



November 30 - December 3, 2004 ♦ MGM Grand Hotel Las Vegas

Creating a Web-Based CAD Manual

Matthew Stachoni (speaker)

CM23-1 Learn the ins and outs of moving your CAD Systems Manual to the online world. We'll discuss various web tools you can use to create a highly interactive, flexible and easily maintainable website. We'll cover organizing content, formulating topic features and templates, processing graphics and images, printing, and incorporating DWF™ and multimedia in your online manual. This class will benefit CAD managers responsible for company CAD standards who wish to publish their work online. Attendees should understand the basics of HTML, image and multimedia formats, and generally how websites work.

Who Should Attend:

CAD managers and others responsible for creating and publishing company CAD standards

Topics Covered:

- Benefits, expectations, and realities of moving to an online CAD manual
- Planning and managing the project; developing a basic workflow task list
- Skills you need and skills you do not need
- Software requirements and evaluating Help development systems
- HTML Help vs. Web-based Help
- Introduction to RoboHelp: using dedicated Help applications to create a single source of content
- Creating Topic templates for your content
- Using Conditional Text to tailor output for different purposes from the same source material
- Taking effective screenshots, image processing and workflow.
- Incorporating DWF files
- Creating “active content” such as onscreen demonstrations to more easily explain concepts

About the Speaker:

Matt is the CAD and IT Manager for Bernardon Haber Holloway Architects PC and their affiliate firm, BHH|Tait Interiors LLC. He oversees the CAD and IT operations for both companies at their three offices located in Pennsylvania and Delaware. He has worked as a CAD and IT systems manager for several architecture firms for over 13 years, and was also a consultant to the Massachusetts General Hospital's Planning Department. Matt has been using, customizing, and programming AutoCAD® and related products since 1987, is active on the Autodesk newsgroups, and has attended Autodesk University® since 1998.

Email: mstachoni@comcast.net
mstachoni@bhhtait.com

Introduction

As a CAD Manager you've probably already answered the easy question, "Do I need to put my CAD Manual online?" But now you need to answer the hard questions, like, "What steps do I need to take?" "How much time is involved?" "What software tools are required?" "What's the process?" and most importantly, "Where do I start?"

Before you can answer any of these, you need to determine what kind of CAD Manual you wish to produce, because the Web allows you to move from a stale, static set of pages to an entirely dynamic learning experience for your users. However, with such dynamics comes increased complexity, and you need to understand the tools and techniques required to fully publish the kind of Manual you want. The good news is that it's *not* as hard as you probably believe.

There are basically three different types of CAD Manuals, in order of increasing complexity:

1. **A reference guide to CAD standards**

Reference-only CAD Manuals are by nature minimalist documents, where content is restricted to "just the facts." It will include only the minimum information needed to set Standards in place, such as filing standards, layers, text and dimension styles, color-pen styles, and plotting. There isn't any content that discusses technique or standardized production practices in depth. While a Reference Guide could easily live out its days as a print-only document, it can certainly take advantage of Web-based features and benefits, like enhanced navigation and search capabilities.

2. **A "Best Practices" guide**

In addition to CAD Standards, a "Best Practices" type of manual includes technical step-by-step instructions for production. This type of manual is best suited to those organizations that have standardized their CAD procedures and customization to support those procedures, where a simple Standards Manual would fall short. These type of "How To" guides have much more content, and thus structure, than a simple Standards-only manual. Thus, publishing a Best Practices guide online is more involved, as you need to include more organization and think about the user's experience. This type of Manual pushes you to include more advanced Web-only features, such as an active table of contents, links to related topics, popup windows to explain terms, drop down text, and keywords and indexes for performing easy searches.

3. **E-Learning tool as an adjunct to in-house training**

Internet-based content offers a full range of multimedia capabilities that you can exploit for instructional purposes. An online format can leverage technologies like DWF and video, lowering the learning curve for all employees. Such an "e-learning" CAD Manual may offset the need for much formal instruction, which is reason enough for some to make the investment in time and money. With the glut of very inexpensive screen capture utilities available, it can make a lot of sense to replace the pages of text needed to describe a procedure with a one minute movie. An e-learning manual is also great for people who don't use CAD on a daily basis, such as Project Managers who just need to know how to plot. If your organization is large and/or geographically distributed, you may need something that can serve as a training tool for remote users where dedicated one-on-one training is much more expensive. Finally, For Value Added Resellers (VARs) and consultants who specialize in providing CAD related services, being able to deliver a highly polished multimedia based CAD Manual, tailored to an organization's processes, can be a lucrative part of your business.

Start Small; Grow Over Time

The level to which you take your CAD Manual is really up to you, and is dictated by many factors. You can start off easily enough, importing your existing content into a set of HTML pages linked together with navigation, which gets people used to reading the online system in lieu of print. Then you may want to include more online-only features, such as Dynamic HTML (DHTML) effects like popup windows and drop down text, DWFs and instructional videos. For better or worse, the wonderful thing about CAD Manuals is that they are never quite "finished."

Regardless of the size or scope of your CAD Manual, your work will need to be able to mature over time in an easy fashion. If you keep yourself organized using the kind of tools and techniques discussed herein, you will find it simple to add, modify and publish your content, and have it up to date for all of your users. While it's easy to see why publishing an HTML-based CAD Manual makes sense, getting started is usually the hardest part.

Online Help System Requirements and Features

When creating an online CAD Manual (no matter what kind), you are primarily creating an Online Help system much like the Help provided with AutoCAD. As such, your documentation can now provide things that make an HTML-based system a uniquely competent mechanism for delivering such content:

- **A Dynamic Table of Contents (TOC).** A Help System is most effective when it is accompanied by an active and easy to use Table of Contents. Each entry in the TOC links to a Topic, which is an HTML page of content on a particular subject. In contrast to the static TOCs in printed manuals, online documentation allows the TOC to follow the user as they wind their way through the Topics, even through hyperlinks. This creates a context where the user always knows where they are in the vast sea of content they are presented with.
- **Links to internal and external sources of related material.** Providing links throughout your CAD Manual breathes life into it and ties it all together. Jumping from one page to another is certainly useful, but what's more effective is linking to material that *relates* to your current topic. Your links to related material can be either internal to the CAD Manual itself or to external web pages, and they can be anything – an HTML page, a popup window that defines a term, a DWF of an example drawing, or an AVI movie demonstrating technique. In contrast to normal Web sites on the Internet, it's most effective in a Help context to launch this related material in an external window, keeping the original page available. When the user is done with the additional material, they can close it and easily return to the topic. This eliminates unnecessarily hitting the Back button.
- **Search and Index abilities.** People have quickly become accustomed to routinely searching for online content using keywords, so much so that “Google” is considered a verb. Even in a paper document, the Index was a very important section. Complex things like CAD systems require the ability find disparate information quickly and lend themselves to incorporating search features. CAD users faced with finding information in a lengthy online manual will want thorough Search capabilities immediately.
- **Flexibility in Content Change.** We all understand that a typical CAD Manual is obsolete the minute we hit the Print button. They are very much “living” documents, and as your particular CAD infrastructure matures, new content will constantly need to be added and stale content discarded. It's easy to see why printed documentation is on the way out, as traditional printing costs have skyrocketed while publishing over the Web is essentially free - and will always remain so. It must be easy and inexpensive to publish your changes without killing large numbers of trees.
- **Flexibility in Output.** The ability for your Help System to output different formats and content from a single managed source is essential. Over time you will want to leverage Web and multimedia technologies in the online version, yet also be able to output a PDF which includes print analogues of the online material. As we will see, you can use something called Conditional Text to distill parts of your content out to different formats.
- **Flexibility in Design and Presentation.** HTML provides a mechanism called Cascading Style Sheets (CSS) to separate Presentation from Content. Presentation is concerned with the overall look and feel of your online manual – the graphical attributes of text, tables, and so on. Using CSS is very much like working with Microsoft Word's Styles and Formatting feature. In HTML, these styles are defined in external .CSS files that are then applied to *structural content* (headers, footers, tables, paragraphs, etc.) defined by codes in the HTML content. By modifying the styles externally, you can redesign the look of your CAD Manual without modifying the actual content at all.
- **Active Content.** A printed manual offers two choices for content type: text and graphics. HTML allows for embedding more “active” content for learning. In addition to the usual hyperlinks to different pages, all browsers support Dynamic HTML (DHTML) and Frames, which is used to create navigation panes, animated text effects, popup windows, and inline glossaries.
With browser-based Help it's possible to include AutoCAD drawings, through publishing to DWF and the DWF Viewer application. This installs an ActiveX Control that allows for the viewing of DWFs embedded inside of your Topics in Internet Explorer (IE). For non-IE browsers, you can also link to DWFs normally, which will launch the full DWF Viewer. Used appropriately, DWF can be an important e-learning tool as it includes layer and lineweight information. More advanced technologies like Flash animation and linked .AVI movies, complete with sound, take your CAD manuals to a new level of instruction and long-term usability.

