

## Problem

How do I create Java Help (online help) for an application?

## Solution

### JHelpGen

Sun provides a help viewer facility called "Java Help" which usually is contained in a jar file `jhall.jar`, or something like `jhall-2.0.22.jar`. `acs` provides a utility called "JHelpGen" which can *i)* convert existing html pages into the format understood by Java Help, *ii)* start a help browser from the commandline in order to verify the produced help, and *iii)* supports an application in bringing up a help browser at runtime. This page describes usage of the JHelpGen tool.

### What does JHelpGen do?

The tool scans a directory recursively for html files. It will parse each file for Html-Header-tags and create a Table-Of-Contents (TOC) from them. The results will be written to some meta-information files that can be used by a Java Help Browser.

The results can be viewed afterwards directly to sanity check what you've created.

Finally at run-time you can use the tool to bring up a help browser for the users.

See `acsJavaHelp -h` for usage info.

### What do I need to do?

#### Html Contents

You need to author the help content: html pages, CSS, images, ...

The help content must reside in one directory tree, the root of which we refer to as the *HELP-DIR*.

#### Example:

```
ACS/Documents/ACSCommandCenter/AcsCommandCenter-News.html
ACS/Documents/ACSCommandCenter/Acs_Command_Center_-_User_Guide.html
ACS/Documents/ACSCommandCenter/style.css
ACS/Documents/ACSCommandCenter/images/alma_art_3.jpg
ACS/Documents/ACSCommandCenter/images/eso-logo.gif
ACS/Documents/ACSCommandCenter/images/figureA.jpg
ACS/Documents/ACSCommandCenter/images/figureB.jpg
ACS/Documents/ACSCommandCenter/images/figureC.jpg
ACS/Documents/ACSCommandCenter/images/figureE.jpg
ACS/Documents/ACSCommandCenter/images/figureF.jpg
ACS/Documents/ACSCommandCenter/images/figureG.jpg
ACS/Documents/ACSCommandCenter/images/rightarrow.gif
```

Here, the *HELP-DIR* is "ACSCommandCenter"

#### Help Generation

To convert your help contents into the format understood by Java Help, they need to be enhanced with Java Help meta files. To do that, you run the following command:

*NOTE: In this version, it is important that you run the command in the parent directory of the HELP-DIR (in our example this is "ACS/Documents")*

```
acsJavaHelp -gen HELP-DIR {TOC-DIR|TOC-FILE}+
```

This will parse all specified files (the TOC-FILES) for Html-Header-Tags and build a Java Help index from them. For the parsing process to succeed, the Html-Header-Tags need to comply to a certain structure:

```
<h1><a name="Duke">The Duke</a></h1>
```

A Header-Tag needs to contain an Anchor-Tag (`<a>`). The text contained inside the anchor tag will be used for the entry in the TOC.

#### Notes:

- Header-Tags can be nested arbitrarily as long as it is valid Html
- You can easily exclude parts of the header text from being included in the TOC entry:
 

```
<h1>4. <a name="Duke">The Duke</a></h1>
```

#### TOC-FILES:

If you don't specify TOC-FILES, all html files will be considered TOC-FILES. All will be processed recursively, in alphabetical order. If you require an order different from this, or if you want to parse only certain files you need to specify the TOC-FILES manually. A TOC-DIR is a directory in which all files are considered TOC-FILES.

#### Example:

For the commandcenter help, the following command is used:

```
cd ACS/Documents
acsJavaHelp -gen ACSCommandCenter \
  ACSCommandCenter/Acs_Command_Center_-_User_Guide.html \
  ACSCommandCenter/AcsCommandCenter-News.html
```

This is because with the default alphabetical sorting *AcsCommandCenter-News.html* would be processed before *Acs\_Command\_Center\_-\_User\_Guide.html*, and thus the *News* would appear before the *UserGuide* in the Java Help table of contents – which we don't want.

## Result Checking

To check the outcome of the generation process, you can run an ad-hoc help viewer with this command:

*NOTE: In this version, it is important that you run the command in the parent directory of the HELP-DIR (in our example this is "ACS/Documents")*

```
acsJavaHelp -view HELP-DIR
```

## Creating a help jar

When done, you will want to pack the HELP-DIR into a jarfile, so you can deploy it easily with the application.

The tool does not support you in creating that jar. But this is a trivial task:

```
jar cf MyAppHelp.jar HELP-DIR
```

## Showing the Help at runtime

You can use the following code:

```
import java.net.URL;
import alma.acs.jhelpgen.Gui;
import alma.acs.jhelpgen.Const;

[...]

protected Gui helpGui;

void bla() {
    try {
        String HELP_DIR = "ACSCommandCenter";
        URL url = getClass().getResource(HELP_DIR + "/" + Const.SET_FILENAME);
        helpGui = new Gui(url);

    } catch (Exception exc) {
        exc.printStackTrace();
    }

    helpGui.showHelpBrowser();
}
```

#### Notes:

- Obviously, you need to adjust the HELP-DIR variable to your HELP-DIR name
- Your code could check the helpset URL for null. If it is null (that is, the helpset could not be found), you could fall back to another helpset.

-- [MarcusSchilling](#) - 07 Dec 2006

## Related articles

- How can more people do development with ACS on the same machine without disturbing each other?
- Which ports are used by ACS?
- Problems connecting to ACS servers on a remote machine: bad /etc/hosts
- Why does the getComponent method of ZLegacy/ACS.ContainerServices return an object of type None?
- Why are some of my print statements not showing up in the container output section of acscommandcenter?