

DFBFlg

COLLABORATORS

	<i>TITLE :</i> DFBFlg		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 5, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	DFBFlg	1
1.1	DFBFlg Documentation	1
1.2	History	1
1.3	Copyright	1
1.4	Requirements	2
1.5	Introduction	2
1.6	Installation	2
1.7	Headers	3
1.8	How To Use	3

Chapter 1

DFBFlg

1.1 DFBFlg Documentation

Welcome to DFB's File List Generator's manual

Select on any topic to continue...

Copyright

Requirements

Introduction

Installation

How to use

History

1.2 History

Version	Date	Comment
V1.0	24/09/95	Initial release.
V1.1	31/08/96	Can now generate file listings in Amigaguide & HTML.

1.3 Copyright

DFB is copyright 1994-7 by Andrew Leppard.

DFBFlg is in the public domain. If you find a use for it without needing a registered dfb.library then lucky you! 'Nuff said.

1.4 Requirements

DFB requires WB2.04 or higher, it requires MAX's BBS running with (initially) at least one file section. It's recommended that MAX's BBSs running DFB should have at least 1.5 megs of memory. Also at least an 8 colour workbench is recommended for viewing the icons :-)

DFBFlg requires DFB to be properly set up, it also needs the DOS command append (so copy it to your sys:c/ drawer!).

DFBFlg is written for the 1.1+ releases of DFB, although almost all of its features will work with the 1.0, 1.0a, 1.0b & 1.0c libraries.

(The only feature that won't work will be the ability to change the ansi colours of the file lists.)

1.5 Introduction

Liquid Softworks is proud to present DFB's file list generator.

The wait has been immense, one of DFB's problems is the serious lack of specific utilities for it. There have been a couple of utilities released for DFB (e.g Tag list editors / deleting doors) but nothing as major as a file list generator !

When I wrote DFB I released some serious documentation on how to programme for it. One of the problems, though, was it was aimed for assembly programmers! It looks as so there aren't as many around as I thought.

Which is a pity since if you can code in assembly, DFB has to be one of the easiest doors to expand upon. Have a look at this door for instance, it is less than 9K long!! It includes a full GUI interface, DFB specific interface code, ANSI capabilities, Amigaguide output etc etc...

Anyway, I digress!

This programme will cover your entire file list generation problems. You can generate file lists in either ansi (which I have YET to see a document viewer for), ascii, Amigaguide or HTML. Generate sectional file lists (i.e: each section's file list goes into a different file), generate different file lists for sysops and users (ie select the 'access' level that the file list will be created for. Handy for remote sysops.).

This programme FULLY supports all of DFB's layout files and also the header capabilities.

1.6 Installation

Installing DFBFlg is very easy.

Copy the append command which is included in the C directory to your C: directory and you are away!

1.7 Headers

Headers are fully explained in the DFB 1.1 documentation, since however I am releasing this before DFB 1.1 I will briefly go through how to use them.

If you already have V1.1+ of DFB then ignore this section and go straight to your main manual - since if you have the full release of 1.1+ you will find that you can do a lot more with headers than what is mentioned here. (What else you can do with them requires the V1.1+ release of DFB).

Ok, so what are headers?

Well, all you need to know for this programme is that headers are ansi (or ascii if you wish) text files. When DFBFlg creates a section's file list it will first look for a header file, that is some ansi text to display at the top of the file list.

The default prefix for the headers is: dfb:headers/header

This means that if, say, DFB was creating the file list for section 25 it would look for an ansi text file under the name of dfb:headers/header.25 and it would then place it at the top of that section's file list.

So if you want the text 'My BBS's file list' at the top of your file list create a text document with those words in it and save it under dfb:headers/header.0. (Presuming section 0 is your first file section).

Since DFB sorts each file into its own section you can then label each section in your file list using the appropriate header.

Enjoy!

1.8 How To Use

Basic Use

Click on the DFBFlg icon and you should get a simple interface. The first string gadget, which is labelled FileList Name, is the name of the file list that you wish to create (simple enough :).

The second one is the header prefix, see the section of
Headers
for
more information about them.

The first cycle gadget (which is labelled Layout) controls how you wish the files to be displayed. The four different options (which are Default, Opt Layout 1, Opt Layout 2, Opt Layout 3) correspond to the boxes in the DFBPrefs programme. For more information on layouts refer to your DFB.guide.

The next cycle gadget (which is labelled Text) controls whether your

file list is created in ansi (i.e: having all those fancy colours that you see when DFB lists files on MAX's), ascii (plain text), AmigaGuide (ie a hypertext document displayed using multiview or amigaguide) or HTML (the language used to display documents on the World Wide Web).

Then there is gadget labelled 'Use access', this controls what access level DFB creates the file list for. This corresponds to the access levels you have assigned each section in DFBSecton. If you sections require 25 access to view the files then your file list should generate file lists for a user with 25 access.

If, however, you wish to generate a file list for your friendly Guam remote sysop then you can select 10000 here so he can see all those extras that sysops like to have written next to each file.

The next gadget is a tick box labelled 'Separate FileLists'. When DFB generates the file lists it ALWAYS generates them in a separate file for each different section. If this box is checked then DFB won't 'glue' the files together afterwards.

If this box IS checked and you have 'dfb:FileList' in your FileList Name gadget then the file lists will be called:
dfb:FileList.0, dfb:FileList.1, dfb:FileList.2 etc...

The Amigaguide and HTML options work slightly differently. Both options will make the lists differently if you check the sectional file lists box. This enables programmes like Multiview/Amigaguide/Netscape/Aweb etc to follow the links across files. So you cannot just paste the files back together in one document, otherwise the links won't work.

When amigaguide generates its lists in sectional form, it will put the main menu in the first section's file list. So if your first section is 23. The amigaguide menu will be at the start of dfb:FileList.23 for example.

HTML works slightly differently, when you generate separate file lists, it creates a separate file for the menu which, in this example would be dfb:FileList. Ie it does not append a section number.

It is recommended that before creating file lists you delete the old file lists.

The remaining four buttons are pretty obvious:

The Create & Quit button will generate the file list for you then quit the programme.

The Create button will generate the file list but remain in the programme.

The Save button will save your current file list settings.

The Quit button, oh please - just guess!

Advanced Use

Here's the situation: You've been using DFBFlg for about a week now, diligently generating file lists every day of the week. You start to

think, hmmm this is getting annoying, I want DFB to automatically do this for me!

Well DFBFlg has some more advanced features that can only be used from the CLI. (Don't worry workbench viewers - the CLI features are sparse and easy to use.)

Automation

You can generate a file list from the command line by typing:

```
DFBFlg create
```

Put this in your s:user-startup file and DFB will create the latest file list for you every time you boot up. Or alternatively you can use MAX's autoexecute programme to get it to run this programme once every so often (see MAX's documentation on how to do this).

This is all fine and good if you only wish to create one kind of file list, but what if you want to create both an ascii AND an ansi file list, but do not wish to have to keep changing the settings everyday?

Well, from the CLI you can specify which file you wish to load in as DFBFlg's options, e.g:

```
DFBFlg create -fdfb:optionsfile.flg
```

(If you specify the -f<optionfile> you ***MUST*** have the word create before it).

DFBFlg always saves its preferences to the file: DFB:FileListData.FLG and will always check there unless otherwise told not to.

If you wish to use multiple .FLG files then you'll have to get DFBFlg to save its settings then you'll have to copy them to different files using the CLI (or workbench).