An Online CAD Manual Reality Check

Many CAD Managers view an online CAD Manual as something of a “Holy Grail” of sorts. Anyone who has written a paper CAD Manual can easily imagine migrating to a fantastic web site full of every piece of information any user could every want to know. However, moving from traditional print methods to the online world presents a whole host of new challenges, and the results may not quite meet expectations. In this section we will try to dispel some of the more common myths and misconceptions of what the process and finished product will bring to the table.

Myth #1: Because the CAD Manual is online, everyone will read it and follow it to the letter.

Reality: Ha! Of course, CAD Users will continue to ask you questions and/or ignore the Standards as written no matter if the CAD manual is a printed document, a Web site, or indelibly tattooed on the backs of their eyelids. It is up to you, as a Technical Writer, to make the content accessible and easy to read, in both print and online. Putting more active content online – especially in screenshots and animated tutorials – may help to cut down on the phone calls, but it will not eliminate them. In this context, you should view creating an online version as way of making it easier to revise and publish material over traditional print-centric methods.

Myth #2: Writing an Online Manual means I’ll never have to worry about a print version.

Reality: While it would be great to never have to deal with mangling Word documents again, users and managers alike will – at least initially - expect there to be a print version of your manual, or at least a mechanism to print out an individual Topic. People have a common tendency to want things on paper, so they can write in their own notes and tape them up next to their monitors. Some simply have an aversion to reading electronic-based media because it may not be as comfortable. With Single Source documentation (discussed later), you can take into consideration the content and output print-specific versions to Word or PDF, but as we’ll see it is additional overhead and needs to be taken into consideration.

Myth #3: My current CAD Manual is a .DOC file. I think I’ll just use Word’s “Save as Web Page” feature.

Reality: Stop. Even if you have tons of Word documents (or even one big Word doc), putting it online inside a cohesive structure is a not the same as doing a “Save as Web Page” operation. Think of this as an important public safety tip from me to you:

**For the love of everything good and right with the world,
do not use Microsoft Word’s “Save as Web Page” feature as a way to make a Web page.**

First, depending on your version of Word and the options you set, you’ll either get a single “Web archive” file (that can only be read in IE) and/or some extremely **bad, proprietary** HTML/XML output. It will not look exactly like what you had in Word, is almost completely uneditable, and will not comply with the World Wide Web Consortium’s (W3C) standards. In other words, it will be a complete mess. In fact, MS Word’s Save as HTML functionality is so horribly broken that Macromedia’s Dreamweaver (a leading Web site development application) has an explicit “Remove Word HTML” feature.

Second, and most important, you won’t get any sort of site navigation, one of the most compelling features of a Web-based document. Remember, your end product is not just a single document that can be viewed online. You are creating a complete Web Site with what can amount to tens or hundreds of separate but inextricably linked HTML pages, images and multimedia files, managed under the umbrella of some sort of navigational control. Content goes hand in hand with Structure and Navigation, so outputting a single Word document to an HTML file really doesn’t do much for you. Some Help applications will allow you to directly import your existing Word document(s) as source data for later editing, but *please* don’t do the conversion from Word itself.

Myth #5: I’ll need to set up a Web server.

Reality: Any web site can be hosted by any PC without dedicated web server software installed, as long as it is normally accessible over the corporate LAN. All you need to do is store it on a network server someplace and provide a link to the first topic’s HTML file. You only need a Web server if you wish to make the site accessible to remote users over the Internet.

Myth #5: Structure and navigation aren't important.

Reality: In Web design, navigation is always king. In a CAD Manual, the structure of your navigation is arguably just as important as the actual content. If people cannot find what they want fast, they will not use the online version no matter how well the topics are written. Writing content is comparatively easy, but navigation and structure is much harder to accomplish. The Table of Contents (TOC) needs to be thoroughly thought out, organized in a logical fashion, and best tested by users during development.

In some cases this is very difficult because the desired content is so interrelated. That's where hyperlinks, indexing and keywords come into play, and are an essential part of the online documentation process. Without the ability for users to search for topics via keywords, they may as well be reading from a paper bound manual.

Myth #6: HTML is hard. I don't know JavaScript programming or DHTML. I'll never be able to do this.

Reality: You actually don't need to know much about HTML to create an effective online manual. Using a dedicated Help application that can import existing material (e.g., Word .DOC) and convert it into solid HTML and allow you to add content in an easy, intuitive way is the key to success.

Myth #7: This will be easy. I can blast this sucker out over a weekend.

Reality: This one isn't even close. Although the process of putting together an online manual is not *that* hard, it is time intensive. Much of your time will be spent experimenting with HTML and becoming proficient at solving HTML related problems, which you *will* incur even with the best Help application. In addition, you most likely have to learn some new software and its idiosyncrasies to complete the task. Combined with the large amount of information you have to classify and structure, pulling everything together to create a unified web site is a long process that undergoes revision after revision. Even with all of the time-saving tools at your disposal, expect to spend anywhere from two weeks to 3 months on the project, at 40 hours a week, depending on your requirements and expectations.

Myth #8: Making a PDF of my Word document is the same as building an online manual, right?

Reality: The PDF option is more and more prevalent these days, as everyone has the Adobe Acrobat Reader. Output from a word processor to PDF is rather easy, using either the full version of Acrobat or a third party printer driver. Formatted correctly, you can output a large document (or set of document files) to a single PDF and get a navigation pane of bookmarks. However, PDF suffers from being bound to the print medium from which it came. You cannot include more active content like ActiveX Controls (i.e., DWF), embedded videos, and DHTML effects. Also, Acrobat's ability to render nested bookmarks that remain easily readable is rather problematic. Note that with certain dedicated Help applications it is possible to output to PDF as a Printed Documentation option from your source files, just as output to a Word .DOC is possible. I personally view PDF as an output option in addition to a Web site. It's possible, for instance, to output each topic as a separate PDF (tailored for the print medium), and include a link to print that PDF from the online Web page. This is an effective means of getting around the whole print question.

Basic Task List

1. Create a folder on your system for your project. If you are using a dedicated Help package, create your project someplace that is easy to get to. Build additional folders inside the project at the same folder level as the project for holding assets like screen shots, DWFs and movie files. Depending on the Help development application you use, it may be more efficient to work locally, then save your work to the server.
2. Download and start evaluating Help software. Conduct dry runs on demo versions. Develop a very simple initial manual with a few topics, chapters, a cover page, index and keywords. Most allow you to import Word documents, so try this method on a few packages to see how formatting and styles convert in the process. Don't be afraid of trying out several applications before settling on one and *read their manuals carefully*.
3. Create or modify a complete production project with DWG files that represent a standard example of what users should work toward. You would then relate your documentation of CAD Standards and Practices to drawings in this project. Convert the sheet set to DWFs and link to it from your topics. Making your CAD Manual based on a complete project keeps things in perspective for users, eliminates confusion over acceptable standards, and can be used as a general reference point for production.
4. Start thinking of an overall organizational structure by formulating some initial topics, without getting bogged down in details too early. Use index cards or sticky notes to write down ideas for topics, e.g. "Plotting a full

Creating a Web-Based CAD Manual

sized sheet,” “Importing a Layout from a Template,” and so on. Ask your users for specific topics they want to read about.

