JavaHelp Basics

This page covers:

- An outline of what javahelp is.
- What type of help you need.
- The tools you need.
- Producing javahelp.

If you prefer to print and read this topic, click the Show All button and then use your browser's print function. The images are deliberately not scaled down so that you can see them without distortion.

This is an old topic that I will not be updating. However, JavaHelp is an old format so you may still find it useful. Please do note the caveats before you go too far.

June 2019 - JavaHelp can only be created from Classic versions of RoboHelp. It is not supported in RoboHelp 2019.

20 February 2020 - I am told the download links below no longer go to the downloads required. I have left them in case they help you track down the required downloads.

To the best of my knowledge, there are few users of javahelp now, partly because it was never very well supported and partly because other more modern outputs can be used instead.

Introduction

This topic was written following a crash course in producing some JavaHelp. I do not have extensive experience of producing javahelp and prior to the request to produce some, I knew nothing of this area. I had to do some quick research but fortunately I work with some very good developers who eased the learning curve tremendously. I am grateful to Paul Symonds in particular for explaining the concepts and to Jon Hawkins for getting the necessary software working for me.

Before you go too far, I suggest you read Mike Hamilton's comments in Updates below. It is a poorly supported format and usually something else will do the job. There seems to be a misconception that for Java apps you have to have Java help.

Java v Javascript

Let's get this clarified first as it does cause some confusion for people new to this area.

- Java is a language to write applications.
- Javascript, whilst from the same stable, relies on the browser for its functionality. The syntax is similar but not identical.

This topic is about the help for an application written in Java.

What's different about help for a Java application?

Programs are written in Java as this enables the same code to be used on a variety of operating systems, Windows, Unix, Linux and Mac for starters. The way Java applications work also means that many of the usual rules about help do not apply here. For example, under Windows with XP Service Pack 2 applied, webhelp cannot easily be run if installed on the users hard disk. Such rules do not apply to Java applications. The best way it was described to me was that a Java application "can get under the covers" and it's down to the developer what can be done. Using a CHM file is an exception to the "rules" the developer can get around as a CHM file is reliant on files that will only be on a Windows operating system.

That means that there are three options for producing help for a java application.

Option	About
WebHelp	You produce this in exactly the same way as you normally would. The rule about not running it from a hard disk under Windows XP Service Pack 2 does not apply. Anyway that's for the developer to worry about and I understand that it is not a major problem. I believe it is true to say that map ids cannot be used but that does not mean that context sensitive help cannot be provided. The developer has to call each topic by using its html file name.

Compressed JavaHelp	This packs the files used into a .JAR file which for our purposes is much like a winzip file. The developers can access the files they want from the .JAR file and display them in a javahelp viewer.
Uncompressed JavaHelp	This is similar to webhelp in that the files remain in a folder structure, are not compressed and are not packed into one file. Like compressed javahelp, it is viewed through a javahelp viewer which we will cover later.

Which method you use will depend on several factors:

- Webhelp gives you exactly the same appearance to which you are accustomed. The user views it in a familiar browser, do not underestimate the importance of that.
- The Sun JavaHelp User Guide (October 2004 version) indicates that it is possible to view JavaHelp via the native browser. That is not covered in this topic and you would need to discuss that with your developers.
- The display of both types of javahelp depends on whose javahelp viewer you use. I found the Sun javahelp viewer gave an unacceptable display by default whereas the Oracle javahelp viewer was fine. If you are required to use the Sun viewer, do make sure that management are aware from the outset how it will impact on appearance unless you spend additional time working on it.
- Technical issues may drive the developers to insist upon a particular method. Be sure though that the arguments put forward are genuine rather than for their convenience!

What next?

Before developing your help, I suggest you really need to sit down with the project and product managers to ensure they fully understand the differences between webhelp and javahelp and the differences between javahelp displayed in the Sun viewer and the Oracle viewer.

