Program Flow in readJDX

Bryan A. Hanson[a]

[a]Dept. of Chemistry & Biochemistry, DePauw University; hanson@depauw.edu

This version was compiled on November 12, 2019

This vignette is based on `readJDX` version 0.4.29.

Program Flow

`readJDX` is coded in such a way that it should be easy to add features. Contributions to improve or expand the package, including pull requests, are always welcome! Table 1 shows the overall flow of the function calls. Only a couple of these functions are exported, so take a look at the source code for documentation. Be sure to check out the *MiniDIFDUP_1* and *MiniDIFDUP_2* vignettes for additional information about the JCAMP-DX file structure and how `readJDX` functions extract the data.

Table 1. Program Flow.

function	input
readJDX	file name
- findVariableLists	character vector: all lines from original file
- extractParams	character vector: just the metadata lines
- processVariableList	character vector: a single VL
- - decompressXYY	character vector: a single VL
- - - decompLines	character vector: a single VL
- - - - getComp	character vector: a single VL
- - - - unSQZ	character vector: named with the ASDF mode, called via lapply
- - - - insertDUPs	character vector: named with the ASDF mode, called via lapply
- - - - - repDUPs	a string of length one
- - - - deDIF	*list* of character vectors from a single VL; the character vectors are named with the ASDF mode, the list is named with line numbers
- - - - - unDIF	character vector: one line from VL, entries named by ASDF code
- - - - yValueCheck	*list* of character vectors from a single VL; the character vectors are named with the ASDF mode, the list is named with line numbers

Note:
VL stands for variable list, as defined in the JCAMP-DX standard. For examples see the *MiniDIFDUP_1* vignette.