5. Look for topic sequences, where one topic naturally leads to the next topic to give it some flow. Group related topics together into chapters or sections. A good methodology is to work loosely. Avoid a convoluted navigational structure that puts important information too deep inside of a single chapter, and steer clear of the temptation to nest many chapters within chapters – break them out into more top level chapters. You will probably find yourself fighting to put the most important content up front. Try offering a “Chapter in a Nutshell” topic. There is no magic rule to follow for organizing content, so testing content flow for findability is time consuming but crucial.
6. Sketch out a basic topic page for your content, including a Header and Footer. Answer some initial functional design questions common across all topics: Does your company logo want to go on the page? Should you include a copyright notice? A “print” button? An email link for feedback/help? A “Related Topics” link?
7. Look for design ideas from existing web sites and from Help (.chm) files on your hard disk. Look at CSS files for technical details about how something is done. Start creating prototypes for your own CSS styles as you write content. Search for common stylistic features you want to include in your pages, such as shaded topic titles, font assignments, horizontal rules and padding to break up pages or to set off text. Test styles on some dummy “Lorem Ipsum” text to fill out a page (check out <http://www.lipsum.com/> to generate some).
8. Once you settle on an application, start writing the body for your topics in a skeleton fashion. This allows you to see where the status of each topic is. During this process write inline notes to yourself, which can be set off using a dedicated “annotation” style like **bold red**. Think about where screen shots, expanded text boxes, DWFs or other media will logically want go, and mark a space for it and give it a caption. That way, you can write the meat of the content first, then go back and take the screen shots later. I do it like this:

[shot of XYZ Tool Palette]

XYZ Tool Palette

9. Come up with a workflow for producing screenshots. While you are working on a screenshot, always remember to keep a copy of the “raw” .BMP image for safety in case you mangle it with Photoshop.
10. Understand your intended audience’s browser capabilities. Does everyone use Internet Explorer? If so, maybe you can get by with embedding ActiveX controls inline, such as the DWF viewer. If they use another browser that doesn’t support ActiveX controls, such as Mozilla’s Firefox, you’ll need to work in a friendlier manner that plays to a common denominator. Browser compatibility is a much bigger issue than it was in years past, as IE now has some real competition in the marketplace. Some of the assumptions you could safely make in the past may not apply.
11. Evaluate the capabilities of your final format(s). Will the output be solely limited to a Web site? Do you need to have a printable version of the entire manual, or do you want to have printed articles for each topic? Start building Conditional Text rules for classifying specific content to a target layout. Develop images that are more appropriate for Print (i.e., at a higher resolution) than the Web. Understand that print and the Web are very different means of communication; as a general rule they do not mix very well without taking your topic layout and image resolution into account.
12. If desired, start creating AVI movies of instructional material. If you need to output a print version, create an adjacent text based set of instructions.
13. Decide on distribution mechanisms. You do not need a real Web server for your CAD Manual, because you just need to link to the first topic or main *index.html* file on your network. However, you may want to create a customized Help menu in AutoCAD for your users that launches the welcome page, or create a desktop shortcut that people can access.

Managing your Project

Creating a CAD Manual, either online or not, requires some basic Project Management skills to be successful. Simply locking yourself in a room and cranking out the product is NOT the way to go (trust me on this). Get a plan together and implement things in small, manageable steps. There are two points that you should follow to make this easier on yourself and your company:

Treat your boss like a client, and the CAD Manual as any other project with a budget and timetable.

Come to an agreement with management (the client) as to the amount of work that needs to be done before it can be considered presentable. Make him/her/them understand that it will most likely never be as complete as everyone wants, and work out a schedule for delivering acceptable chunks of content and design. Most importantly, make sure you are able to get the dedicated time to do so without getting pulled away for other projects. It's important that you have as much backing as possible to stay focused on the task.

Work with your users at all times.

Create dry runs of your proposed navigation and layout for feedback in the early stages. Even if you are producing a simple Reference Manual, ask them for feedback on content, and get a wish-list on topics that they would want to see written in the manual. Remember that users and managers alike will see the CAD Manual as a Bible of sorts, because they tend to view anything that gets written down to be more authoritative than something informally passed along via word of mouth.

The benefits of having a plan

Regardless of the scope of your CAD Manual, you will gain several benefits by managing the task as a real project, and presenting each of the pieces to your ~~coworkers~~ clients as you complete them:

1. You are forced to analyze the immediate work ahead of you, and be able to design each milestone as a series of discrete steps that are easily reached. It's tempting to keep looking at the big picture, but that can immediately get frustrating and intimidating. By initially mocking up your topic structure based on major topics, and thus idealizing your table of contents, before writing much of the actual content, you can pre-visualize your manual and easily see the numerous smaller tasks ahead of you.
2. You can get feedback on structure early that could affect the final outcome. It's a horrible experience to write a ton of content and structure it the way *you* think it should work, and then have criticisms heaped on because you did not include things that people want, or (worse yet) included topics that are deemed trivial or not important.
3. Small issues that may become huge issues can be better identified earlier in the process. For example, if you charge through your writing process without first creating CSS styles up front, you may end up with a mish-mash of inline font and paragraph formatting that would be a nightmare to fix to make consistent across the whole project.
4. Before you begin, no one will appreciate the tremendous amount of work you have to do. However, if you present a plan and stick to it, your clients/managers/staff will respect your honesty and organizational skills in dealing with such a high workload. Don't give a deadline for an unknown amount of work, and then have to pull all-nighters to get the project done. Leave yourself time to initially plan the project.

Skill sets required

Developing an online CAD Manual requires, at a minimum, some basic knowledge of HTML syntax and the ability to make sense of the source code of a Web page. However, there are skills required above and beyond this. It's important to also understand what you do NOT have to know before you commit to the project.

Skills you need, rated from 1 (low skill level required) to 10 (high skill level required):

- Technical writing ability (10)
- Organizational skills (9)
- Understand your audience and the technical details of their browser of choice (10).
- The ability to learn and effectively use unfamiliar software applications (7)
- Digital art / bitmap editing skills (6)
- Understanding of basic HTML syntax, including the ability to read nested tables to make quick fixes (6)
- Cascading Style Sheets (CSS) (5)

Skills you probably do not need:

- Advanced web page layout skills based on tables or framesets.
- JavaScript programming skills

Note: a great resource on basic to advanced HTML is found at <http://www.htmldog.com>

Software tools you will need¹

I've compiled a small list of some of the applications that can help you realize your goal.

Help Development Applications. This is the biggie, the one you will most definitely need, just as CAD software is needed for modern drafting. All include a WYSIWYG HTML editor.

- RoboHelp X5: www.ehelp.com
- Deva Tools for Dreamweaver: <http://www.devahelp.com/devatools.htm>
- Doc-to-Help: <http://www.softexport.com>
- Help and Manual: <http://www.ec-software.com>
- Webworks Publisher <http://www.quadralay.com>

HTML Editor. This may be considered an "extra" based on the capabilities of your main Help development application, but it's very useful as a complimentary tool.

- Macromedia Dreamweaver MX 2004: <http://www.macromedia.com/software/dreamweaver>
- Microsoft FrontPage: <http://office.microsoft.com/home/office.aspx?assetid=FX01085802>

Screen capture program. To make screen shots easy you can get a dedicated screen capture utility, one that will grab not only the screen but any menus or portions of menus you require. Luckily, these are pretty cheap.

- Snagit: <http://www.techsmith.com/products/snagit/default.asp>
- Screen Hunter: <http://www.wisdom-soft.com/products/screenhunter.htm>

Bitmap editing and image viewing software. These are absolutely essential, and in many cases can be purchased in a bundle with other graphics applications, e.g. the Adobe Creative Suite.

