

FDS-310

COLLABORATORS

| | | | |
|---------------|---------------------------|-------------------|------------------|
| | <i>TITLE :</i> FDS-310 | | |
| <i>ACTION</i> | <i>NAME</i> | <i>DATE</i> | <i>SIGNATURE</i> |
| WRITTEN BY | | February 12, 2023 | |

REVISION HISTORY

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------|-------------|------|
| | | | |

Contents

| | | |
|----------|---|----------|
| 1 | FDS-310 | 1 |
| 1.1 | FDS-310 -- Copyright ©1995, Peter Bornhall | 1 |
| 1.2 | Section 0.1 -- Section list | 1 |
| 1.3 | Section 1.0 -- Preface | 3 |
| 1.4 | Section 1.1 -- Distribution | 3 |
| 1.5 | Section 1.2 -- Disclaimer | 3 |
| 1.6 | Section 2.0 -- Introduction | 3 |
| 1.7 | Section 2.1 -- Purpose | 4 |
| 1.8 | Section 2.2 -- Using the FDS | 4 |
| 1.9 | Section 2.3 -- Glossary | 4 |
| 1.10 | Section 3.0 -- Using the standard | 5 |
| 1.11 | Section 3.1 -- Creating the file description | 6 |
| 1.12 | Section 3.2 -- Using the file description | 6 |
| 1.13 | Section 3.3 -- Adding the file description to an archive | 7 |
| 1.14 | Section 3.4 -- Adding the file description to a text file | 7 |
| 1.15 | Section 3.5 -- Adding the file description to a binary file | 8 |
| 1.16 | Section 4.0 -- Format of the file description | 8 |
| 1.17 | Section 4.1 -- Components of the file description | 10 |
| 1.18 | Section 4.2 -- Default description | 11 |
| 1.19 | Section 4.3 -- Short description | 11 |
| 1.20 | Section 4.4 -- Extended description | 12 |
| 1.21 | Section 4.5 -- Sections in an extended description | 12 |
| 1.22 | Section 4.6 -- Keywords | 13 |
| 1.23 | Section 4.7 -- List of current keywords | 14 |
| 1.24 | Section 4.7.1 -- %AUT: (Author information) | 15 |
| 1.25 | Section 4.7.2 -- %CHR: (Charset used in description) | 15 |
| 1.26 | Section 4.7.3 -- %DAT: (Releasedate) | 15 |
| 1.27 | Section 4.7.4 -- %DES: (Language-specific descriptions) | 16 |
| 1.28 | Section 4.7.5 -- %DIR: (Directory) | 17 |
| 1.29 | Section 4.7.6 -- %END: (Stop parsing) | 17 |

| | | |
|------|---|----|
| 1.30 | Section 4.7.7 -- %EXT: (Extension to the %DES: keyword) | 17 |
| 1.31 | Section 4.7.8 -- %FIL: (Original filename) | 18 |
| 1.32 | Section 4.7.9 -- %LAN: (Supported languages) | 18 |
| 1.33 | Section 4.7.10 -- %LEN: (Original filelength) | 19 |
| 1.34 | Section 4.7.11 -- %OBJ: (Object keyword/category) | 19 |
| 1.35 | Section 4.7.12 -- %PID: (Program identity) | 20 |
| 1.36 | Section 4.7.13 -- %REC: (Recommended system setup) | 20 |
| 1.37 | Section 4.7.14 -- %REG: (Registration fee) | 21 |
| 1.38 | Section 4.7.15 -- %REP: (Replace files) | 21 |
| 1.39 | Section 4.7.16 -- %REQ: (Required system setup) | 22 |
| 1.40 | Section 4.7.17 -- %REV: (Revision) | 22 |
| 1.41 | Section 4.7.18 -- %TYP: (Distribution type) | 22 |
| 1.42 | Section 4.7.19 -- %VER: (Version string) | 23 |
| 1.43 | Section 4.8 -- List of languages | 23 |
| 1.44 | Section 5.0 -- Developer addendum | 24 |
| 1.45 | Section 5.1 -- Creating a file description | 26 |
| 1.46 | Section 5.2 -- Adding the description | 26 |
| 1.47 | Section 5.3 -- Extracting the description | 26 |
| 1.48 | Section 5.4 -- Parsing the description | 27 |
| 1.49 | Section 5.5 -- Keywords in general | 28 |
| 1.50 | Section 5.6 -- Keywords in specific | 28 |
| 1.51 | Section 5.7 -- Keyword recognition | 29 |
| 1.52 | Section 5.8 -- Keyword processing | 29 |
| 1.53 | Section 5.9 -- Parsing 'Old Style' descriptions | 29 |
| 1.54 | Section 5.10 -- Converting FILE_ID.DIZ | 29 |
| 1.55 | Section 5.11 -- Date formatting | 30 |
| 1.56 | Section 6.0 -- Miscellaneous | 30 |
| 1.57 | Section 6.1 -- The author | 30 |
| 1.58 | Section 6.2 -- Acknowledgments | 31 |
| 1.59 | Section 6.3 -- History | 32 |
| 1.60 | Section 6.4 -- Last words | 33 |

Chapter 1

FDS-310

1.1 FDS-310 -- Copyright ©1995, Peter Bornhall

File Description Standard

-- -- -- --

Version 3.10, Release 3
(FDS-310)

Index

Preface

Introduction

Using the standard

Format of the file description

Developer addendum

Miscellaneous

==== Endorsed by ====

USERS STANDARDS GROUP

Copyright 1995, Peter Bornhall
All rights reserved.