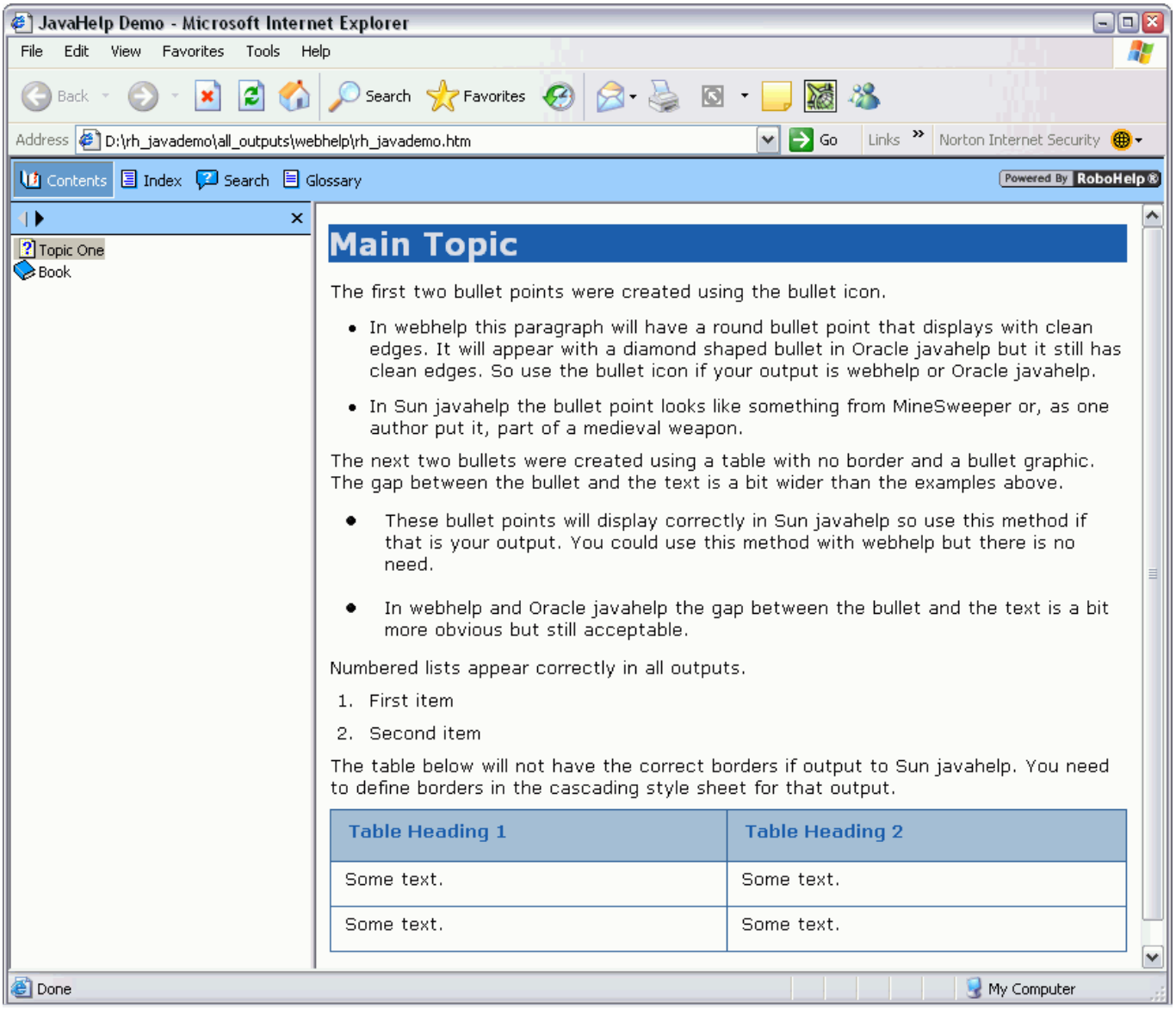
Click the links below to see the same topic displayed in various formats.

- There is no difference in the display of compressed and uncompressed javahelp. The differences depend on the viewer that you use.
- Note the information in the topic displayed in the screenshots, it explains the problems you will run into and forms part of this topic.