- Photoshop: <http://www.adobe.com/products/photoshop/main.html>
- Paint Shop Pro: <http://www.jasc.com/products/paintshoppro> (now owned by Corel)
- Corel Photo-Paint (part of the Corel Draw suite): <http://www.corel.com>
- ACDSee: <http://www.acdsystems.com>
- IrfanView: <http://www.irfanview.com>

Screen recording software. These applications allow you to record on-screen demonstrations, which you can annotate with a voice-over.

- Camtasia Studio: http://www.techsmith.com/products/studio/tour_recorder.asp
- RoboDemo: <http://www.ehelp.com/products/robodemo/>

DWF Creation and Viewing. All versions of AutoCAD from 2002 on have the ability to publish DWF files. 2004/5 ships with DWF Viewer 4.1, which gives you the ability to view DWF files. Earlier versions would require you to download and install the DWF Viewer separately, which registers the ActiveX Control required to embed DWFs into your HTML pages.

- Autodesk DWF Viewer: <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=2787358>

Evaluating Help Development Software

Part and parcel of the online CAD Manual creation process is evaluating the software you will need to put it together. **You will need specialized software for this task, and some of it is not cheap.** As discussed previously, you will not be able to output a functional and useful online CAD Manual with all the "goodies," using MS Word alone. Things to look for in a Help Development application are:

1. Fully functional demonstration versions of the software that are active for more than 15 days
2. Navigation controls in a sidebar or frame
3. Support for configurable Conditional Text (Print, Online, in development, etc.)
4. Multiple output formats (Web Help, HTML Help, Word, PDF)
5. Support for Indexing and Keywords
6. Keep track of rearranged content (database-driven content management system)

¹ Disclaimer: The author is not affiliated with and in no way endorses any of the software applications discussed in this class. All applications used in class are for demonstration purposes only.

7. Related tools, such as an HTML Help inspector
8. Overall ease of use, with few if any "gotchas"
9. Price

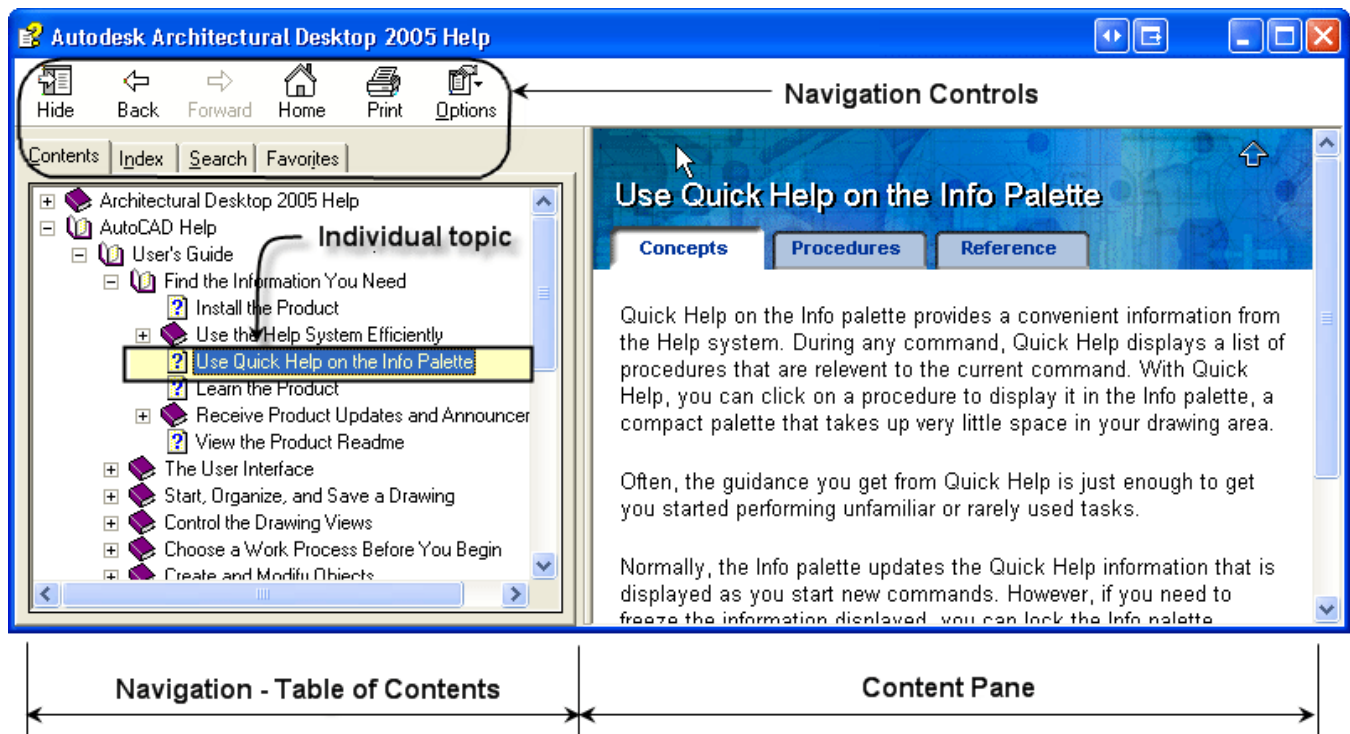
A decent comparison is found here: <http://www.helpstuff.com/downloads/toolsv5.pdf>

You will need to evaluate how easy the application helps you complete your tasks. In particular, you want to look for weaknesses and situations that would force you to go through every .html file to do something. For example, does the application include templates that can be redesigned in one place and automatically reapplied to all existing topics?

Obviously, quality of output in all areas is a big factor. If you are looking towards Web-site output only, you need to make sure your output has the look, feel, speed and polish you are pushing for. Does it track changes and progress? Does it only republish those items that have changed? Does it have any multiple-authoring capabilities built in?

Types of Online Help: HTML Help

If you are used to reading online .CHM Help files bundled in software (e.g., AutoCAD's online Help), what you are seeing is everything that makes up a web site - HTML files, linked images, css and JavaScript (.js) files - all compiled into a single file, presented through an ActiveX-based HTML Help control system. This "wrapper" application is built on Internet Explorer (IE), and provides things like a navigation pane, a search edit box, a print button and other features. Because many vendors provide Help through the HTML Help mechanism, and because it uses Internet Explorer as the base engine, you have to have IE installed as a prerequisite for many applications.

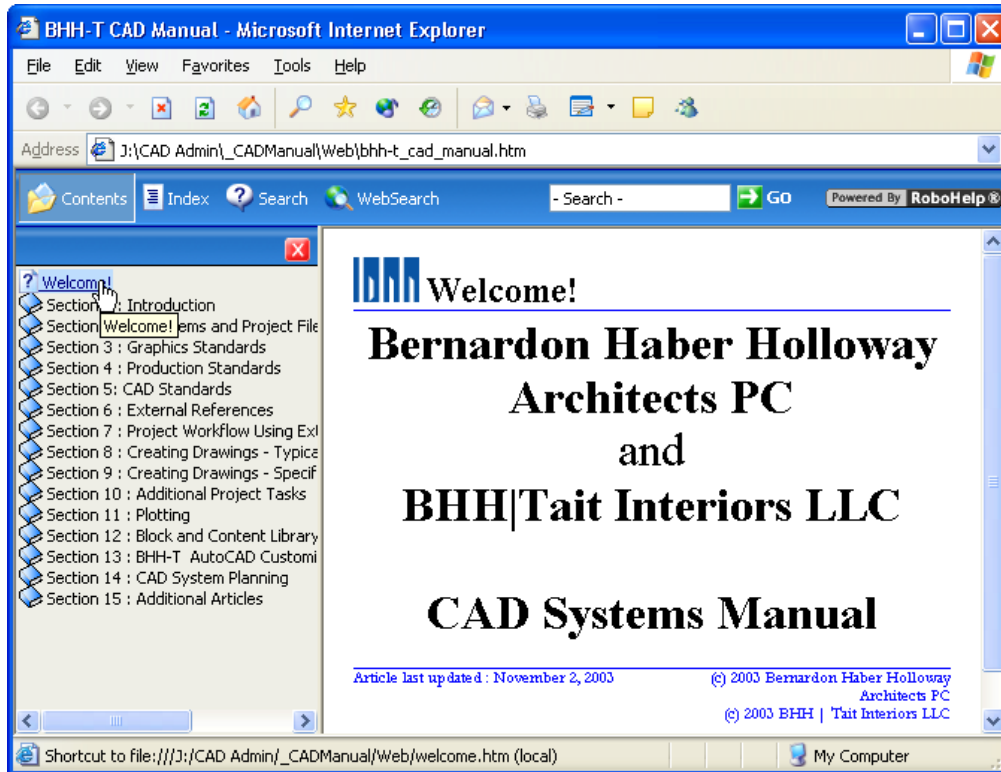


Autodesk Architectural Desktop 2005's online Help, with a Navigation pane, content pane, with Navigation controls

To create HTML Help, you can use a variety of third party specialized Help applications, from free to expensive in cost. But because HTML Help is a Windows-only specification (using the IE engine), many Help apps will transpose your content to build a regular Web site, so online help is available in operating systems like Linux. Our focus is in creating a Web Site of material, not a single file that would be distributed to users, so it's important to be able to render as many of these features as we can in HTML.