1.2 Section 0.1 -- Section list

| | |
|-------|-----|
| INDEX | 0.0 |
|-------|-----|

0.1 SECTION LIST

| | |
|--|--------|
| Index | 0.0 |
| Section list | 0.1 |
| Preface | 1.0 |
| Distribution | 1.1 |
| Disclaimer | 1.2 |
| Introduction | 2.0 |
| Purpose | 2.1 |
| Using the FDS | 2.2 |
| Glossary | 2.3 |
| Using the standard | 3.0 |
| Creating the file description | 3.1 |
| Using the file description | 3.2 |
| Adding the file description to an archive | 3.3 |
| Adding the file description to a text file | 3.4 |
| Adding the file description to a binary file | 3.5 |
| Format of the file description | 4.0 |
| Components of the file description | 4.1 |
| The default description | 4.2 |
| The short description | 4.3 |
| The extended description | 4.4 |
| Sections in an extended description | 4.5 |
| Keywords | 4.6 |
| List of current keywords | 4.7 |
| %AUT: (Author information) | 4.7.1 |
| %CHR: (Charset used in description) | 4.7.2 |
| %DAT: (Releasedate) | 4.7.3 |
| %DES: (Language-specific descriptions) | 4.7.4 |
| %DIR: (Directory) | 4.7.5 |
| %END: (Stop parsing) | 4.7.6 |
| %EXT: (Extension to the %DES: keyword) | 4.7.7 |
| %FIL: (Original filename) | 4.7.8 |
| %LAN: (Supported languages) | 4.7.9 |
| %LEN: (Original filelength) | 4.7.10 |
| %OBJ: (Object keyword/category) | 4.7.11 |
| %PID: (Program identity) | 4.7.12 |
| %REC: (Recommended system setup) | 4.7.13 |
| %REG: (Registration fee) | 4.7.14 |
| %REP: (Replace files) | 4.7.15 |
| %REQ: (Required system setup) | 4.7.16 |
| %REV: (Revision) | 4.7.17 |
| %TYP: (Distribution type) | 4.7.18 |
| %VER: (Version string) | 4.7.19 |
| List of languages | 4.8 |
| Developer addendum | 5.0 |
| Creating a file description | 5.1 |
| Adding the description | 5.2 |
| Extracting the description | 5.3 |
| Parsing the description | 5.4 |
| Keywords in general | 5.5 |
| Keywords in specific | 5.6 |
| Keyword recognition | 5.7 |
| Keyword processing | 5.8 |
| Parsing 'Old Style' descriptions | 5.9 |
| Converting FILE_ID.DIZ | 5.10 |
| Date formatting | 5.11 |
| Miscellaneous | 6.0 |

=====

What am I trying to accomplish with
 2.1 Purpose
 this new standard? Find out!

-- -- -- --

Restrictions and rules on how you may
 2.2 Using the FDS
 use the File Description Standard.

-- -- -- --

A small glossary to study before you
 2.3 Glossary
 dive into the depths of this document.

1.7 Section 2.1 -- Purpose

2.1 PURPOSE

The File Description Standard (or FDS, for short) is a new standard for adding descriptions to ANY kind of file. The FDS also contains many new features not available in any other description-standards. For example, the ability to use multiple languages in the same description. The main goal with the FDS is to provide a simple, yet powerful, standard for the users and bulletin board systems alike.

The FDS also tries to maintain a flexible style, which will allow us to expand and enhance the standard if the need should arise. This document should provide users with enough knowledge to create descriptions.

1.8 Section 2.2 -- Using the FDS

2.2 USING THE FDS

Users and developers choosing to use/implement this standard MUST, with no exceptions, follow the standard. The standard is this document, plus the extended document for developers. Unless a feature is documented in any of these two documents, it is NOT a part of the standard.

1.9 Section 2.3 -- Glossary

2.3 GLOSSARY

There are some abbreviations and CAPITALIZED words throughout the docs, and you should familiarize yourself with the, sometimes, cryptic meaning of them. Here is a small list of abbreviations and other cryptic words.

