LoadDB

COLLABORATORS			
	TITLE : LoadDB		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		October 27, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

LoadDB

1.1 LoadDB - Create A-Kwic databases

```
                              LoadDB

              A program to generate A-Kwic databases


                            Written by
                         David W. Lowrey

                           Produced by
                      Starbound Enterprises


              Copyright 1993,1994 by Starbound Enterprises

                           Version 2.0

                    Introduction
                    Running LoadDB
                    Keywords
                    LoadDB input format
                    Appendix
```

1.2 Introduction

```
INTRODUCTION -

LoadDB, the database loader, creates a database from a set of input records.

The general format of a record is:

 #R  (One "Record" header per record)
 #H title (The Title for this record)
 #K keyword [keyword] [...] (keyword records)
 #T text  (text that is NOT to be scanned for keywords)
 text (free form text that is to be scanned for keywords)
```

LoadDB reads these records, and produces an A-Kwic database. You then use the
A-Kwic program to search the database for records that match user specified
keywords.

1.3 Running LoadDB

Running LoadDB -

LoadDB can be run either from the CLI, or from the WorkBench.

CLI usage:

The syntax of the LoadDB command is: (options in [] are optional)

```
  LoadDB INPUTFILE ifile DATABASE dbname [KEYLIST] [KWSIZE size] \
         [MAXKW num] [LINELEN len] [TRIVIAL fname]
```

```
 Where:
        ifile   = The name of the file containing the input records.
                  This parameter is required. You must specify the
                  "INPUTFILE" keyword, along with it's value

        dbname  = The data base name. Note that this is actually the
                  "stem" of the database name. If you specify a database
                  name of "foo", then the actual files created by
                  LoadDB will be named "foo.bitmap", "foo.index",
                  "foo.key", "foo.record", and "foo.text".

        KEYLIST = If this optional keyword is specified, then LoadDB
                  will print out a (probably very long) list of keywords
                  it found, along with some generally useless statistics.
                  This list can be useful for determining appropriate
                  "trivial" keywords (see below). The list can be written
                  to a file by redirecting LoadDB's standard output via the
                  '> filename' specifier on the command line.

        size    = This optional parameter specifies the number of
                  significant characters in keywords. Any characters
                  beyond this number are ignored. So, if KWSIZE=5, the
                  keyword 'ABCDEFG' will match 'ABCDE', 'ABCDEQW', and
                  'ABCDE43'. It will not match 'ABCD' or 'ABCD1'.
                  The default value is 16

        num     = This optional parameter specifies the maximum number of
                  keywords per record. The default is 1000. You will need
                  to change this if you are short of memory, or if you
                  could have more than 1000 keywords per record. Note that
                  this is the total number of keyowrds, not UNIQUE keywords.
                  So a record with 800 words, all the same, has 800 keywords.
                  This setting does not effect the size of the database, so
                  most users will not need to change it.

        size    = The length of the largest line in the input file.
                  This is present only to conserve memory. The default
```

                        length of 255 should be adequate for most users. This
                        parameter does not effect the size of the database

             fname   = The name of a file containing "trivial" words. Trivial
                        words are words, such as 'and' and 'the', that are not
                        to be considered keywords.
                        The format of the file is free form text. The text lines
                        are scanned for keywords, just as the input records are.
                        You can have zero, one, or more keywords per line. The
                        length of the line should not exceed the value of the
                        LINELEN parameter.

   Example:

       LoadDB INPUTFILE input.data DATABASE db:customer KWSIZE 10 LINELEN 90

          This will scan the input file 'input.data' for records. The
          database file names will be in the db: directory, and will
          start with the name 'customer'. Keywords will have 10 significant
          characters, and the largest input line is 90 characters.

WorkBench usage:

  LoadDB can be launched from the WorkBench via a tool or project Icon.

  Parameters are passed as ToolTypes within the Icon. If a "project" Icon
  is used, then the ToolTypes will be read from it, and not LoadDB's Icon.

  See the CLI syntax for a description of the parameters.

  LoadDB opens a console window to display messages. To remove this window,
  click on it's close gadget in the upper left hand corner.

1.4 Keywords

LoadDB's job is to create a list of keywords that are in a database, and
pointers to the records that contain those keywords.

A Keyword is an alphanumeric string of characters, at least two characters
long. Single digit numbers are also valid keywords. Case is not significant.

The following characters are valid in a keyword:
    0-9 a-z -./_

Only a certain number of characters in a keyword are significant. That
number is specified when a database is created, via the KWSIZE parameter.