Show All Hide All

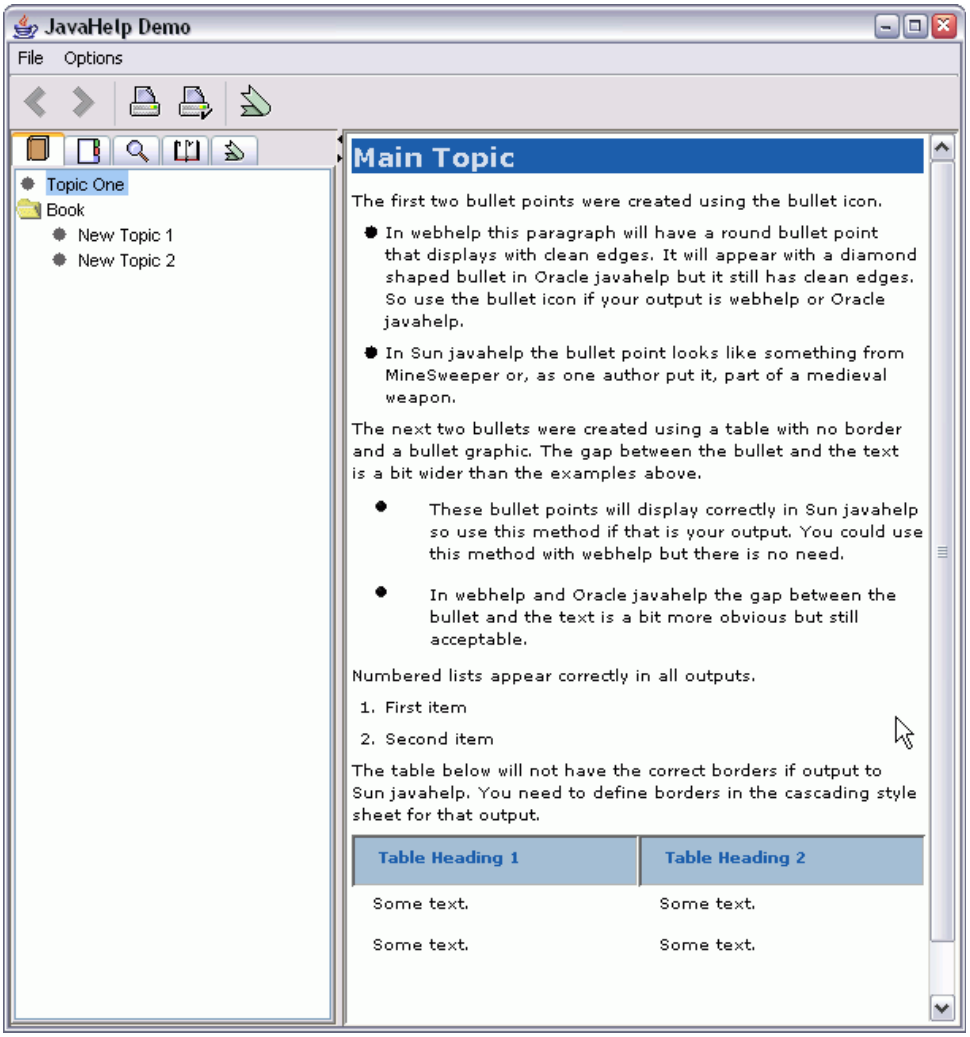
Click here to see webhelp

In this screenshot, everything displays exactly as you would expect. Note all the bullet points are clean and the table cells have a thin border.



Click here to see javahelp in the Sun viewer with no tweaks applied

See how the bullet points look like they are from MineSweeper and the table cell borders are wrong. Also note the appearance of the navigation pane.

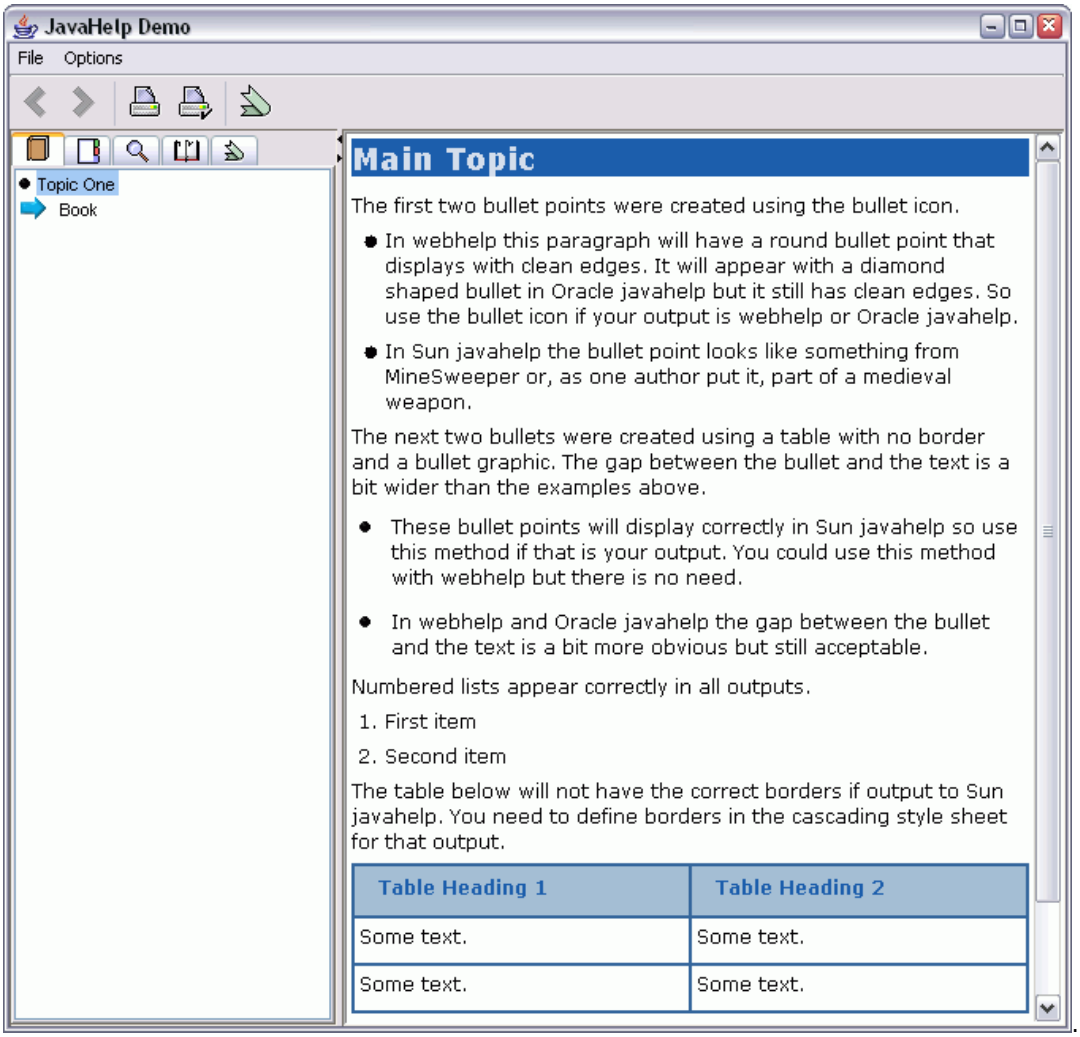


Click here to see javahelp in the Sun viewer with tweaks applied

The right pointing arrow has been set up to work like a Lotus Notes twisty. When you click on it, the arrow will point down and show the topics available. For lower levels of book, there will be plus / minus signs which I don't think you can avoid.