==[GLOSSARY]=====

| | |
|---------|--|
| ARCHIVE | A file consisting of several compressed files. |
|---------|--|

| | |
|-----------|--|
| BINARY | All files except ARCHIVEs and TEXT files. |
| DEVELOPER | The developer implementing the standard in his software. |
| FDS | File Description Standard. |
| FILE | The file which the file description is referring to. |
| FOOTER | The string "@END~USG_FILEDESC". |
| HEADER | The string "@BEGIN~USG_FILEDESC". |
| TEXT | A file consisting only of readable text. |
| USER | The user who reads or writes a description. |
| USG | Users Standards Group. |

=====

1.10 Section 3.0 -- Using the standard

=====

USING THE STANDARD

3.0

=====

This section will describe how to create a file description, and how it is used in the three different filetypes. This section will not go into the format of the description, only how to create and use it.

3.1 Creating the file description

How do you create a file description? Well, it's not that hard. Read this section to find out how YOU can make use of some specialized software.

-- -- -- --

3.2 Using the file description

How do you go about "connecting" your description to the file you want to describe? There are three different filetypes to consider, and this section describes the difference between them.

-- -- -- --

3.3 Adding the file description to an archive

This section sheds some light on how to use your favorite archiver to add the file description to an archive.

-- -- -- --

3.4 Adding the file description to a text file

This section will teach you how to include a description directly into a text file.

-- -- -- --

3.5 Adding the file description to a binary file

This section mentions the difficulty with adding file descriptions to a binary file, and why you should not attempt to do it yourself.

1.11 Section 3.1 -- Creating the file description

3.1 CREATING THE FILE DESCRIPTION

Creating the file description is pretty straightforward actually. What you have to do is to get hold of a text-editor which is able to save the description as pure ASCII text. Then you just type in your description, according to the standard of course, save it as FILEDESC.TXT or maybe as a temporary file, and it is ready to use.

Maybe you are lucky enough to have an editor dedicated to the standard? In that case, you will only have to input the information needed and the editor takes care of saving it correctly. Some editors actually include online help, which means that you won't have to know the entire document by heart. There are, to my knowledge, only two editors available. Both for the Amiga range of computers. FileDescEditor by Mattias Palsson and DescMaker by Robert Bjorn.

If you are writing it manually, it is highly recommended to be familiar with the format of the file description, and to know the keywords really well. I can recommend using a template for your descriptions, where you remove the lines you don't want.

1.12 Section 3.2 -- Using the file description

3.2 USING THE FILE DESCRIPTION

Alright, now you have your file description written and ready. Next up is to add it to the FILE it describes. While there are three filetypes, there are two ways of adding the description. The first filetype is the ARCHIVE file, where the file description is added to the archive with an archiver. The second filetype is the TEXT file, where you simply put it as text inside the text file. The last filetype is the trickiest to use as it isn't easily done by hand. I will explain the details in the next sections, 3.3 through 3.5.

3.3 Adding the file description to an archive

This section sheds some light on how to use your favorite archiver to add the file description to an archive. ↔

— — — — —

3.4 Adding the file description to a text file

This section will teach you how to include a description directly into a text file. ↔

- - - - -

3.5 Adding the file description to a binary file

This section mentions the difficulty with adding file descriptions to a binary file, and why you should not attempt to do it yourself.

1.13 Section 3.3 -- Adding the file description to an archive

3.3 ADDING THE FILE DESCRIPTION TO AN ARCHIVE

This is the most straightforward way to use the description, and you're encouraged to ALWAYS archive your files when distributing them. Not for your sake, but for everyone else's. Anyway, to add the file description to an ARCHIVE, you simply use the archiver to add it in the archive with the filename FILEDESC.TXT (uppercased if possible).

Here are a few examples on how to add it;

```
.ARJ  [PC]   Arj 2.30      ARJ a {archive file} FILEDESC.TXT
.LHA  [AM]   LhA 1.50     LHA a {archive file} FILEDESC.TXT
.LZH  [PC]   LhA 2.13     LHA a {archive file} FILEDESC.TXT
.LZX  [AM]   Lzx 1.01     LZX a {archive file} FILEDESC.TXT
.RAR  [PC]   Rar 1.52     RAR a -std {archive file} FILEDESC.TXT
.ZIP  [PC]   PkZip 4.1v   PKZIP -a {archive file} FILEDESC.TXT

[AM]   = Amiga range of computers.
[PC]   = IBM/PC compatibles.
```

1.14 Section 3.4 -- Adding the file description to a text file

3.4 ADDING THE FILE DESCRIPTION TO A TEXT FILE

To add the file description to a TEXT file, you will need to include it in the actual FILE. Due to the fact that it is a pure text file, that's not very complicated. Here is a "step-by-step" instruction;

Let's say that you've written a text, and saved it as NICETEXT.TXT. As you are about to upload it to your local BBS, you want to create and add a description to go with it. Let's assume you have already created your description, and saved it as FILEDESC.TXT. These files should be merged together, either by loading both files into a texteditor and copying the file description to the NICETEXT.TXT file, or by using some sort of join command to merge them. Anyway, you'll end up with something like this;

```
--- 8< -----
This is the fantastic text you have written. Or at least you
could imagine it.
-> %DES: Default
-> This is the actual text for the english/default description.
-> %END:
```

```
--- 8< -----
```

