request is the ability to move the order of cards within a block more easily. While this is much more complex than moving blocks, it should be possible to write a similar macro.

- Add .docx support. More extensive testing is necessary with the docx format, and it would require changing the macro code for the two Send macros.

- Auto cite request macro. Since a macro is already written to quickly convert cards into “cite” format, it should be simple to add a macro which automatically converts an entire Speech document or file to cites, to even more quickly facilitate intel requests.