Note that although I have got borders around the table cells by defining them in the style sheet, the appearance is not the same as in the webhelp.

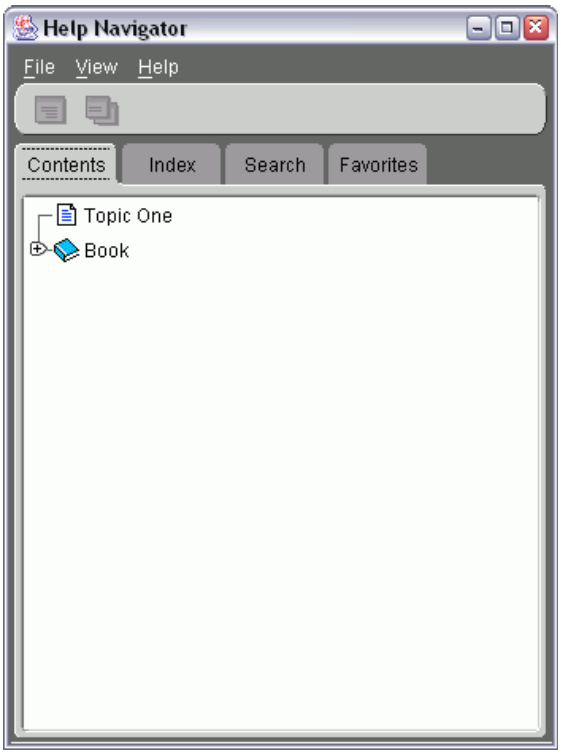
Because the border is defined in the style sheet, it will affect all tables so I created a class to give more than one type of border. Nonetheless it still took a lot of tweaking to get rid of the border around the second set of bullet points. That tweaking would be necessary in all topics every time you generate. Not really viable unless you can make it work better. Using this viewer, I think you would end up deciding to accept no borders around table cells



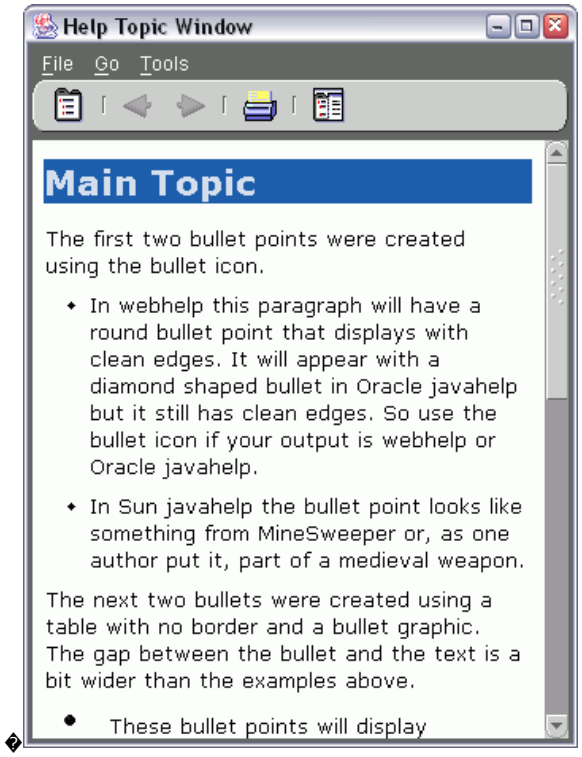
[Click here to see javahelp in the Oracle viewer with no tweaks applied.]