The lines marked with arrows are the file description. This is not the final result though. To make it possible to know which lines are a part of the file description we use two marker-strings, which I will refer to as HEADER and FOOTER. The header string is "@BEGIN~USG_FILEDESC", while the footer string is "@END USG_FILEDESC". The header and footer MUST be typed in uppercased letters. What you want to do now, is to use the two marker strings like this;

```
--- 8< -----
This is the fantastic text you have written. Or at least you
could imagine it.
-> @BEGIN~USG_FILEDESC
%DES: Default
This is the actual text for the english/default description.
%END:
-> @END~USG_FILEDESC
--- 8< -----
```

As you can see by the arrow-marked lines, the markers are inserted just before and after the file description. Isn't it simple? You could also write the file description when you write the text, that doesn't matter, just remember to ALWAYS include the marker-strings.

And if this sounds to complicated for you, keep an eye out for software which allows you to do this operation almost automatically.

1.15 Section 3.5 -- Adding the file description to a binary file

3.5 ADDING THE FILE DESCRIPTION TO A BINARY FILE

Adding a file description to a BINARY file is done the same way as with the TEXT file. With the notable exception that you can't normally do it by yourself. Most BINARY filetypes are very strictly constructed, which means that simply joining the file description to the end of the file is NOT recommended. As a user, you should NOT attempt to add a description to a binary file unless you have VERY good knowledge of the fileformat.

Conclusion; you need specialized software to add file descriptions to a binary file. Be warned, tampering with binary files may destroy them.

Still, you are probably better off archiving the file instead. That is the only solution if you don't have means to add the file description in any other way.

1.16 Section 4.0 -- Format of the file description

```
=====
FORMAT OF THE FILE DESCRIPTION                                4.0
=====
```

This section of the document describes the format and components of the file description. This is essential knowledge for everyone who wants to use this standard. This is what the FDS is all about.

4.1 Components of the file description

This section explains it all. Well, almost. It describes the way in which a file description is constructed...

-- -- -- --

4.2 The default description

What is this "default description" thingy? It is the text that is supposed to be used when the requested language couldn't be found. Default language is English.

-- -- -- --

4.3 The short description

This section explains what the SHORT description is, and why we want it. Or, why some people want it...

-- -- -- --

4.4 The extended description

OK, but what the heck is this "extended" description you are talking about? The extended description is the lines following the short description, and it's supposed to tell you a whole lot more about the software in question.

-- -- -- --

4.5 Sections in an extended description

Sections?! Now you've got me confused! Well, it's pretty simple really. The FDS supports wordwrapping of the file descriptions, which is more or less unique. Find out how!

-- -- -- --

4.6 Keywords

This section explains how the keywords work, and how they are constructed. Go ahead, it's a quick read...

-- -- -- --

4.7 List of current keywords

This section takes you on a guided tour of the different keywords available. Welcome... To Jurassic Park!

-- -- -- --

4.8 List of languages

This section contains a list of language strings that you are allowed to use in the FDS. Go international! ←

1.17 Section 4.1 -- Components of the file description

4.1 COMPONENTS OF THE FILE DESCRIPTION

A file description is made up almost entirely of keywords. The keyword concept makes the standard flexible, and expandable if needed. Keywords are used to enter all kinds of information about the FILE.

All keywords, except %DES:, uses one line each in the file description. The format for all one-line keywords is;

```
%KEY: {data for this keyword}
```

%KEY: does NOT exist, it is just used as an example. All keyword has a percent-sign as the first character and a colon as the fifth. The chars between the % and the : is the keyword identifier. Everything after the colon is data for the keyword. For example, the keyword for author;

```
%AUT: Peter Bornhall
```

As I mentioned above, there is one exception to the one-line rule. The %DES: keyword is used for the actual description (do NOT confuse it with the FILE description) text.

Well, how DO you write the actual description text?

That is where the %DES: keyword comes into action. First of all, %DES: tells us which language is used by the description text that follows the keyword. For example;

```
--- 8< -----
%DES: Default
-> FilePather 1.3 -- BBS adder/remover/history.
   FP creates a history within an archive or a
   textfile. Each time FilePather processes a file
   it will update the list with one line of text
   which is configurable by the user.
%AUT: Johan Torin
%VER: FilePather 1.3
--- 8< -----
```

The first line tells us that this is the DEFAULT description. The next line, marked with an arrow, is the SHORT description, which always comes directly after the %DES: keyword line. The lines following the short is the EXTENDED description. The extended description uses all lines until the next keyword, which is %AUT: in this case. See sections 4.3 through 4.4 for more info on the short and extended descriptions. You will also find a full explanation of the %DES: keyword in section 4.7.4.

Descriptions should be written using ordinary ASCII characters, but the standard allows the use of different charsets through the %CHR: keyword. The default charset is ISO 8859-1, also known as LATIN-1.

