

The Un-Standard Library

Introducing the un-standard library for Liberty BASIC.

This project is an attempt to create a code library to aid in general software development. It is composed of individual modules that can, for the most part, stand on their own. You only need to learn how to use the functions in the module you need to use. Guidelines on contributing to unstdlib will be created when the project's first version is released.

How to Use unstdlib in Your Project

Using UNSTDLIB in your project is as easy as opening "unstdlib.bas" and saving it as a different filename wherever you like. All you have to do is insert your code in between the section with the global variables and the functions in the library.

When Liberty BASIC 5 arrives, the UNSTDLIB will be ported over to the new LB5 syntax, and will become much easier to interface with as a result.

Module Documentation

Below is the documentation for the modules contained within the UNSTDLIB.

List Module

The list module stores data using a key-value system. This isn't an incredibly fast method, but it does work quite well and allows for a lot of flexibility in data storage. The only limit to how much can be stored is determined by the maximum string length in Liberty BASIC, which appears to be several gigabytes worth of data.

Here's a code example using the list module.

```
items = 7      ' Number of items to make.
myList$ = ""   ' List with values.
keys$ = ""     ' List to hold the keys in myList$.
keys = 0       ' Number of keys in myList$.
For i = 1 To items
    Call List.Set myList$, "key ";i, "Entry ";i
Next i
' Get the list of keys now:
```

```
keys = List.Keys(myList$, keys$)
For i = 1 To keys
    keyName$ = List.Get$(keys$, Str$(i))
    Print "  Key ";i;": ";keyName$, "Data: "; List.Get$(myList$,
    keyName$)
Next i
```