Web Help

Web Help will mimic these controls using JavaScript and Dynamic HTML (DHTML), in most cases to maximize cross-browser compatibility. Because of this, the output look and feel is closely tied into the abilities in the development application itself. Some applications can generate very slick looking Web sites that function very similarly to an HTML Help file. The following is an example of RoboHelp's output:



A sample corporate online CAD Manual created using RoboHelp

Other examples, like in the online Help for Photoshop CS are, perhaps, not so slick:

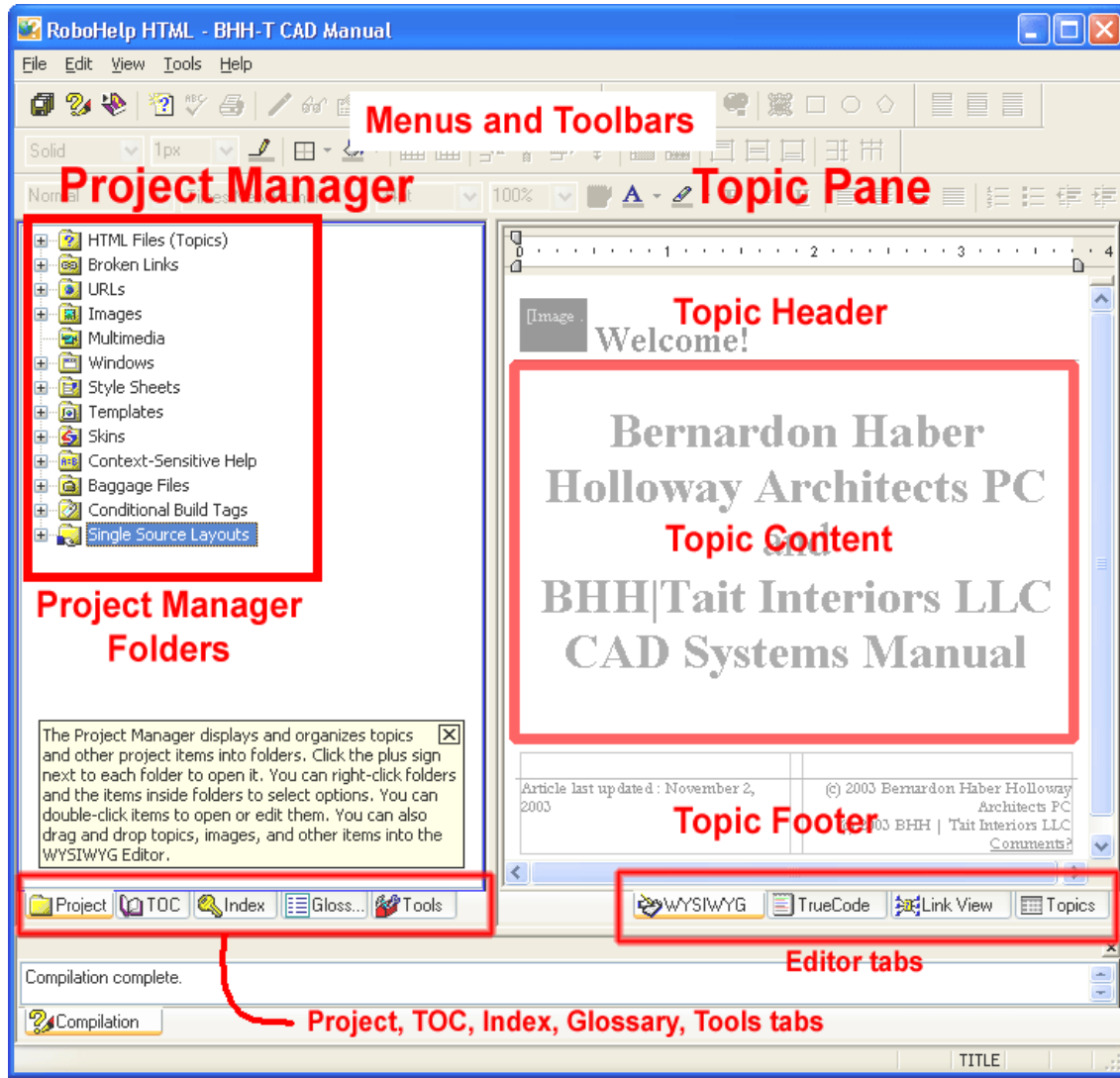


Photoshop CS Web-based Help Manual

Introduction to Help Development using RoboHelp

As an example of using Help development applications, we will be looking at RoboHelp's user interface to demonstrate a basic "Single Source" workflow.

Here is RoboHelp's main screen, with the important components highlighted:

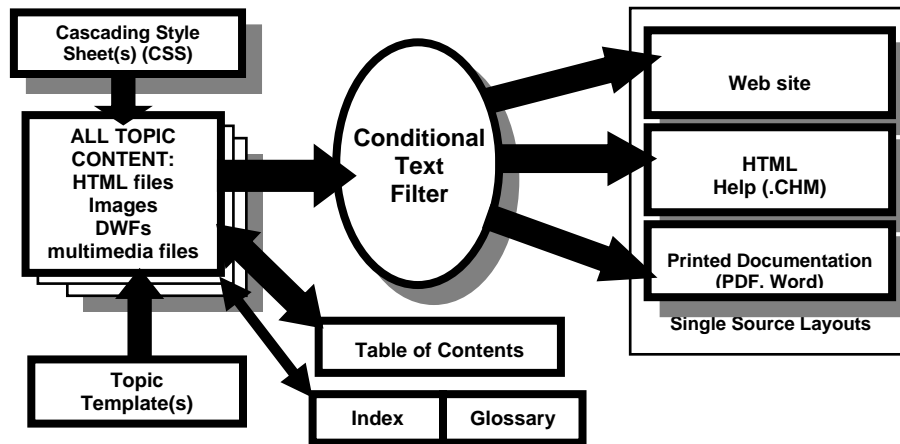


RoboHelp's main interface screen

The main area of the application is split up into two panes. On the left side is the Project Manager. The Project Manager Folders contain all of the content for your project – Topics (HTML content files), Templates, CSS Style Sheets, Skins, URLs, Multimedia files, and so on. The tabs underneath determines the Project Pane's view, showing your Table of Contents, Index, Keywords, Glossary and Tools.

The right side is the Content pane, where you design and input your topics in a WYSIWYG editor. You can choose to display the True Code (raw HTML), links, and the list of topics.