All lines in the file description may be of any length, but try to keep the linelength less than 81 characters for practical reasons.

See also:

- Section 4.2 -- Default description
- Section 4.3 -- Short description
- Section 4.4 -- Extended description

1.18 Section 4.2 -- Default description

4.2 THE DEFAULT DESCRIPTION

As you know, the FDS allows multiple languages in the file description. One of these is English, which is also known as the DEFAULT description. You should ALWAYS have a Default/English description in all descriptions you create. Here is an example of a multi-language file description;

```

--- 8< -----
%DES: Svenska
{short description in Swedish}
-> %DES: Default
-> {short description in English}
%DES: Espanol
{short description in Spanish}
--- 8< -----

```

For further examples, see the enclosed example file.

1.19 Section 4.3 -- Short description

4.3 SHORT DESCRIPTION

The SHORT description is a one-line description of the FILE. It should be a short but informative line. Remember that some BBS's only use some 40-50 chars for their one-line descriptions, so please keep it brief.

The short description always follow the %DES: keyword. You must have a short description in every %DES: description.

The SHORT description should be something like these examples;

Phonebill 3.2 - Keeps track of your phonebill.

GIF 800x600, HQ scan of space shuttle Discovery.

S3M: Nightfall, atmospheric tune by unknown author.

See also:

Section 4.4 -- Extended description

1.20 Section 4.4 -- Extended description

4.4 EXTENDED DESCRIPTION

Just as the title says, the EXTENDED description is an extension to the SHORT description. It's in no way intended to replace the short, but to complement it. You should actually consider the short description to be the first line of the extended description. So please try not to repeat yourself in the extended description. An extended description is ALWAYS located directly after a short description.

```

--- 8< -----
%DES: Default
FilePather 1.3 -- BBS adder/remover/history.
-> FP creates a history within an archive or a
-> textfile. Each time FilePather processes a file
-> it will update the list with one line of text
-> which is configurable by the user.
%AUT: Johan Torin
%VER: FilePather 1.3
--- 8< -----

```

See also:

Section 4.3 -- Short description

Section 4.5 -- Sections in an extended description

1.21 Section 4.5 -- Sections in an extended description

4.5 SECTIONS IN AN EXTENDED DESCRIPTION

Another feature in the FDS is the possibility of word-wrapping the text to any linelength. This allows a BBS to wrap the description to desired linelength. There is a small sub-feature that is called SECTIONS.

OK, let us assume you have an EXTENDED description that tells you which features a certain software have. Something like this;

```

--- 8< -----
* This program support the File Description Standard release 2.
* Works with ALL systems!
* Source included.
--- 8< -----

```


Three lines, three features. What would happen if these lines were put through a word-wrapping routine? Probably total chaos. That's where the magic comes to work. Note how none of the lines begin in the very first column, they are all indented one char. That is the secret used to find the SECTIONS. When a line begins with a whitespace character like space or tab, it is considered to be the first line of a new SECTION.

Each section is then wrapped separately, which would yield this;

| | |
|---|---|
| <pre> Wrapped for 40 chars --- 8< ----- * This program supports the File Description Standard release 2. * Works with ALL systems! * Source included. --- 8< ----- </pre> | <pre> Wrapped for 20 chars --- 8< ----- * This program supports the File Description Standard release 2. * Works with ALL systems! * Source included. --- 8< ----- </pre> |
|---|---|

Of course, the result may look horrible in some cases, but I can almost guarantee you that very few boards use 20 chars as maximum linelength.

Still, it is possible to construct a word-wrapping routine that is more advanced. A routine that could indent wrapped lines properly, like;

```

Wrapped for 40 chars
--- 8< -----
* This program supports the File
  Description Standard release 2.
* Works with ALL systems!
* Source included.
--- 8< -----

```

Note how the second line is indented correctly. This isn't a very hard thing to accomplish if you have enough programming skill.

See also:

Section 4.4 -- Extended description

1.22 Section 4.6 -- Keywords

4.6 KEYWORDS

KEYWORDS are used for entering specific information regarding the FILE, the author, charset and similar things. The keyword concept also ensure that the standard is expandable if needed.

Keywords begin with % (ASCII 37), followed by a three-letter identifier and a colon. With the exception of %DES:, all keywords consists of only one line each. Each keyword line contains a keyword and possibly a line of text directly after. A file description may have unlimited keywords.

The three-letter identifier is the unique part of the keyword, which is

used to identify the keyword. You could actually use a %ABC: keyword if you wanted to. It would be recognized as a keyword, but as it's unknown it will simply be ignored. The three-letter identifier consists only of combinations of A-Z and 0-9.