Example: If 4 letters are significant, then the keyword "ABCD" will match the
         following words: ABCD ABCD123 ABCDEFG

1.5 LoadDB Input Format

LoadDB accepts a standard input format, so most users will have to create
a program that puts their data into this format.

For most cases, this is fairly easy, and can be done using ARexx, C, or
even AmigaDOS script files.

The general format of a record is:

```
 #R  (One "Record" header per record)
 #H title (The Title for this record)
 #K keyword [keyword] [...] (keyword records)
 #T text  (text that is NOT to be scanned for keywords)
 text (free form text that is to be scanned for keywords)
```

The #R and #H records are mandatory.

#R Header:

 There has to be one #R header per record. The current record is ended when
 the next #R header is found. Any text on this line is ignored.

#H Header:

 There should be one #H header per record. The #H header contains the title
 of the record. This is the title displayed by the A-Kwic program when
 a keyword has matched a keyword in this record. Note that the title
 contained in this header is NOT scanned for keywords. The #H header can
 occur anywhere in the record.

#K Header:

 The optional #K header allows you to add additional keywords to this record.
 An example would be to add the date a record was created. This record does
 not show up when the record is viewed. You can have multiple #K headers
 in a record.

#T Header:

 The optional #T header allows text to be placed in the record that will NOT
 be scanned for keywords.

text:

 This is just free form text, that will be scanned for keywords, and
 added to the record.

Example:

```
#R
#H The First Record's Title
#K Addedkw1
This is record 1. It is the first record, not number 200
#R
#H The Second Record's Title
#K Addedkw2
```

#T This will be in the record, but not keyworded.
This is also a record. It is record number two! But, who cares what number
it is because record numbers are not presented to the A-Kwic user.

By using the #T and the #K headers, it is possible to exactly
control the keywords for each record, if you wish.

Example:

#R
#H Customer #543 - Incident #10
#K cust/543 inc/10 odate/10jan94 status/cl workq/dwl1 udate/11jan94
#K A-Kwic help function doesn't work
#T Problem: When the customer clicked on the help button, no help appeared
#T Resolution: Make sure that the AmigaGuide program is in WB's path

In this case, only the #K headers will provide keywords, while the text
on the #T headers will not be scanned, but will be displayed in the record.

1.6 Appendix

Resources

  LoadDB tries to be as efficient as possible when it comes to memory
  usage.

  The amount of disk space used depends on the number of keys that are
  in the database. There are 5 files created for each database. This allows
  most data to reside on disk, but allows rapid access by the A-Kwic
  program.

  The number of significant characters in a keyword can effect the amount
  of disk space. Use the KWSIZE to change this from the default of 16.

  The following statistics were from a run where a database was created
  using the content description of 930 Fish disks as input. Each program
  on a disk was a database record:

   Original disk listing size: 1896015 bytes
   LoadDB input file size: 1893080 bytes
   Number of records: 4353
   Length of run (on a 68020, 16mhz Amiga): 40 Minutes
   Maximum amount of memory used: 1.7 meg
   Number of keywords: 20098
   Total size of database files: 2710824 bytes

  Keep in mind that the A-Kwic program, which is used to search
  the database, uses very little memory, and is very fast. An investment
  on the front end can yield excellent results for the end user.

The author may be reached at:
   Internet: dwl10@juts.ccc.amdahl.com
   BIX: dlowrey

```
    US Mail:
    Dave Lowrey
    Starbound Enterprises
    7391 Hosbrook Road
    Cincinnati, OH 45243
```

This program is freely distributable for non commercial uses.
Commercial usage, such as use on a "for profit" CD-ROM, requires prior
approval from Starbound Enterprises. Contact the author for details.

The author retains all rights to the code and concepts contained
within the code.

Credit:

    The following products were used to create the A-Kwic system:

```
        SAS/C Version 6.51 - the SAS Institute
        GadToolsBox        - Jan van den Baard
        Flex               - Amiga port by Anders Hybertz
        YACC               - Amiga port by William Loftus (I think)
        CygnusEd Editor    - ASDG
        RCS                - Amiga port by Heinz Wrobel
```

    Many thanks to the folks on BIX, who answered my many questions regarding
    C programming. The more you think you know about C, the more you realize
    what you don't yet know!

 To Do:

    o ARexx - This will require a re-design of the A-Kwic system from a
              "stated" to a "non-stated" system. This will also require
              (possibly) a re-design of the GUI.

Custom Code:

    If you or your company think that A-Kwic could be of use to you in
    your business, the Author would be glad to talk with you regarding
    customization of the code for your specific purposes.

Any comments or suggestions would be welcome!