Working with RoboHelp



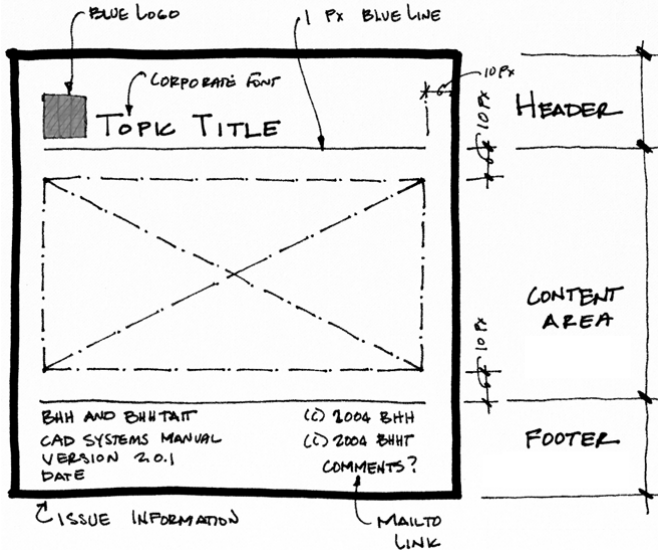
Many of the workflow tasks using RoboHelp are applicable to other applications as well. A typical RoboHelp project would proceed like this:

1. Create your Web Help project using the New Project Wizard in RoboHelp. You will define your project using a primary Single Source Layout output format (Web Help, HTML Help, PDF, etc). RoboHelp will create a folder hierarchy automatically as you create project folders in the program. Tips: name your project using a Version number for the folder, which allows you to create successive versions more easily. Also, set up your project locally on your PC, and back up to a server often. Because it is a database application, RoboHelp does not really respond well to network overhead and slowdowns.
2. In Explorer, create some adjacent folders at the same level of the main project's folder which will house your "working" area for additional content like images, movies, music, etc. that you will generate outside of RoboHelp.
3. Create a single topic you can use for design and testing, and write up some dummy content.
4. Create a Style Sheet for your project, which defines how the topics are to look, either by using one of the default files or creating your own from scratch. Style Sheets in RoboHelp are true Cascading Style Sheets (.css files), which you assign to topics. Create some simple styles that will be applied to structural elements, like Normal, Heading 1...3, Hyperlinks, Numbered lists, and so on. Once defined, a style is available in the Formatting toolbar and can be assigned dynamically to selected content.
5. Create a standard Topic Template that you'd like Web pages to follow. Implement any design ideas you get for header and footer areas. You can insert special Fields that update automatically, like the Date or Topic Title. You can create as many templates as you like, enabling you to experiment easily.
6. Once the initial design tasks are complete. Use the Topic Composer (TOC tab) to create new "pages" (topics) and "books" (chapters). Assign the Template and CSS files to your Topics through the Topic Properties. Start to experiment with topic organization and testing navigational structural flow.
7. In the Project Manager's Project pane, create additional folders under HTML Files to logically group your topics. Note that this folder structure can be defined first and a TOC can be created automatically from this folder structure, but that required a certain amount of overhead in topic and folder design which may not be appropriate.
8. Create some Conditional Text rules for Print and Online content.
9. Develop one or more Single Source Layouts - HTML Help, Web Help, Printed Documentation, etc.
10. For exporting into Word, develop a DOT template that translates your web-friendly CSS styles into print-friendly styles. Note for Printed Documentation layouts (.doc) you can define a different TOC.
11. Create screen shots, DWFs, and AVIs, and insert them into your Topics. Put them in the same folders as the topics for maximum portability so the links are relative.
12. As you further develop your topics, flesh out the content with Index keywords, Glossary entries and links to related content. This pulls the entire Manual together for the user.

Designing a Template for your Content

As indicated above, one of the first tasks I recommend that you do in the early design phase of the actual project work is to sketch out a Template for your Topics. In traditional Web page design, a team of designers would be using tools like Macromedia Dreamweaver to design templates that would lay out the complete web site page, including navigation rollovers, banners, etc., and a team of programmers would create the code to make it work.

In RoboHelp, however, we are only dealing with the Topic Content itself, alleviating any real need for designing complex frames, tables or programming rollover effects. The first step was to create a quick sketch for my CAD Manual topic page, and then create a new Topic Template in the Project Manager's Templates folder.



I then started experimenting with a simple header/content/footer design. RoboHelp's Templates include a Header and Footer area. I wanted to design one that was simple yet understated and elegant.

The company's signature color is blue, and our logo is a small, blue stylistic "BHH" graphic, which would look good before the topic title on each page. I also set off the header and footer from the content area with a thin 1 pixel blue line, with 10 pixel padding between the three sections.

The footer includes two parts. On the left is a general information area that states the company names, the manual's title, version number and date. The right side of the footer contains a copyright notice and a "Comments" mailto: link to sends email to "cadmanager@bernardon.com"

For the footer I wanted the left side text to be left justified, and the right side text to be right justified. To do this I had to create a table with two columns, so that the text on each side could be justified as required.

For fonts, I wanted to match our corporate identity as closely as possible. In this case I used Times New Roman, Black, 18pt for the topic tile text, and Arial, Blue, 8pt for the footer. For the topic content, I used Verdana 10pt, a Sans-Serif font to maximize online readability.

For the header, I inserted a Topic Title field for the title text. This value relates to the Topic title in the Topic Properties, so the title will change automatically as required. Stylistically, I wanted the baseline of the text to align with the blue logo. To make this happen with descending lowercase letters, I added some white space to the bottom of the logo .gif, so that the alignment with the logo image would work without floating too high above the blue line. Tiny adjustments like this are common and can take some time to fully tweak.

For the horizontal lines, I first implemented them as HTML <HR> horizontal rules, but I could not get their placement quite right under the title text and logo. Instead, I moved to CSS styling to do this. Content in a Web page is based on a "box" model, where you have content surrounded by borders, surrounded by padding, surrounded by margins. I created a Header style which added a 1px thin blue line as a bottom border, with a 10 pixel bottom margin, and applied this to the table which held the header contents. For the footer, I added the same line as a top border, with a 10 pixel top margin, and applied it to the bottom table. The finished template design is shown here, with a dotted area that designates where the content would fall:

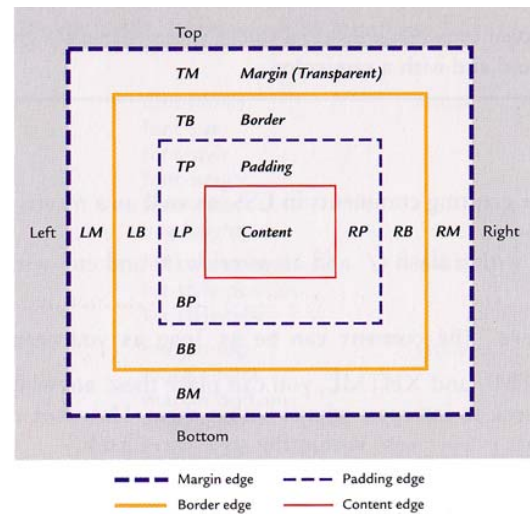
Topic Template

----- Content Here -----

Understanding the Box Model

To better comprehend how HTML content is rendered in the browser (and how to read the attributes that make up the look of an HTML file), it's important to understand the Box model of margins, borders, padding and content, as seen in this figure.

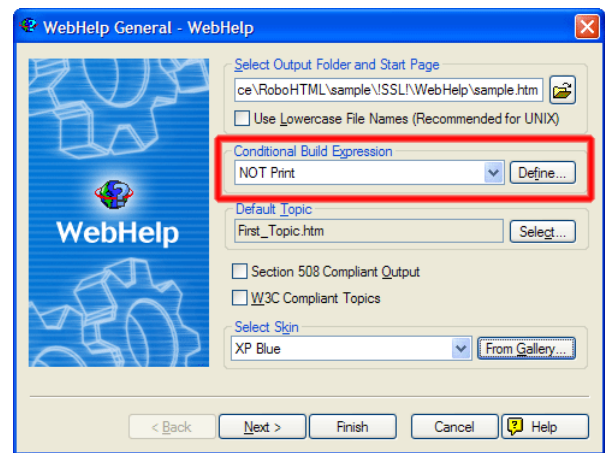
For the header portion of your topic template, you might want a top margin of 0 if you want to use a graphic banner, which leaves no white space to the top of the browser window. For charts and tables, you will probably want to use a low number for padding inside the cells and set the table borders to a width of 0 to make them disappear. This gives you a nicely formatted table without looking like an Excel spreadsheet.