1.23 Section 4.7 -- List of current keywords

4.7 LIST OF CURRENT KEYWORDS

This is a list of all currently supported keywords. For more info, see the sections 4.7.1 through 4.7.19.

```
==[ KEYWORDS ]=====
```

```
%AUT:
  <name>          Author information

%CHR:
  <charset>       Charset used for description

%DAT:
  <date>          Releasedate

%DES:
  <language>      Language-specific description

%DIR:
  <directory>     Directory for the file

%END:
                  Stop parsing

%EXT:
  <commands>      Extension for the %DES: keyword

%FIL:
  <filename>      Original filename

%LAN:
  <language>      Supported languages

%LEN:
  <bytes>         Original filesize, exact bytes

%OBJ:
  <category>      Categorize the file

%PID:
  <name>          Created by which program?

%REC:
  <setup>         Recommended system setup

%REG:
  <fee>           Registration fee
```

```

%REP:
  <filename>      Replace file

%REQ:
  <setup>         Required system setup

%REV:
  <revision>      Revision of the object

%TYP:
  <distribution> Distribution type

%VER:
  <version>       Versionstring
=====

```

All keywords are of course optional.

1.24 Section 4.7.1 -- %AUT: (Author information)

4.7.1 %AUT: (Author information)

This keyword should reflect the name of the author or development group responsible for the software. You may add some additional info like for example email address or similar. Keep it short though.

```

Examples:  %AUT: John Doe, jdoe@nowhere.com
           %AUT: Peter Bornhall, 2:201/417.41@fidonet

```

1.25 Section 4.7.2 -- %CHR: (Charset used in description)

4.7.2 %CHR: (Charset used in description)

This keyword should reflect which charset was used for the description. This makes it easier to convert the description to other charsets, like IBMPC, or codepage 437. If this keyword is omitted, LATIN-1 is used as default charset.

| | | | |
|-----------------------------|---------|------------|-------|
| Available charsets include: | ASCII | Standard | 7-bit |
| | IBMPC | CP437 | 8-bit |
| | LATIN-1 | ISO 8859-1 | 8-bit |
| | MAC | MacIntosh | 8-bit |
| | SWEDISH | Swedish | 7-bit |

```

Examples:  %CHR: IBMPC
           %CHR: MAC

```

1.26 Section 4.7.3 -- %DAT: (Releasedate)

4.7.3 %DAT: (Releasedate)

This keyword should reflect the releasedate of the FILE. In some cases it's useful to know when a software was released, for example to know if it is outdated or not. Dates should always be written '<d>.<m>.<y>'.

Examples: %DAT: 27.11.95
 %DAT: 8.1.96

1.27 Section 4.7.4 -- %DES: (Language-specific descriptions)

4.7.4 %DES: (Language-specific descriptions)

The %DES: keyword could more or less be considered the core of the FDS. This keyword allows multiple languages in the same description, and uses always at least two lines. All lines between the %DES: keyword line and the next keyword (invalid keywords too) belongs to the same description. Please note that "description" in this case does NOT refer to the entire file description, just the description part of the %DES: keyword.

Every %DES: description consists of the keyword and its language, SHORT description and optionally an EXTENDED description. See also chapter 4.

For example, take a look at this (also from chapter 4);

```

--- 8< -----
%DES: Default
-> FilePather 1.3 -- BBS adder/remover/history.
--- 8< -----

```

This is a simple but valid description. The first line, with the %DES: keyword, tells the parser that a description will follow, and that it is the DEFAULT (English) language. The line following the %DES: keyword is always (unless it is another keyword) the SHORT description.

```

--- 8< -----
%DES: Default
FilePather 1.3 -- BBS adder/remover/history.
-> FP creates a history within an archive or a
-> textfile. Each time FilePather processes a file
-> it will update the list with one line of text
-> which is configurable by the user.
--- 8< -----

```

If the line following the SHORT description isn't a keyword, then it is the EXTENDED description. All lines until the next keyword then belongs to the DEFAULT (in this case) description.

See also:

Section 4.2 -- Default description

Section 4.3 -- Short description

Section 4.4 -- Extended description

Section 4.8 -- List of languages

%EXT: (Extension to the %DES: keyword)

1.28 Section 4.7.5 -- %DIR: (Directory)

4.7.5 %DIR: (Directory)

This keyword is used to sort the file into a specific directory, for example for use on AmiNet. You should only enter the path, without any drive information. Directories are separated with a forward slash, and a trailing slash is not obligatory.

Example: %DIR: docs/misc/

1.29 Section 4.7.6 -- %END: (Stop parsing)

4.7.6 %END: (Stop parsing)

This keyword is used only to mark the end of a file description. It is not required, but you should use it just to be on the safe side. Please note that this keyword ends the ENTIRE file description. It is NOT used as an end-marker for the %DES: keyword!

Example: %END:

1.30 Section 4.7.7 -- %EXT: (Extension to the %DES: keyword)

4.7.7 %EXT: (Extension to the %DES: keyword)

This keyword is only used in conjunction with the %DES: keyword, and it only applies if it is found directly after the description.