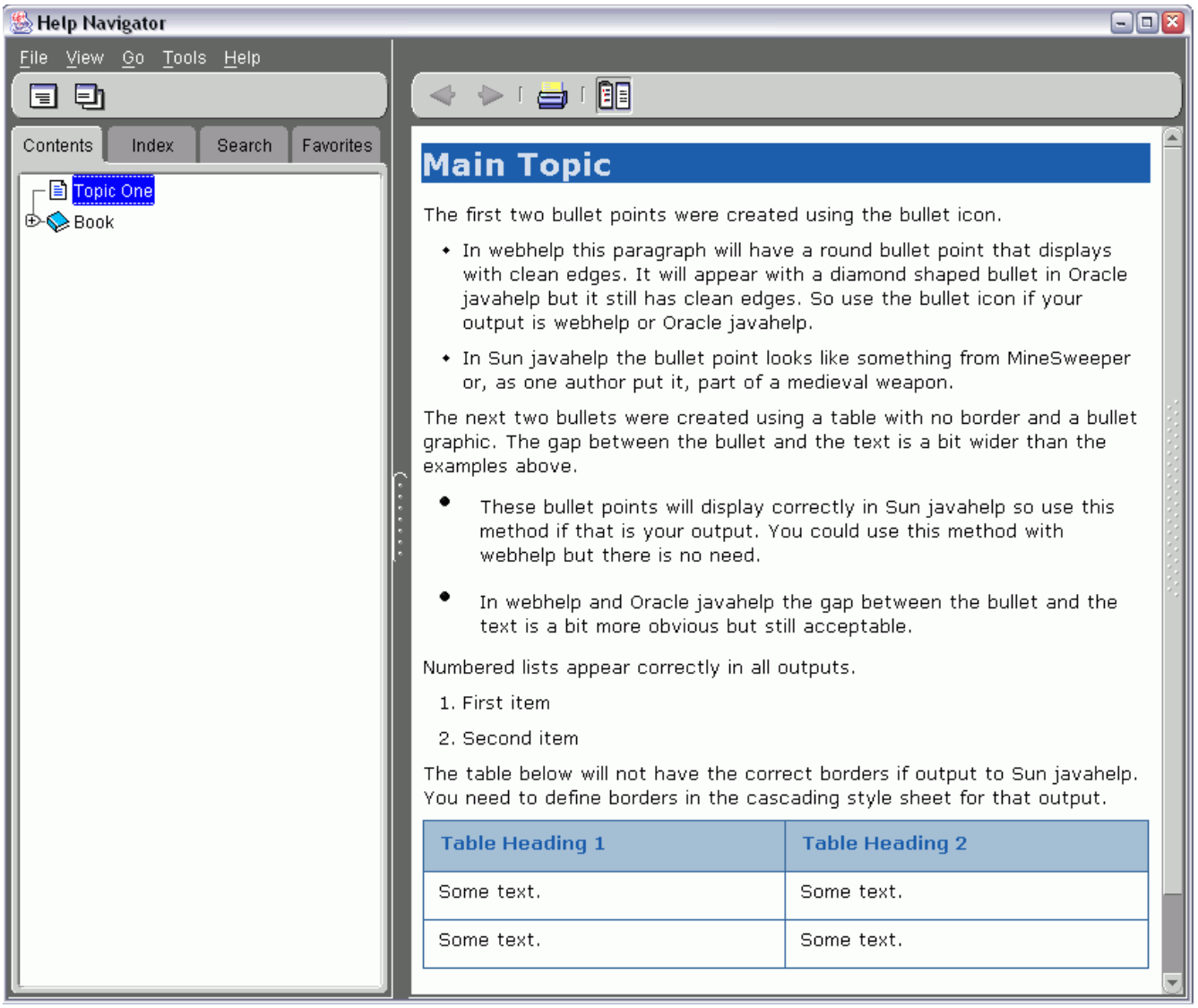
Initially only the Navigator is displayed but note the familiar books and the cleaner appearance, with no tweaks. The user can double click a topic or highlight it and click the icon below File in the menu. A second window will then open alongside with the topic displayed. The two windows are easily docked if that is the user preference.

You can change the images in the navigator pane by following the method described for The Sun Viewer.



On screen, this second window will display alongside the navigator window.





After considering all the pros and cons of these methods, it is then up to the project manager to balance the impact of you having to learn / work with javahelp and any developer issues with the help format.

- If the decision is to go with webhelp, smile and read something else on this site.
- If the decision is that one of the javahelp flavours is required, then read on.

Limitations

Javahelp has limitations as you have seen. In RoboHelp go the Help menu, click Contents and Index. In the index look for JavaHelp and then limitations. Be ready to scroll down the list!

The good news is that the Oracle viewer seems to handle them. For simplicity, in this topic I will assume that those of you still going with javahelp will persuade the powers that be to go with the Oracle viewer. If for some reason you are compelled to use the Sun viewer, there are some workarounds that are described later.

What tools do you need?

Update 19 April 2019 - The links in this section have changed since I created this page some years ago. I have tried to find the equivalent pages but they may not be the version I tested.

For the production of the topics, the index and the contents, I have written this topic assuming you are going to use RoboHelp but you can use other tools, you will need to amend the procedures. Anything I say about RoboHelp from hereon in this topic is based on RoboHelp HTML X5 although RoboHelp has supported javahelp since RoboHelp 2000.

You will need to download some resources from the Sun and Oracle sites. Speak to your IT people before installing any software.

- This link will take you to the main area of Sun's site for javahelp. You need to download and install the JavaHelp resources.
- I downloaded JH2.0 from http://java.sun.com/j2se/1.4.2/download.html but the page has moved. Try this link instead.
- This link will take you to an area of Sun's site where you can download the current J2SE Software Development Kit (SDK) that you will also need. You need a download with a title something like j2sdk1.4.2_08. You do not need the Enterprise version.
- My original link took you to an area of Oracle's site where you could download Oracle Help for Java. Try this page but note it may not be the same as the version I used.

