6834f3d0-0

COLLABORATORS

	TITLE : 6834f3d0-0	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 7, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

6834f3d0-0

1.1 StatsFuncs.bb2

```
                Statements & Functions for Blitz Basic 2

Compiled by James L Boyd -  jamesboyd@all-hail.freeserve.co.uk


               Introduction

               Beginners

               Contact
```

1.2 Introduction

```
Introduction

This file is a huge set of useful statements and functions
for Blitz Basic 2.

Some of them perform operations unavailable in Blitz, while
others are intended as replacements for existing commands,
which will often reduce the size of your program (since it
doesn't have to include 3rd party libraries - yes, Blitz
includes the FULL library of any command you use!), and
sometimes the replacement is simply better (eg fixes a bug).

You'll find instructions and demos in the source file.

For a full list and brief descriptions, run the ShowStats&Funcs
program supplied (note that if you move it elsewhere, you must
also copy the (hidden) showsf.asc file to the same place).

I'll probably rewrite ShowStats&Funcs (and rename it ;) so that
it uses listviews for easier "browsing". Soon, honest :)

Please read the notes at the top of the file before sending
```

```
bug reports :)

I did quite a lot of these, but not all of them. If you did
any of these, let me know and I'll credit you!

Oh, yeah, I've tested EVERY SINGLE THING in this file!
It all works here (though apparently some stuff won't work on
systems with < OS2.0, but that's not my problem ;)

Um, that's it...go away.
```

1.3 Contact

```
Contact

This file was put together by James L Boyd :

  jamesboyd@all-hail.freeserve.co.uk

Any comments or contributions are welcome :)
```

1.4 Beginners

```
  Beginners

  If you've never used statements or functions, here's a quick
  and easy guide as to how to use the stuff in the
  statements&functions.asc file.

  Format

  I've arranged everything in the file according to this
  format :

  .Label - the name which appears in Blitz's label list (at
           the right hand side of TED/SuperTED).

  Statement/Function - tells you the name and parameters
                       required.

  Author - who made the statement or function.

  Brief description and notes.

  The statement or function itself.

  A short demo.

  General Usage

  Just cut and paste the statement(s) or function(s) you need in
  your program, and place them somewhere near the top (anywhere,
```

actually, as long as your program doesn't try to call them before they've been defined).

Cut from the tokenised (highlighted, usually blue) "Statement" or "Function" command up to the tokenised "End Statement" or "End Function" command, then paste into your program.

Usage (Statements)

All you do here is type the name of the statement, with parameters required (if any) in curly brackets {}. Note that statements which don't need any parameters still need brackets after them.

eg. Statement : BFWindow { window }

To put a backfill into a window (white, like reqtools requesters) you'd just write :

BFWindow {0}

Where 0 is the number of the window.

Just fill in any extra parameters as specified in the description of the statement.

Usage (Functions)

There are several ways to use functions.

Generally, functions are intended to return a value to the program, often to check whether the function has been successful in whatever it's doing :

eg. Function : Del { file }

This function tries to delete a file, and returns a value depending on whether it managed to delete it or not : 0 (or False) if it failed to delete the file, or -1 (True) if it was successful, so as an example :

```
If Del {"s:startup-sequence"}=True
  Request "","I've just deleted my startup-sequence!","Doh!"
EndIf
```

Some functions are intended to return a value, and you can use them like any other variable :

Function : WBColours {}

This function tells you how many colours are in the Workbench, and you could report it to the user like this :

```
Print "You have a "+WBColours{}+" colour Workbench!"
```

The easiest way to learn how to use them is just to go through the demos in the source file, uncommenting them

```
(removing the starting semi-colons) individually, compiling
and running the source, and re-commenting each one after
trying it.
```