PERSONAL KNOWLEDGE BASE

Although we often think of knowledge creators as technical specialists, everyone creates knowledge, for example a personal address book, a list of sellers with whom you have had good or bad experience, a personal medical history, etc. Often this information exists only in scribbled notes or old emails. Technical disciplines demand organized documentation but it usually is produced after most of the creative effort is done and there is no organized explanation for why things were done one way instead of another. This is especially a problem in multidisciplinary efforts, whose final documentation often is in the form of discipline-specific "silos" with no reference to other areas that actually explain why something had to be done a certain way.

The most extensive knowledge base in the World is the World Wide Web, where hyperlinks enable connections without content duplication. This does not support personal or local group knowledge and web pages must be stable and considerable expertise is required to create them. The most effective personal or group knowledge base documents information as it is being created. This is constantly changing. Old information may be superseded but still have value if only as a reminder of something that doesn't work.

All modern word processors support hyperlinking but provide little help in using it and hyperlinking to other documents and non-document files does not work very well. The LinkXall library for LibreOffice addresses these problems. It uses its own Artificial Intelligence to predict what you are trying to do and automates any action that is obviously valid while warning you about mistakes and suggesting alternatives. It automates finding and fixing obsolete hyperlinks and moving hyperlinked topics, simplifying maintaining a dynamic knowledge base. It provides superior hyperlinking to other files, supporting not only relative and absolute paths but also locations defined by environment variables, simplifying rearranging a complex knowledge base and sharing it between incompatible systems.

All programs that support hyperlinking have Back and Forward functions to revisit recently visited hyperlink targets. LinkXall has a smart version of this capability. It records both the hyperlink and its target in its link points history and it doesn't duplicate any point. It also allows the reader to set a "pinpoint" for subsequently revisiting a point that is not a hyperlink.

LinkXall uses its own fundamental functions to support a hierarchical document framework, which includes bidirectional linking between table of contents and topics. It simplifies managing this by automating topic rename and level, with both local hyperlinks and those from other files automatically updated.

Hyperlinks usually target a bookmark. LinkXall automates creating a variety of visible and invisible bookmarks and uniquely supports hideable bookmarks. It assists the document creator in finding bookmarks by name or perusing them by location. It also can show all hyperlinks to a bookmark both locally and from other files.

All of LinkXall's document bookmark and hyperlink capabilities are useful for both personal and technical knowledge bases. It has additional features for documenting technical projects. It supports bidirectional hyperlinking between documents and plain text files, including Python, HTML, other scripts, and LibreOffice libraries. It also provides an Emacs library to enable this popular programmer's editor to fully participate in the knowledge base. LinkXall operates in both Linux and Windows. LibreOffice alone cannot follow a hyperlink to a specific web page bookmark (anchor) in Windows but LinkXall can do it.

The LinkXall LibreOffice library can be installed by itself but many of its unique capabilities require support files and a full installation and configuration is complicated. This can be done manually but is automated by the lxainstall.zip archive. LibreOffice must already be installed. You can unzip lxainstall.zip anywhere. Its LxaInstall.odt document describes the installation, including details of the computer in which it is opened. Macros embedded in the document with help from OS-specific scripts in the archive, automatically install and configure as much of the total package as you want. For example, you can choose whether to install support for Emacs and source code and project management files for working on LinkXall itself.

LxaInstall.odt not only installs LinkXall but can completely remove it with no lingering effects. Consequently, you can install it just to see what it is like, but you might first want to study it in more detail. The doc folder in lxainstall.zip contains LxaUserGuide.odt and LxaTut.odt, which you can read directly in the archive without unzipping it. Both of these documents are examples of LinkXall's document framework. Even without LinkXall being installed, the local hyperlinks work (by Ctrl-click) and can give you a sense of the value of the framework for a document that could be part of a personal knowledge base. However, none of LinkXall's unique capabilities will be available. LxaUserGuide is a very large document, containing both practical and theoretical information. LxaTut is a relatively short tutorial, which guides experiments with LinkXall. You might want to look at this to get a sense of how much effort would be needed to learn how to use all of LinkXall's capabilities. However, none of the experiments will work until LinkXall is installed.