Example: %DES: Default
This is the SHORT description.
This is the EXTENDED description.
The extension keyword is only valid if it follows directly after the description, like this.
-> %EXT: NOWRAP

This keyword has, currently, only one command; NOWRAP. If you use this command, the description will NOT be wrapped. Although descriptions are supposed to be written with wrapping in mind, this keyword allows you to prevent the description from wrapping.

See also:

%DES: (Language-specific descriptions)

1.31 Section 4.7.8 -- %FIL: (Original filename)

4.7.8 %FIL: (Original filename)

This keyword should tell the original name of the FILE when it was sent into distribution. Could be used to check for duplicates, which used in conjunction with the %REP: keyword could help a sysop keep his HD clean.

If the filename exceeds the 8+3 limit imposed by MS-DOS, you might want to use two filenames, where one of them conforms to the 8+3 standard and the other does not. Separate filenames with space and use quotes if the filename contains any spaces.

```
Examples:  %FIL: GreatWhite_demo.lha
           %FIL: MANHATTN.GIF "Manhattan Skyline.gif"
```

See also:

```
%REP: (Replace files)
%LEN: (Original filelength)
```

1.32 Section 4.7.9 -- %LAN: (Supported languages)

4.7.9 %LAN: (Supported languages)

Do NOT confuse this with the %DES: keyword. This keyword could be used to check if a certain software is of any use to a user. For example, in Germany there are many programmers whose software is German-only. For a person who can't understand German, the software is more or less useless to him. Could save you some hard-earned download credits. For example, IF your software is German-only, add '%LAN: Deutsch'.

Languages should be entered in the NATIVE tongue, "Deutsch" for German, "Svenska" for Swedish and so on. ALL languages that are supported would be entered on the same row, separated by a / (forward slash).

If the software supports user-changeable languages through the OS, like locale on AmigaDOS 2.1 or higher, you should enter the first language as 'Locale'. The first language following the locale string should reflect the built-in language. Any languages following after, are supplied with the software.

If this keyword is omitted, the FILE is considered to be international, or English. English is more or less understood by most computer-users.

```
Examples:  %LAN: English/Deutsch/Dansk/Svenska
           %LAN: Locale/English/Deutsch/Espanol
```

See also:

```
Section 4.8 -- List of languages
```

1.33 Section 4.7.10 -- %LEN: (Original filelength)

4.7.10 %LEN: (Original filelength)

This keyword should reflect the original length of the distributed FILE in bytes. Can be used to check if the user has stripped away BBS ads or not, for example. Filelength is the exact number of bytes the FILE plus the file description occupies. For an ARCHIVE, this might be difficult, but not impossible. First sequence of numeric chars should be used.

Examples: %LEN: 81962 bytes
 %LEN: 12281

See also:

 %FIL: (Original filename)

1.34 Section 4.7.11 -- %OBJ: (Object keyword/category)

4.7.11 %OBJ: (Object keyword/category)

This keyword is used to categorize the FILE, which makes it possible to sort the file into different areas, and allows the users to search for a certain type of software. You may use as many categories as you like, a file might fit into more than one category, but the FIRST category is to be considered the most important category. Always use the categories in the order they are to be sorted by.

Examples: %OBJ: Communications Utility
 %OBJ: Programming Assembler

You are allowed to use any name for a category, but you should consider using one or more of the categories listed below. If none of them suits your particular software, use some simple category name, instead of some cryptic category nobody understands. Or use the Miscellaneous category, used for hard-to-categorize files. If you feel that it might be useful, use multiple categories. Multiple categories can make it easier to find files with certain properties.

Examples: %OBJ: Programming
 %OBJ: Programming C

The first example refers to something that has to do with "Programming" in general. The second example is more specific, referring also to "C". Use this feature wisely.

Here is a list of categories you can use;

```
==[ CATEGORIES ]=====
3D                Action                Adventure
Amiga             Analysis               Animation
Archiver          Assembler              BASIC
BBS               Business                C
C++               CAD/CAM                  CDROM
```

| | | |
|--------------|-------------|----------------|
| CLI | Commodity | Communications |
| Composition | Compression | Conversion |
| Database | Demo | Disk |
| Display | Drawing | DTP |
| DTV | E | Editor |
| Education | Game | Graphics |
| Image | Library | Logo |
| Macintosh | Mail | Miscellaneous |
| Modula-2 | Mouse | MS-DOS |
| Multimedia | Music | Network |
| Oberon | OS/2 | Painting |
| Pascal | Picture | Player |
| Presentation | Printing | Processing |
| Programming | Publishing | Racing |
| Raytracing | Shell | Simulation |
| Sound | Source | Sport |
| Spreadsheet | Strategy | Terminal |
| Text | Tool | UNIX |
| Utility | Video | Viewer |
| Windows | WindowsNT | Wordprocessing |
| Workbench | | |

=====

1.35 Section 4.7.12 -- %PID: (Program identity)

4.7.12 %PID: (Program identity)

This keyword should only be used by software that creates or edits this kind of descriptions. As a user, you should not use this keyword.

