LANGUAGE DOCUMENTATION & CONSERVATION

Vol. 6 (2012), pp. 282-291
http://nflrc.hawaii.edu/ldc/
http://hdl.handle.net/10125/4570

Wunderkammer Import Package: A Tool for the Display of Multimedia Dictionaries on Mobile Phones

From Project for Free Electronic Dictionaries

Reviewed by Clair Hill, *University of Leuven, Belgium and Max Planck Institute for Psycholinguistics, Nijmegen, Netherlands*

1. INTRODUCTION. Wunderkammer (henceforth Wk), a tool for the adaptation and display of multimedia electronic dictionaries on mobile phones, is an open-source software package developed in 2008 by James McElvenny and colleagues under the rubric of the Project for Free Electronic Dictionaries (PFED: www.pfed.info/).

Wk was predominantly developed for the creation of mobile phone-compatible dictionaries for small endangered languages, particularly in Aboriginal Australia. Kaurna in South Australia and Wagiman in Northern Territory are two key languages for which dictionaries were produced in the development phase of Wk. These are language reclamation and language maintenance situations where there is much community need for resources that make traditional languages accessible on a daily basis. It is the concerns of these types of language situations which informed the development of this software: linguists involved in PFED wanted to take advantage of the spread of mobile phone technology into these communities as a medium for accessible multimedia dictionaries.

This review includes general information about the software and its capabilities, as well as a discussion of the creation and distribution of a Wk dictionary for Umpila and Kuuku Ya'u, two dialects of a moribund language from the northeast coast of Australia. I will provide a workflow overview based on my experience using the Wk Import Package (section 3), followed by some discussion of the distribution of—and the end-user response to—the resulting Umpila/Kuuku Ya'u Wk dictionary (section 4).

2. A WUNDERKAMMER DICTIONARY. Wk is a tool for converting an existing digital dictionary file into a Java-based format for use on a mobile phone. This software does not assist with initial dictionary creation; rather, it imports and converts a Shoebox/Toolbox dictionary file or an XML marked-up dictionary file.[1]

[1] Use of Wk Import Package is not limited to a specific XML structure. Wk developers specifically envisioned porting from an XML dictionary file prepared for use with Kirrkirr software (www-nlp.stanford.edu/Kirrkirr/). KirrKirr is a dictionary presentation tool that, like Wk, has largely been developed using Aboriginal Australian languages, and it can be used with any XML dictionary file with just a few formatting specifications. For more on formatting for compatibility with Kirrkirr, and hence Wk, see www-nlp.stanford.edu/Kirrkirr/dictionaries/xmldictionary.html.

E-ISSN 1934-5275

Wk creates a simple dictionary with the following features:

- "Flat" entry structure with no hierarchy below the entry level (i.e., subentries are not hierarchically organized)
- Nine possible fields per entry in addition to the headword
- Ability to attach one sound file and any number of images to each entry
- Linked entries
- Creation of any number of customizable menus and sub-menus to suit the user's purposes;
- Headword search tool

PFED developed Wk in 2008, with a focus on the most commonly used mobile technology available at the time. In addition to a wide range of lower-specification phones of this era through to current models, I have successfully run a Wk dictionary on a number of phones using the Symbian[2] (Nokia smartphone) operating system, as well as on phones using the Android[3] operating system. While Wk does not utilize the functionality now available on smartphones,[4] it worked on the Android phones' touch interface without any modifications—only the search functions were unusable because the phones had no physical keypad.

3. CREATING THE DICTIONARY.

3.1 PREPARATION OF DICTIONARY FILES. The Wk Import Package can create a Java dictionary file from a Shoebox/Toolbox or XML dictionary file (e.g., a file prepared for use with Kirrkirr: www-nlp.stanford.edu/kirrkirr/) with very little transformation or preparatory work. The dictionary creator can produce a Wk dictionary within several minutes of downloading the installer. That said, the creator may prefer to spend some time reconfiguring the existing dictionary for optimal display on the mobile phone medium. Three considerations for adaptation are 1) limited fields and lack of hierarchy in dictionary entry structure, 2) phone screen size, and 3) inclusion of multimedia elements. I will comment on each briefly.