Conditional Text

Conditional Text is a feature in some Help applications that allows you to generate multiple sets of content from a single project source. It works by defining rules for output, and “tag” topics or content with those rules. For example, you can generate a full version of your CAD Manual for mainstream purposes, yet generate a “Project Managers” version which includes much less content, targeted for their specific needs (e.g., printing). From a management viewpoint, you can define topics or portions of topics that are under development, and keep those from being output except in an “admin” layout.

When it comes to print output, the differences between the online and print formats can be rationalized by substituting printable versions of online content, such as bitmap images in lieu of DWF files, or textual instruction over an embedded AVI movie demonstration.



RoboHelp supports Conditional Text throughout the project, enabling you to tag content at the topic level, paragraph level or character level.

Conditional Text in your Template Design

One of the strange things in RoboHelp in particular is that Headers and Footers as designed in the Template do not make it to printed documentation – it uses its own header and footer which include the topic title and page number. If you want to include a printed header for the topic title similar to what was designed in the Template, you need to include in your template a copy of the topic header, with a Conditional Text “Print” tag in the *body*.

New topics based on this template will automatically inherit this body content. A Web Help layout configured to not contain Print content will not include this second header, but a Print layout will contain this topic title design as well as the default header and footer (page number) information for printing..

However, if you change the body content in the template, existing topics will not inherit this change. Topic templates allow automatic updates of header and footer content only.

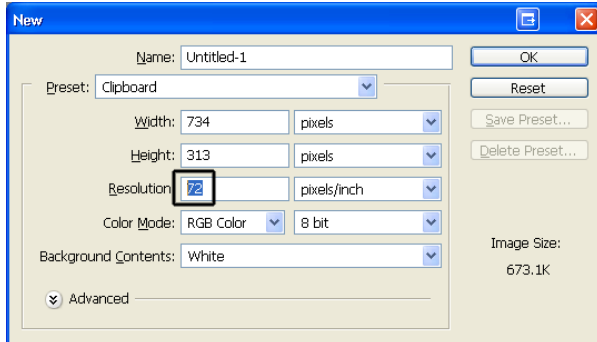
Screen Shots

Taking good screen shots is actually easy, once you do everything wrong a hundred times and finally figure out the best way to get decent results. Here's the technique I use with Photoshop, but any graphics application is similar.

The subject of resolution, expressed in dots per inch (dpi) is somewhat confusing. In printed output, dpi of images is important – anything less than 96 dpi and you run the risk of fuzzy output. But when you are working with online images, dpi doesn't matter – the image's pixel dimensions are what matters, and a pixel is a pixel is a pixel. Most modern monitors are somewhere between 96 dpi and 140 dpi, depending on resolution and physical size. I set up my screen shots in Photoshop to be 96 dpi, which seems to work well for both print and web.

The first step is to set up the content for the shot. Most screen shots are of dialog boxes or of resizable windows. When you can, resize the dialog or window to be as small as possible yet contain all of the pertinent information. You want the smallest physical screen shot that will get your point across. Use a white background for AutoCAD if you can.

After you size the window/dialog box correctly, hit (ALT + Print Screen). This captures the active window to the clipboard. Fire up Photoshop, and select File... New. The dialog box that you see looks something like this:



Note the width and height are exactly the same as the contents you copied to the clipboard. Also, note that for new files, Photoshop defaults to 72 pixels per inch – an old Macintosh screen standard. Change this to 96 ppi and hit OK. Then, do a CTRL+V to paste your clipboard contents into the new image.

Note: A screen shot image at 72 dpi is the same physical file size as is 96 dpi or 300 dpi. Remember that it is the pixel dimensions that count.

This technique is sometimes limiting because if your choices for screen capture are the full screen (if you just hit

PrintScreen) and the active window using ALT+PrintScreen - more useful, perhaps, but sometimes it's too much or too little. That's where software like Snagit™ comes into play, which allows you to take screen shots easily using a variety of methods. While Snagit has some bitmap editing tools, in the end I always use Photoshop to alter the images to suit.

Insert your Image into your topic, and you may notice that it's too big for the space. You will want to resize the image in Photoshop to a smaller physical size. Don't resize the image in your Help application.

Screen formats for online use

Historically, the best file formats for web images were either GIF or JPG. GIF files are limited to a palette of 256 different colors, one of which can be transparent. You would normally use JPG for photographs or high color images. The problem with JPG is that it is a lossy format, so low resolution images in JPG lose visible detail every time you save in that format. Today the new format of choice is PNG, especially for high-color images. PNG is relatively new, and was designed to replace GIF. It offers fully lossless compression almost equal to GIF but does not lose quality with successive saves like JPG. It also includes alpha channels (variable transparency), gamma correction and interlacing. However, PNG does not support multiple image animations as GIF does.

Because most screen shots of UI elements have only a few colors, GIF will probably result in a slightly smaller file. A quick comparison between the three formats for the above screen shot show the GIF file to be 13KB, PNG file to be 17KB and the JPG file at medium compression to be 61KB. However, sometimes GIF images don't look as good on screen, so PNG is a better choice. Information on PNG is available at <http://www.libpng.org/pub/png/>

Additional Screen Shot Workflow Tips

After you first take the screen shot, and before you do *any* work to it, save it as a BMP file into your working /images folder for your project. BMP is an unprocessed, uncompressed, unaliased, and overall very "raw" graphics format. This allows you to go back to the original artwork before any changes were made in Photoshop, should the need arise.

In Photoshop, you will be taking the original screen grab, possibly adding layers for content and effects, and saving an optimized version of the file for use in your document. Once you add this content on top of the original image, save this composite in PSD format before you do any further optimization to keep the layers.

Creating a Web-Based CAD Manual

Finally save a copy of your final graphic in the either PNG or .GIF. For GIF images, use the Save for Web option in the File menu, which will load the image into ImageReady, Adobe's web-graphics application. You can further optimize the image there to output to .GIF. Experiment with different file formats and parameters to make sure your screen grabs are always acceptable quality yet the smallest file size possible. Even though you are probably transmitting the pages over a fast corporate Intranet, optimizing images for the Web is always prudent.

Using DWF Files in your CAD Manual

DWF is a relatively new format but has undergone considerable improvements over the past few years. DWF allows you to create usable, interactive, online viewable drawings, complete with colors, layers, and even design information like distances.

There are several real advantages to using DWFs in your CAD Manual. First, the file size is incredibly small compared to PDF or bitmap formats like JPG. Second, DWFs are printable to scale. Third, it's a way of communicating CAD Standards, because you can allow users to turn layers on and off and display the DWF drawing either in color (useful if you use color based plotting) or in a plotted lineweight. Fourth, you can publish a whole set of drawings to a single DWF, giving your users a single reference source for how a baseline set of drawings is to look.

Browser Considerations

There are two ways of including DWFs in your manual. You can link to the DWF file, or embed an instance of the DWF Viewer ActiveX Control into the HTML, which itself links to the DWF.

When you link to a DWF, the browser launches the DWF using whatever application the system is registered to use for that file format, namely, the DWF Viewer. It is launched as an application in a separate window.

When embedded on a web page, the DWF is rendered inline of your content by the DWF Viewer's ActiveX control that is automatically installed on your computer. This ActiveX Control is installed either by AutoCAD or the DWF Viewer application, or as a separate plugin downloadable whenever the browser hits a page with a DWF on it.

Unfortunately, being an ActiveX control, this means that only Internet Explorer on the PC is able to exploit the power of inline DWFs natively. Other platforms need to launch the DWF as a separate application, which may fail if the DWF Viewer is not installed on the system.

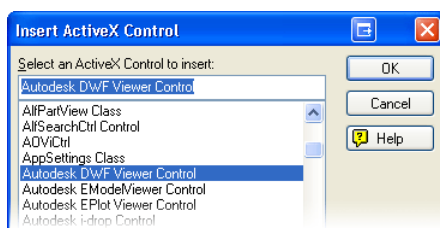
You aren't limited to using a Web Browser to view DWFs in documents. You can embed Autodesk DWF Viewer in any application that supports ActiveX controls, such as Microsoft PowerPoint, Autodesk Buzzsaw, or Microsoft Word. You can embed the viewer in an HTML file or web page and use the Autodesk DWF Viewer Application Programming Interface (API) properties and methods to design the page.