The downloads come with documentation and / or online help. Do read these resources. They contain a lot of useful information from which I found the tweaks I describe. Many of the methods described in the Sun documentation also apply to Oracle and vice versa.

After you have downloaded and installed the Sun resources, you need to create some Environment Variables before installing the Oracle Help for Java. These can be set up as User Variables or System Variables. If that is meaningless, you may want to enlist the help of your IT people or a developer in undertaking these steps.

1. In Windows Explorer right click on My Computer and select Properties.
2. Click the Advanced tab and click Environment Variables and set up the following as either a user variable or a system variable.
3. Create two variables. JAVA_HOME and JHHOME. For each one, enter the variable name and in the value enter the top level folder, for example for JAVA_HOME the value on my system is C:\j2sdk1.4.2_06
4. Scroll down and locate the Path variable.
5. Enter %JAVA_HOME% followed by the rest of the path to the folder containing the java.exe file. Enter a semi colon after this.
6. Enter %JHHOME% followed by the rest of the path to the folder containing the hsviewer.jar file. Enter a semi colon after this.
7. OK your way back out.
8. To check things are OK locate one of the Sun supplied help files and open the Run command. Enter hsviewer.jar - helpset followed by the path and filename. It will probably look like this.
   hsviewer.jar - helpset c:\javahelp\demos\hs\newmerge\mergehelp.hs
   The instructions in the JavaHelp guide prefix the above with java - jar but I found that did not work.
   If this does not work, try rebooting your PC.
9. Now you can install the Oracle Help for Java after which you are set to discover the joys of working with javahelp.

Using RoboHelp to produce the topics

You need to decide which viewer is going to be used before starting production.

- If you are using an existing project, that may rather force the issue as to which viewer you use if the content is not compatible with the Sun viewer.
- If you are going to use the Sun viewer, you need to sidestep to the separate topic on that subject. It explains the limitations I found and the workarounds. Click here if you are using the Sun viewer, otherwise read on.

OK, if you did not have to click the link will be pleased to know you just work normally in RoboHelp. Create your bullet points as normal and create your tables as normal. Use your conditional text and as far as I could make out, any of the other features. They all work fine in the Oracle javahelp viewer.

Generating the Help

Welcome back to those of you who had to sidestep and read up on the Sun viewer.

In RoboHelp Explorer, look under Single Source Layouts and see if there is a JavaHelp or Oracle output option already shown. If there is not, right click and select JavaHelp or Oracle from the list that will be displayed.

Start the wizard and go to the Output frame on the first page. Select one of the options for compression and the javahelp version you require. Everything covered in this topic is based on selecting the JavaHelp 2.0 options. Work through the wizard.

- If after clicking Next on the first page you get a Java SDK requirement dialog referring to Search Text, then RoboHelp is not finding the SDK that you installed earlier.
- If you are producing javahelp for the Oracle viewer, the first time you run the wizard you will need to point to the installation directory.

Viewing the Output

You will be able to see the output after generating it by right clicking the layout and selecting View.

Sun Viewer

You can also view the help by entering something like the following into the Run prompt.

For compressed javahelp

hsviewer.jar -helpset D:\rh_javademo\all_outputs\uncompressed_oracle\rh_javademo.jar

For uncompressed javahelp

hsviewer.jar -helpset D:\rh_javademo\all_outputs\uncompressed_oracle\rh_javademo.hs

Oracle Viewer

For compressed javahelp

I was only able to open this via RoboHelp. Not sure why.

For uncompressed javahelp

Go to Start and Programs. Find Oracle Help for Java and run the viewer.

Conclusion

I hope you find this topic of some use in getting started with javahelp and would welcome any feedback.

May I again express my special thanks to Paul Symonds and Jon Hawkins without whose help I would not have been able to produce some javahelp for my employers or write this topic.

Download

The topic shown in all the screenshots is contained in a RoboHelp HTML X5 project that you can download. That gives you the opportunity to try out what I have described and see how well you get on. Your developers may also find the outputs useful to test their side of the job.