Regarding entry structure, Wk has seven standardized fields onto which the existing dictionary is mapped: *lemma* (headword); *sd* (semantic domain); *pos* (part of speech); *glossdef* (gloss or definition); *sound* (link to sound file); *image* (link to image file); and *link* (link to another entry). There are three additional fields with no conventionalized association that can be customized by the dictionary creator—note data in these fields will be treated as plain text. Fields in the source dictionary can easily be excluded in the Wk conversion process. The entry structure has no hierarchy below the entry level, thus there is no easy way to include multiple senses within a single entry or to link subsections of one entry to those of another. If the source dictionary contains multiple senses, the best strategy for porting to Wk format is to convert multiple senses into separate homonymous entries.

---

[2] en.wikipedia.org/wiki/Symbian

[3] en.wikipedia.org/wiki/Android_operating_system

[4] The *iPhone* does not support *Java* applications of any kind.

As for phone screen size, dictionary creators may need to tailor entry content to fit the restricted space, limiting the need to scroll down to view the entire entry. Long definitions or long illustrative examples will quickly fill smaller phone screens.

With regard to adding multimedia content, an important consideration is balancing file size and phone storage capacity: long, numerous, or uncompressed sound and image files may render the application unusable on some phones. Thus, with smaller overall file size, more phone models are able to run the dictionary file.

3.2 WUNDERKAMMER IMPORT PROCESS. To assist the dictionary creator with the import process, there is a helpful user's guide bundled with the software, which is also available from the PFED website (w). Because it is also helpful to see some standard configuration choices as users familiarize themselves with the software interface, I point prospective users to two samples of import configurations, which are available in the program bundle: one was used in the conversion of an XML dictionary (Kaurna language), the other a Shoebox/Toolbox dictionary (Tura language). In addition, a blog hosted on the PFED site provides accounts of dictionary creators' experiences in producing and disseminating Wk dictionaries.

Wk Import software has a simple and clean interface that leads the user through the process of converting an existing dictionary to Java format for use on a mobile phone. The conversion process is straightforward: it requires the user to enter basic information about the existing dictionary, as well as links to existing files, and then make decisions about the entry and menu structure of the Wk dictionary to be produced. This process requires average computer literacy and some basic understanding of dictionary structure and development practices[5] (the user's guide presumes both)—that is, knowledge of the nature of the entry structure and fields in the existing dictionary, and how the user wishes these to be presented in the Wk dictionary. Users do not need any knowledge of Java programming language, nor do they interact with it directly in the conversion process.

The Wk Import interface consists of four tabs at the top left corner of the screen, which constitute the four main steps in the conversion process: *Input/Output* (shown in Figure 1), *Mapping, Menu,* and *Console*. Below, I cover the main steps involved; I direct the reader to the Wk documentation itself for further explanation, including illustrative visuals.

The *Input/Output* tab contains fields for information about the source dictionary and output files, including directory pathways to source-dictionary and media files, as well as basic output details like the dictionary name and version. It is here that links to icons and themes are created to control the Wk dictionary's appearance. Dictionary creators can either select one of two standard themes or create customized themes: the standard options include a simple white background with black text, and a black screen with yellow text and a frog icon watermark. I have not attempted to customize a new theme myself; customization requires working in a separate Java program (*ResourceEdit* tool, available from the Lightweight User Interface Toolkit library at www.oracle.com/technetwork/java/javame/javamobile/download/lwuit/index.html).

[5] References on the practice of dictionary making, especially for undescribed languages, include Austin (2003), Frawley, Hill & Munro (2002), Haiman (1980), and Simpson (1993).