These differences are important, because it's been my experience that embedded DWFs are somewhat problematic. First, the page will load slower than normal as IE will initialize the ActiveX Control. Second, you are limited to using Internet Explorer as a browser, which is no longer a given. The main benefit of using the ActiveX Control is that you include instruction for downloading the proper plug-in when required – you don't have to assume the user has the DWF Viewer installed. However, if you are creating a CAD Manual for in-house CAD users, and each user has either AutoCAD 2004 or 2005 installed, then there is no problem. Also, weekly breaches in the ActiveX Control security model mean IE may not be the smart option long term. So it's a trade off.

Using DWFs in RoboHelp

In RoboHelp, it's important to place the DWF as a Baggage File in the project. Baggage Files are external files that RoboHelp will include in the final output. To do this, right-click on the Baggage File folder in the Project Manager and select New Baggage File. Navigate to the drive/folder that contains the file, and select it. Click Yes at the prompt, and the file is added. Note that you want the DWF to be in the same folder as the page you are going to be inserting it into, so the path to the DWF is simply the file name

Inserting the DWF inline in RoboHelp (for IE users)



In RoboHelp, place your cursor at the location at which you wish to insert the DWF file. In the Insert menu, select Advanced, ActiveX Control, and select the Autodesk DWF Viewer Control. You will see a small box appear in your content with the words "ActiveX control" inside of it.

Right-click on the box and select ActiveX Control Properties. We need to specify the source DWF file and its dimensions on the page. Hit the Add button to bring up the Add Property dialog box, and type SRC in the Name drop down list (for some reason, "src" is not an available property). Under Value put the relative path to the DWF file, e.g., floorplan.dwf. Unfortunately there's no Browse... button to select the file, so you have to type everything in correctly. You can also set the Width and Height in the Size tab, which is recommended. In the ActiveX Control properties dialog,

Code Source edit box, include the following link:

<http://www.autodesk.com/global/expressviewer/installer/ExpressViewerSetup.cab>

This allows the browser to download and install the ActiveX Control on machines which do not have the DWF Viewer already installed.

Referencing the DWF ActiveX Control using HTML (Geek Warning: HTML Ahead)

As seen above, the UI in RoboHelp helps you reference the DWF control in your web page. It basically does the grunt work of creating an <object> tag with the correct parameters (called Attributes in HTML) to reference the ActiveX Control on the system. The <object> tag is HTML code that embeds multimedia objects in web pages. You generally use this tag with <param>, which specifies the parameters of the <object>. More information on the <object> and <param> tags is found at <http://www.htmldog.com/reference/htmltags/object/>.

Multimedia "players" and ActiveX controls are referenced in an <object> tag using a classid attribute. The classid parameter specifies the location of the object in the form of a URL or a Windows Registry location, and used as such is a globally unique "serial number" which identifies the particular applet in question. The classid for the DWF Viewer is "clsid:A662DA7E-CCB7-4743-B71A-D817F6D575DF"

Other attributes in the <object> tag will tell the browser how to deal with this instance of the ActiveX Control. For example, you can reference the Autodesk DWF Viewer control resource from the Autodesk web site using the CODEBASE attribute. When you use the CODEBASE attribute, the user will be prompted to download or install the viewer if the viewer is not already installed on the local machine. The "Src" attribute in the <param> tag specifies a DWF file to load:

```
<object id = "viewer"
classid="clsid:A662DA7E-CCB7-4743-B71A-D817F6D575DF"
CODEBASE="http://www.autodesk.com/global/expressviewer/installer/ExpressViewerSetup.cab">
<PARAM NAME="Src" VALUE="http://www.autodesk.com/global/expressviewer/installer/sample.dwf">
```

If you include a build version number in the codebase attribute, the user will be prompted to install a new version of the viewer if the viewer installed on the local hard drive is not at or above the specified build version number:

```
<object id = "viewer"
classid="clsid:A662DA7E-CCB7-4743-B71A-D817F6D575DF"
CODEBASE="http://www.autodesk.com/global/expressviewer/installer/ExpressViewerSetup.cab#version=5,0,0,71">
<PARAM NAME="Src" VALUE="http://www.autodesk.com/global/expressviewer/installer/sample.dwf">
</object>
```

If the Autodesk DWF Viewer is already installed on a local machine, you only need to use the classid parameter. Use the <param> tag to specify a DWF file to load:

```
<object id = "viewer"
classid = "clsid:A662DA7E-CCB7-4743-B71A-D817F6D575DF">
<param name = "Src" value="New.dwf">
</object>
```

As an accepted "best practice," it's a good idea to not hard code in the locations of DWF files (unless they are URLs on a separate server), but make them relative to the current page. You would usually put the DWF in the same folder as the HTML page that references it.

Creating a Web-Based CAD Manual

Creating a DWF link to launch in a separate window

Creating links which launch external windows is easy, and is necessary for non-IE users. Use the `target="none"` attribute in the `<a>` tag to launch the DWF in an external window for IE users, eliminating the need to hit the back button. For Firefox users, a simple link will launch the full DWF Viewer application if installed.

Example:

```
<a HREF="TypFloorPlan.dwf" target="none">Here is a DWF of a typical floor plan</a>
```

Multimedia and onscreen movies

Probably the most useful and important way you can move your CAD Manual to the online world is to include AVI files of onscreen CAD techniques and practices. No other mechanism provides such a direct means of communication to the end user as to how things work. There's no reading involved and, if done correctly, no way to misinterpret the procedure. However, understand that creating successful video is hard work and takes time and practice.

One application you can use to do this is Camtasia Studio (<http://www.techsmith.com/products/studio/default.asp>) Camtasia Studio is a complete package that allows you to record onscreen activities to an AVI video file. You can annotate the movie while you are performing the actions, or you can add a voiceover afterwards.

Making instructional videos is tough and takes practice. Here are a few tips to get you started:

- Use a decent quality microphone/headset combination. These are easy to find and are pretty cheap.
- In you application, only have the UI elements on screen that you need to complete the demonstration. Maximize the amount of editor screen space by hiding unnecessary toolbars or palettes.
- Make the movie at least 800x600, and 1024x768 is better. Although the final movie's file size will be larger, it will be streamed from the web server to users, so you really don't have to worry about sacrificing quality in the name of smaller files.
- Make a text file script of the step-by-step instructions *before* you start recording. Read the script out loud as you perform the actions to time yourself. Sometimes you will want to stop drawing, to give yourself time to explain something.
- Pad the beginning of the video with no action, and voiceover a description of your video.
- Record your voiceover *after* you record the onscreen action. This allows you to concentrate on speaking clearly, which probably won't happen as you try to talk and draw at the same time.
- Don't say "UM..." Re-record the voiceover over if it slips out.
- Don't move the cursor wildly about – be deliberative in your onscreen mouse actions.
- Include special sound effects like keyboard and mouse clicks sparingly. They can provide aural feedback on what you are doing onscreen, but if they are too loud or overbearing they won't be effective.
- Camtasia and other applications include post-production tools to add overlay graphics and text to the movie, e.g., listing the title, a bullet list of explanations, and so on.
- When you link to the video in your CAD Manual, use either with a simple text link or an image of the first frame. The link will launch the movie in a separate window using the preferred media player on the system. While you can embed the movie inline, sometimes these will want to start playing as soon as the page loads or the user will accidentally click on the link to start it off. It also makes dealing with print easier, because you don't have to deal with conditional text.

Additional References

General HTML: <http://www.htmldog.com/reference/htmltags>

Autodesk DWF API: <http://www.autodesk.com/dwfvviewer-api>

Molly E. Holzschlag, Cascading Style Sheets: The Designer's Edge, <http://www.molly.com>

Thank you for attending!