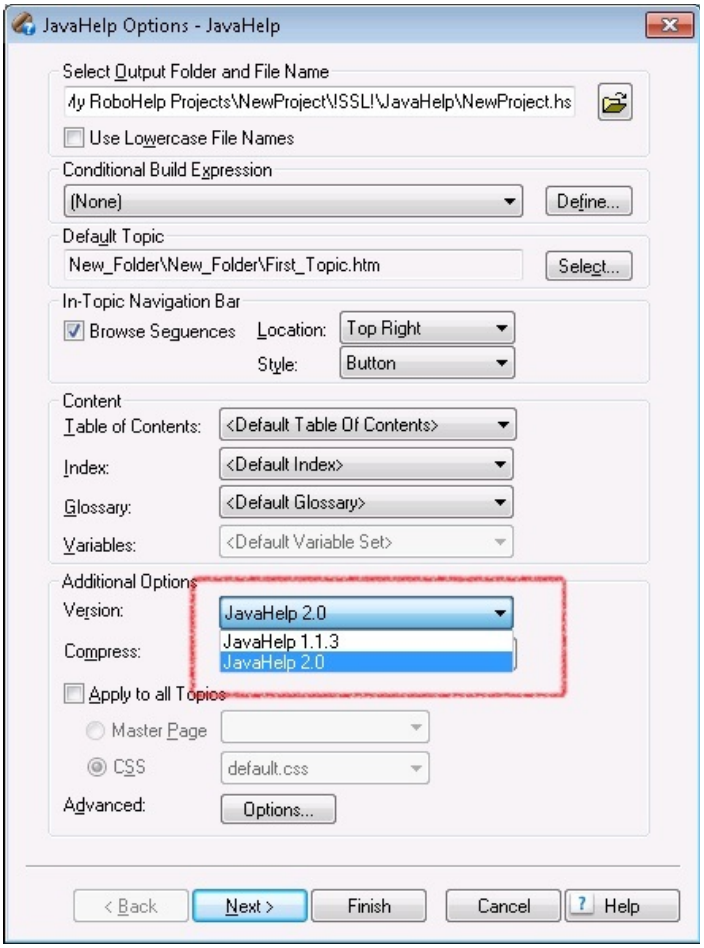
Updates

It is not my intention to develop this topic further as I have no occasion to work with javahelp. However, I will add to this section any useful bits of information that I am sent. The information will be as sent to me, just edited for brevity and / or clarity.

Credit	Information
Kathy Hanten	I have to use the Sun Viewer and one thing I had to mess with is any text that I made bold appeared to have extra spaces around it. After some research, I found that I had to replace the RH code of with something like or and that made the extra spaces go away.
Jim aka Maniac9999 See this post on RH Forum	I was scrolling through this folder and found your post. I don't know if you found an answer to your problem, but here's my two cents. I would STRONGLY recommend that you avoid JavaHelp 1.1.3, unless you really need to use it. In my experience, I found it was full of bugs, most notably that it would crash or hang when I tried to print a topic from within the JavaHelp output. JavaHelp 2.0 is much more stable. Secondly, I just tried building a 1.1.3 JavaHelp in my full version of RH X5 and got a fatal exception error both when I created compressed JavaHelp (which is what we use for JavaHelp 2.0) and uncompressed. I know I created JavaHelp 1.1.3 using a product called Helen, but I must not have built it using RoboHelp. Unfortunately, it looks like it's a flaw in RH. If you really, really need to build JavaHelp

	1.1.3, look at Helen from Software7 (www.software7.biz). It's written in Java, is built for JavaHelp only and it allows you to build for 1.1.3 and 2.0. The only reason we're not using it now is that management didn't want to support a product that could only create one type of output. It has a good wizard to create a project, but it's more of an "assembly" tool - you create your topics outside of the program in HTML, then import them into Helen. My experience with support, even though they're in Germany, was outstanding.
Greg Minter	As promised, I wanted to post what my successful installation looked like so that others might benefit. Unlike Jim suggested, I did not have to uninstall RoboHelp, and I don't think that's necessary. It is, as he suggested, just more important to get the Java install right and have the path and env. variables set properly. 1) RoboHelp Version: RoboHelp 5.0.2, Build 801. I have this installed in the default location: C:\Program Files\RoboHelp Office 2) Java SDK The 1.6 SDK didn't seem to work. So I stripped off all the Java installs from my machine (both SDK and JRE) and started from scratch with the 1.5 release. I downloaded the SDK from this URL http://java.sun.com/javase/downloads /index_jdk5.jsp but that page is no longer available. Try this page instead. Choose the "JDK 5.0 Update 11 with NetBeans IDE 5.5" option. This installs the SDK, the JRE, and NetBeans (the IDE for Java). You probably won't use this as a tech writer, but it's there as part of the install, and won't hurt anything. One thing I did when installing the SDK and JavaBeans was to select a path that had no spaces in it (i.e., not in "Program Files"). Installs like this sometimes automatically update your path, but don't always manage spaces properly, so this was just to be safe. I used the default location for the JRE (runtime environment). Here's a cut and paste from my install dialog: J2SE Development Kit 5.0 Update 11 installation location: C:\Java\jdk1.5.0_11 J2SE Runtime Environment 5.0 Update 11 installation location: C:\Program Files\Java\jre1.5.0_11 NetBeans IDE 5.5 installation location: C:\Java\netbeans-5.5 3) JavaHelp You also have to have JavaHelp installed on your machine. JavaHelp 2.0 is kind of the baseline now, apparently (the previous release 1.1.3, was pretty limited). I downloaded JH2.0 from http://java.sun.com/produts/javahelp/index.jsp but the page has moved. Try https://www.oracle.com/technetwork/java/index.html I put it at the same level as the Java SDK install: C:\Java\jh2.0 4) Environment Variables As Jim pointed out, you need to set the Environmental Variable "JHHOME". The value of that path should be the root directory for your JH install, so: - Right click on My Computer > Properties. - Choose Advanced > Environment Variables > System Variables - Check the value for JHHOME and make sure it matches the install dir. for JH (in my case, C:\Java\jh2.0) 5) Path The path variable tells the OS where to look for executables it can run. Edit that as Jim pointed out. Just for future reference. This was a bit painful, but mostly because I wasn't familiar with Java, but also the doc on the RH side was a bit deficient in guiding me toward a successful install.