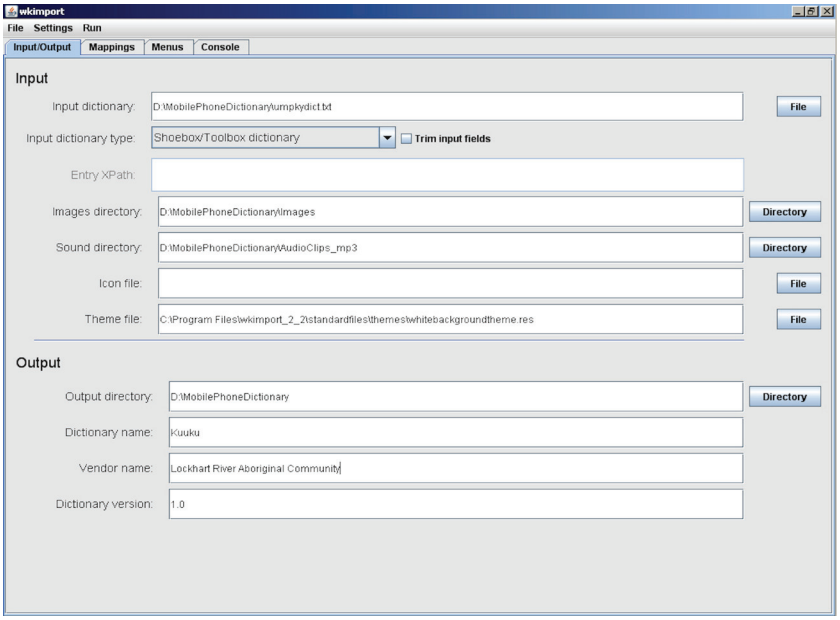

FIGURE 1. The Input/Output tab

The *Mapping* tab is the second stage of the Wk conversion process. The top half of the interface is where the dictionary creator maps the fields in the source dictionary onto the predesignated output labels for the Wk dictionary. The bottom half allows the creator to organize the ordering of fields in the dictionary entry.

The third tab, *Menu,* prompts the user to set up the dictionary entry screen and menu tree structure through which the end-user will navigate—for instance, a menu that sorts entries by headword, or semantic domain, or part of speech, or some other search term. Any number of submenus can be created and produced in any field: for example, a menu might order all entries by semantic domain, with an embedded submenu that orders entries by part of speech. In the case of a bilingual dictionary, the dictionary creator can choose a unidirectional or bidirectional format—that is, one that translates only from the source to the target language (headword menu), or one that translates to and from source and target languages (with menus based on both the *lemma* and the *glossdef* field).

Once *Input/Output, Mapping,* and *Menu* settings have been entered, beginning the conversion process is a simple matter of selecting *Create dictionary* in the *Run* drop-down menu. The conversion process usually takes only a few seconds. Once the configuration is running, the interface will jump to the *Console* tab. This tab provides a real-time report with updates relevant to the conversion process. One such update involves the Wk dictionary structure's treatment of homographic entries (those that share a headword with the same form), which are not supported. Wk automatically creates unique headwords in the conversion process, differentiating homographic entries by appending a number to each one. For example, the Umpila/Kuuku Ya'u dictionary contains two entries with the headword *yikan*, representing two different senses: a plant species and a fighting spear. The import package

automatically distinguished these by renaming them *yikan 1* and *yikan 2*. The *Console* tab provides a running record of items it renames, along with any broken links between entries that may result from renaming the headword. Broken links need to be fixed manually; the *Console* tab simply lists them, and the user must return to the source dictionary files to fix errors and re-run the conversion process.

Wk also allows the dictionary creator to produce multiple configurations for a dictionary. Configurations can be saved and reused with the *Save* and *Open* commands in the *File* drop-down menu. This allows for easy tailoring of the dictionary to, for example, specific phone memory capacities by excluding multimedia elements, or to different end-user needs through quick reconfigurations of menu structures.

3.3 END PRODUCT. The conversion process produces two Java files, one with a .jar extension and the other with a .jad extension, both of which need to be transferred to the phone. The transferral and installation procedures are discussed in section 4.

To give the reader a sense of the size of the Java files produced, let's consider the Umpila/Kuuku Ya'u dictionary as an example. The Wk Umpila/Kuuku Ya'u dictionary I produced is concise, consisting of 455 entries—it was produced as a companion resource to a series of guide booklets for language learners. I created three versions, each for use on phones with differing storage capacities or installation memory thresholds. Two versions include sound files (all in MP3 format), and none of the versions distributed to end-users include image files. A version with 144 sound files uses 2.497 KB, and one with 200 sound files uses 3.366 KB. The version without any sound files requires a mere 367 KB.[6]

Figures 2–4 illustrate the Wk dictionary interface.[7]

[6] To get a better idea of how a Wk dictionary looks and functions, the PFED website provides an interactive tour of Kaurna Wk dictionary: http://www.pfed.info/wksite/onlinedemo/onlinedemo.html.

[7] These images come from a PC emulation of the Wk Umpila/Kuuku Ya'u Java application. Java files, like Wk dictionary output files, are specifically made for the mobile phone environment and do not run on standard PCs. This means that the Wk dictionary is created in a PC environment, but cannot be viewed there. Users who wish to view Java files on a PC, which is not a necessary part of the Wk workflow, will likely need to install a Java emulator. One option is the Java ME emulator, which is available for download at http://java.sun.com/javame/sdk/index.jsp.

