

Application Help

This is the planning page for the OO application help (aka online help, despite not being "online").

Current Situation and Issues

Structure

At present, the application help files are located in the source code module *helpcontent2* with the following structure:

```

helpcontent2
  helpers          these are required or helpful for building the help content,
                  some files are called by make, some others are obsolete
                  xmlhelp.dtd > the DTD for the help XML format
  source          this is the actual content
    auxiliary      this are help relevant files that are not content
                  *.tree > the definition files for the help viewer table of content
                  *.xsl > transformation scripts used to process the help files
                  $LANG/*.css language dependent files for controlling the HTML output
                  $LANG/*.cfg language dependent config files (legacy)
  text            the help content files, organized by modules (rather awkwardly, see below)

```

The help content files are in a home-grown XML format described in `helpers/xmlhelp.dtd` and are processed in various ways when the help is being built:

- when non-English versions are built, the English content inside the help files is replaced by the localized content stored in a different module.
- the help files are XSL-processed to produce the full text search index
- the help files are XSL-processed to extract the extended tips displayed in the UI
- the help files are XSL-processed to extract the keyword index

The final help set consists of sets of various files (jar, cfg, tree, xslt, database files) for each module. A module in the help does not exactly correspond to an application module (Writer, Calc, Impress) for historical and technical reasons. The `schart` module is present in the help although there is no standalone Chart application. There is no `sdatabase` module, although there is a Database application. So: it's a mess.

The biggest module is "shared" that takes all content that is valid for more than one application (which is most of the content) in order to make it available to the help viewer in all applications. But this leads to awkward side effects when content is valid for more than one but not all applications, yet it is still available for all applications. This leads to situations where the users finds content that is actually not in scope for his/her current context (but still visible since in shared) which can be pretty confusing.

Implementation with the application

The **help viewer** is a Writer/Web component wrapped in a **nice** UI and served by the help content provider. The appearance of the help UI is completely outdated, and so are its concepts (to separate between TOC, Index, Search, for example), and the fact that it's just a Writer document window in disguise can lead to all sorts of **funny** nasty side-effects. Also, the Writer/Web component has a hard time even displaying HTML 3.2 correctly, so the help files breathe the spirit of 1985. The full text search does not work correctly and never has.

UI Issues

The content and structure of the help files do not show a clear distinction between referential, conceptual and instructional information. On looking for guidance, users can stumble over files that just list the UI elements of a dialog with one explanatory sentence but with no indication of the context.

Maintenance Issues

- A rather complicated architecture (understatement!) in terms of maintenance. A document describing details of the inline (application) help can be found at the [Online Help](#) documentation area. The provided documentation doesn't directly explain how to assign new ids to help items, where to put them with respect to help file assignments, etc., given the hierarchical nature of the help system.
- New volunteers desiring to get involved with authoring or changing any help item would likely need to install the "Help Authoring" extension, which can be found at: <http://people.apache.org/~jsc/test/helpauthoring.oxt>. Once this extension is installed, adding or changing help texts is a bit easier but followup maintenance using Perl scripts found in the repository are needed for adding new help texts. The other alternative is to directly edit the help files (XML).
- Since current help system is stored **within** the source repository, anyone needing to make changes must be a project committer. This seems rather burdensome to new volunteers who would like to participate in this area.
- Because the Online Help is "homegrown" and difficult to fathom, it will always need special attention. So, it is not very sustainable in its current form.

Moving Forward

Since we're off to a fresh start, this is what we ought to do with the help:
(see also the following mail thread: <http://markmail.org/message/t15lsy4cxaa3s6lx>)

Ongoing Maintenance of Current Help

Helpcontent2 in its own svn area (a copy)

- This will provide volunteers a non-trunk area to experiment with the current online help system
- The "copy" area should be able to be built on its own without the need of a complete build of AOO
- Changes can be ported directly to a volunteer's production version of AOO for review, or alternatively directly accessed through other means
- Once reviewed, changes will need to be ported back to trunk/helpcontent2 for patch submission

A Means of Analysis of Current System/ Files

Currently, there is an xslt file provided for the help system, so individual help file structural correctness can be ascertained. However, there is no overarching analysis tool available for the determining the correctness of the combination of various IDs in place and their cross references. Without some mechanism in place, adding new entries or editing existing ones is nearly impossible. Before we can move to a new system of any kind, it vital that some structural analysis be made on the existing help system. Porting the help files into an XML database in some way might help in this. This needs discussion and expertise to move forward.

Migration to a New/Different System – questions/challenges

- Current Help source files (*.xhp) are xml defined by xmlhelp.dtd (/main/xmlhelp/util) and utilize main_transform.xsl (same directory). How to migrate this content to a new system. (some preliminary testing has been done renaming *.xhp files to *.xml and using the stylesheet to ascertain what this does exactly.)
- How would current indexes be generated?
- What application changes would need to be made to accommodate this?

Help File Source Format

- move to a standard format: DITA, EPUB have been suggested
- highly desirable that new system is supported by a range of open source authoring and validation tools
- highly desirable that the Help system be a stand alone entity apart from the source build process – a separate svn tree  with appropriate permissions for that area alone
- create an easy authoring workflow to enable participation

Help Viewer

- use current technology (HTML 4+, CSS4, bells and whistles)
- don't let the user decide on which route to find info (TOC, index, search), give a search input field and let the help find the best results
- fix the search engine
- integrate the help viewer with the UI

Help Content and Structure

- Limit help content to instructions, and some basic concepts
- Show reference information ("enter your name in this field") where it's needed, on the UI, side-by-side to the element it refers to. For that, we need rich formatting in "bubbles" (extended tips).
- Reduce the amount of offline help content and allow to expand to online sources
- Allow rich media in help files