Jim aka Maniac9999	In repsonse to another much later post Jim added this: Speaking of JavaHelp, I just found out over the last few weeks that RoboHelp 7 only supports JavaHelp 1.1.3, not JavaHelp 2.0, which is much more stable. I found this out after installing RH 7 and trying (and trying) to get it to build JavaHelp 2.0. I finally gave up and called Adobe Support. Adobe emailed me last week asking which JRE I used (1.5.0_06) and I'm hoping to get a reply soon. They also said they didn't have any plans for a future patch to get this to work, but there might be the possibility of me getting a DLL. Like you mentioned before, the authoring community for JavaHelp is probably tiny.
Mike Hamilton	In a post on HATT (7 Aug 08), Mike Hamilton explained the main reasons MadCap Software don't have a JavaHelp output in Flare and these include: A) Java is nowhere near as cross-platform and headache free as is W3C compliant XHTML/DHTML. There are just too many variables such as what JRE is installed, who made the JRE, "oh, my app only runs in the Microsoft JRE, but not the Sun JRE" etc. B) When I was running the RoboHelp program we spent a HUGE amount of time and money supporting both JavaHelp and Oracle Help only to find that a super small percentage of the customer base (as in less than three percent) used either format. To make matters worse, that small group of users generated a huge amount of burden on the tech support staff. C) Sun and Oracle have really had a spotty, on again/off again track record for their own formats. It is tough to justify the dev hours for something the vendor themselves won't champion full time. D) Finally, the functionality just wasn't there. Most of the people who were absolutely begging for JavaHelp support were completely underwhelmed with the format once they were actually able to generate it. Most opted to switch to WebHelp for the better customer experience and the ability to brand the interface to match their product and their corporate look and feel (something difficult with the Java-based formats). Please note, I'm not saying that either JavaHelp or Oracle Help for Java are bad formats, just that we haven't seen enough demand to justify including them in Flare. In fact, we get more requests for formats like AppleHelp and Eclipse Help than we do for JavaHelp. We are always open to adding new output formats, in fact the next release of Flare includes several new output formats, we just have to see a business case for including them. If my phone suddenly started ringing off the hook with people asking for BobHelp, then we would research it and include it.
Praful Jain	Q-Term posted this problem: Since I am generating a context-sensitive help and my customer is only able to implement a help project based on JavaHelp 1.1.3, I cannot use JavaHelp 2.0, though there is a whole series of restrictions with JavaHelp 1.1.3. With RoboHelp v7, everything worked fine when creating a JavaHelp 1.1.3 project. With RoboHelp v8, however, the TOC is missing in the JavaHelp. It is not displayed. Usually, I am working on my main PC under Windows XP. I tried to generate a JavaHelp 1.1.3 version with RH 8 under Vista (which I have on my notebook), and the TOC was displayed. The project source is a Web Help project. When I copy the entire project with the generated JAR file to the PC with Windows XP installed, the TOC is not either displayed. Praful Jain responded with this: Each JavaHelp file has a helpset file with extension .hs. This HS file tells javahelp about the help file structure and where the TOC file, index file, and other information can be found. the problem lies in this .hs file itself. The problem line is <view mergetype="unite-append"> you need to change this line to <view mergetype="javax.help.UniteAppendMerge"> and now try running the javahelp again. You should be able to see the TOC, that's what I did to make TOC appear in javahelp. Better still, Praful then added this screenshot from RoboHelp 8 where you can select the JavaHelp version and that should make the code change for you.



IMKarla2	Some information that may be useful at the end of this thread. https://forums.adobe.com/message/4624201#4624201

Donations

If you find the information and tutorials on my site save you time figuring it out for yourself and help improve what you produce, please consider making a small donation.

Topic Revisions

Revisions

Date	Changes to this page
17 Jun 2019	Amended to cover applicable RoboHelp versions.
06 Feb 2017	Topic reviewed. Caution added. Any screenshots may differ in your version of RoboHelp but the functionality is the same.
18 Aug 2012	Link to IMKarla2's post added.
10 Feb 2012	Updated with Praful Jain's response to a JavaHelp version problem.
14 Aug 2008	Updated with a further comments from "Jim aka Maniac9999" and Mike Hamilton.
04 May 2006	Updated with Greg Minter's RoboHelp forum post.
27 May 2006	Snippets updated with information about version 1.1.3.
26 Feb 2006	What tools do you need? amended to indicate you can use tools other than RoboHelp.
07 Oct 2005	Snippets section added.
12 Apr 2005	Updated to cover editing compressed javahelp and presenting javahelp via the native browser.

10 Apr 2005	Major rewrite to add information about the Oracle javahelp viewer and to separate out the issues when using the Sun viewer for javahelp.
05 Apr 2005	New topic.