FIGURE 2: Main menu screen in the Umpila/Kuuku Ya'u dictionary. Shown: three menus oriented to, respectively, Kuuku headwords (top), English glosses (center), and semantic domains (bottom).

FIGURE 3: Initial screen display in the Umpila/Kuuku Ya'u headword submenu. The user scrolls down or uses the search box to move through the list. Speaker and image icons indicate entries including media files.

FIGURE 4. Entry-level structure for *aachinya* 'burn, set alight', with an image file (shown: pandanus leaves being heated in preparation for weaving) and an attached sound file. The user hears sound files by clicking the button below Play (at bottom right of screen display). This is the lowest level in the Wk menu structure hierarchy.

4. WUNDERKAMMER IN THE COMMUNITY. In this section I discuss my experiences in distributing the Umpila/Kuuku Ya'u Wk dictionary in Lockhart River, an Aboriginal community on the far northeast coast of Australia.

4.1 DISTRIBUTION AND INSTALLATION. The variation in both individual phones and mobile service providers can lead to vastly different installation experiences for both the dictionary creator and end-users. While Wk dictionary files can run on a very wide variety of phone types, the cost of this flexibility is the potential introduction of challenges to installation as a result of two factors: 1) the Wk application file is not built around the capacity and features of a particular mobile operating system, and 2) it is an impossible task for Wk developers to provide detailed troubleshooting advice given the vast differences in

end-users' phones, meaning each user is somewhat alone in working through any installation issues. This section comments on the difficulties I experienced in the distribution and installation of the Umpila/Kuuku Ya'u dictionary, and then offers suggestions for how to work around them.

The process of transferring the compiled Java files from computer to mobile phone is, in theory, quite easy. The simplest method is to connect the phone to a computer via a USB cable and copy the files using the computer's file explorer. For devices that accommodate auxiliary memory, files can be transferred to a memory card. An alternate method is to transfer the files wirelessly, via a Bluetooth-enabled computer or mobile phone. Once the files have been copied and accessed with the mobile phone, the dictionary will self-install.

Unfortunately, transferring the Umpila/Kuuku Ya'u dictionary was not so straightforward: neither of the above methods worked, which greatly increased the difficulty of dissemination between end-users in the community (thus restricting the possibility of it being rapidly and "organically" disseminated from end-user to end-user without orchestration). Lockhart River community, where Umpila/Kuuku Ya'u people reside, is in a remote part of the northeastern coast of Australia, and its only mobile service provider has locked phones to prevent customers from accessing Java files that are transferred via USB or Bluetooth as a means of controlling piracy. Few mobile providers have such policies, and it is unfortunate that this community is reliant on a provider that does. My work-around was to host the Wk Java file on a website and use the phones' internet browsers to download the file.[8]

Once transferred to the phone, file installation also presented some challenges. The wide variety of phones with different operating systems and of different generations in the Umpila/Kuuku Ya'u community at times created barriers to installation. A recurring issue encountered here was that some phone models had inadequate memory to run the installation, a separate matter from the phones' available storage. This was overcome by recompiling a smaller version of the dictionary for phones with this issue. Of the phones I encountered with this restriction, the dictionary installed successfully only at a size smaller than 3 MB. In remote language communities like Lockhart River, where end-users may have older-generation or low-specification phones, I recommend producing several versions of a Wk dictionary with different file sizes so as to ensure that some version of the dictionary will work on every phone type.

One interesting aspect of the challenging installation process was that it became a learning opportunity between myself and the end-user. Installing a Java program was not something any of the end-users had done before (nor did I have much experience, for that matter), and the installation process forced them to navigate menu structures on their phones that they did not normally use. Technological issues such as this could pose a barrier to dissemination of a Wk dictionary in some language communities, if it is attempted without the help of a somewhat tech-savvy person.

---

[8] I discuss a few other aspects of this workaround on the PFED blog at http://pfed.info/2011/04/27/dictionary-distribution-a-workaround-for-bluetooth-block/.

4.2 END-USER FEEDBACK. End-users in Lockhart River were extremely positive about the Umpila/Kuuku Ya'u mobile phone dictionary. Once the obstacles to installation were overcome, users found it an exciting new medium for language learning and a user-friendly resource.

One positive outcome during distribution was that the dictionary stimulated intergenerational use and language instruction. Older and younger generations naturally took on different roles while interacting with the resource as a result of their complementary asymmetry in linguistic and technological experience. Younger people operated the dictionary and showed older, typically less tech-fluent community members how the application worked. In turn, the older community members elaborated on word definitions, repeated entry words, and commented on related word forms. In effect, the older generation brought the dictionary "to life."

Other key positive features include the following—some of which I observed, and others commented on by end-users:

- Simple, "flat" structure: The dictionary is easy to navigate, and the simple internal structure makes it accessible for less tech-fluent users—including those who are unfamiliar with navigating hierarchical file structures.

- Mobility: Wk is a go-everywhere tool that significantly increases everyday language accessibility and exposure for the end-user. As Umpila/Kuuku Ya'u is a moribund language with limited resources and no substantial dictionary, this application has dramatically increased users' day-to-day access to the traditional language.

- Durability: Many language resources in the form of books, CDs, or DVDs quickly become damaged or lost in the community environment. A phone offers high durability as a host, and electronic files are much easier and cheaper to replace than hard-copy materials.

- Broad appeal: The medium was deemed exciting and novel by a broad demographic within the community. It particularly appealed to younger generations, who already use mobile phones as portable entertainment and information dissemination devices (music, image, and video capture and storage). Engaging young people with traditional language materials is an important goal for speakers and community elders, and an important key to increasing intergenerational language transmission.

5. FINAL COMMENTS. The power of the Wk Import Package is that it expands the accessibility and usage of existing dictionary files. It takes traditional dictionary file formats that are used by linguists and language workers and, with a small amount of additional effort, converts these into a mobile product for presentation in a new medium.

In practice, Wk is a straightforward interface that leads the dictionary creator through the conversion process. On the part of the dictionary creatorw, it requires average computer literacy and some basic understanding of dictionary structure at both the menu and entry

levels. For the end-user, the import package produces a simply structured dictionary, with a limited number of entry fields and absence of hierarchy below the entry level. It is this simplicity that maximizes usability, both in terms of access for less fluent technology users, and in Wk's aforementioned compatibility with multiple phone types (from low-specification phones to "feature" phones and smartphones).

Another of this resource's pros is that, as a Java application, it is not reliant on a specific operating system: it can be compiled using multiple computer operating systems and run on multiple mobile operating systems. It is a great achievement that Wk developers have created such a useful, featureful dictionary that can run on a wide range of phones. These very strengths, however, could also be viewed as potential weaknesses: the tradeoffs are that Wk does not take advantage of the extra functionality of newer "feature" phones, and its design for use across many phone types can introduce some installation difficulties (as discussed above, in section 4). The latter tradeoff may necessitate creative workarounds such as those I described, but it also creates a unique opportunity for collaborative learning with speakers and community members as users negotiate the challenges together.

Overall, in my experience while working with one Australian Aboriginal language community, a Wk dictionary—and the mobile phone medium, more broadly—was extremely well-received as a tool for language transmission.

Primary function:	Conversion of electronic dictionary files to Java format for use on mobile phones.
Pros:	Cross-platform, user-friendly software that rapidly converts existing standard dictionary file types to Java format. Allows for multimedia elements and produces a dictionary app that will run on a wide range of phone types.
Cons:	Limitations in dictionary structure (this was a pro in production of a dictionary for a moribund Australian language, but for other language projects the simple dictionary structure could be too restrictive). Installation on some phone types is not always straightforward, and may be incompatible with service provider practices.
Platforms:	Java-enabled platforms (Windows, Mac, Unix)
Open Source:	Yes
Proprietary:	No
Available from:	Project for Free Electronic Dictionaries: http://www.pfed.info/wksite/
Cost:	Free
Reviewed Version:	wkimport package 2.2
Application Size:	9.5 MB
Documentation:	http://www.pfed.info/wksite/readme.html

References

Austin, Peter. 2003. Australian Aboriginal Lexicography. In R.R.K. Hartman (ed.), *Lexicography: Critical Concepts in Linguistics*, 288–294. London: Routledge.

Frawley, William & Kenneth C. Hill. 2002. *Making dictionaries: Preserving indigenous languages of the Americas*. Berkeley: University of California Press.

Haiman, John. 1980. *Dictionaries and encylopedias*. *Lingua* 50. 329–357.

Simpson, Jane. 1993. *Making dictionaries*. In Michael Walsh & Colin Yallop (eds.), *Language and culture in Aboriginal Australia*, 123–144. Canberra: Aboriginal Studies Press.

Clair Hill
clair.hill@mpi.nl
clair.hill@arts.kuleuven.be