Examples: %PID: FileDescEditor 2.0
%PID: DescMaker 1.3

Format of this string is free, but;

<name> <version>.<revision>

...is the preferred standard. See %VER: for more information.

1.36 Section 4.7.13 -- %REC: (Recommended system setup)

4.7.13 %REC: (Recommended system setup)

This keyword should be used to tell the user what the recommended setup for this FILE is. Many times users download things they cannot use with their particular system or have an inadequate system to use the software to its full extent. This keyword should inform the user what the author recommends as a minimum to use it effectively.

For example, a raytracing package will probably work without a FPU, but it is RECOMMENDED to have such a processor installed, otherwise you will have to wait days instead of hours. Another example, a C compiler might

be usable from floppy disk, but a harddrive is highly recommended due to the intense read/write operations performed by the compiler.

Note though that this is a RECOMMENDATION, NOT a requirement.

Examples: %REC: 68020+, HD, FPU
 %REC: 486DX2/66, SVGA/1MB

See also:

 %REQ: (Required system setup)

1.37 Section 4.7.14 -- %REG: (Registration fee)

4.7.14 %REG: (Registration fee)

This keyword is intended for shareware software, where it's supposed to let the user know what a registration costs. If possible, enter the fee in US dollars (US\$). That makes it easier for a user to convert the fee to his or her own currency. You may use multiple currencies, but if you enter US\$, always put it first on the line. Separate with commas.

Examples: %REG: US\$40
 %REG: US\$20, 35DM, 100SEK
 %REG: 150SEK

See also:

 %TYP: (Distribution type)

1.38 Section 4.7.15 -- %REP: (Replace files)

4.7.15 %REP: (Replace files)

This keyword is intended to provide a way to automatically remove older versions of a software, or simply notify the sysop that the old versions can be removed from the system. For example you have just finished your latest version of your program MyGoodProgram, or MGP for short. The new version is 1.2, and you are just about to start distributing it to local BBS's. This latest version will probably succeed the previous versions, let's say you have released v1.0 and v1.1 earlier, which means that they are as obsolete as they're gonna get. To automatically remove the older versions, add the following line in your file description;

 %REP: MGP10.ZIP MGP11.ZIP

This way the author provides a way for a sysop to automatically replace the old versions with the new. And by doing that, keeping his harddrive clean from obsolete, old, buggy, and otherwise unwanted files.

Note also that this keyword allows you to enter multiple filenames with one or more spaces between them. Filenames with spaces should be quoted

to prevent any mistakes. See examples.

```
Examples:  %REP: WMAN122.ZIP WMAN123.ZIP
           %REP: CUCUMBER.LZH
           %REP: "NICE BIRDY.IFF" "NICE BIRD.BMP"
```

See also:

```
%LEN: (Original filename)

%LEN: (Original filelength)
```

1.39 Section 4.7.16 -- %REQ: (Required system setup)

4.7.16 %REQ: (Required system setup)

This keyword is used to let the user know what the MINIMUM REQUIREMENTS for this particular FILE is. If the software requires a particular kind of processor, amount of memory or something similar, enter it here.

```
Examples:  %REQ: Amiga, 68020, 1 MB FAST RAM
           %REQ: MSDOS 3.30, 80286, CGA
           %REQ: Win95, 32MB RAM
```

See also:

```
%REC: (Recommended system setup)
```

1.40 Section 4.7.17 -- %REV: (Revision)

4.7.17 %REV: (Revision)

This keyword is a complement to the %VER: keyword. This keyword should tell you the REVISION version, if applicable, of the software.

```
Examples:  %VER: Spot 1.3 (16.8.94)
           %REV: 37.4515
```

This example taken from real life, means that this is Spot 1.3. That's what is considered to be the RELEASE version. In many cases this number is the same as the version of the actual software, but in this case it's the version of the entire package. The actual REVISION version, of Spot in this case, is 37.4515.

See also:

```
%VER: (Version string)
```

1.41 Section 4.7.18 -- %TYP: (Distribution type)

4.7.18 %TYP: (Distribution type)

This keyword tells the user if the author wants anything in return when you use the software. Some authors want money (Shareware), some settle for a postcard (Cardware) or a gift of some sort (Giftware). Please try to fulfill the demands of the author. Programs worth using are programs worth paying for.

There are many types of "payment", involving money in some cases;

```

====[ TYPES ]=====
Public Domain - Free to use/sell/change without restrictions.
Freeware     - Free to use without any kind of payment.
Shareware    - You must pay a small fee to use the software.
Crippleware  - The same as shareware, but less functionality
              until you have paid for the software.
Cardware     - The author wants a postcard as payment.
Callware     - The author wants you to call his BBS as payment.
Giftware     - The author wants a gift of some kind as payment.
Charityware  - The author wants you to donate money to charity.
=====

```

```

Examples:    %TYP: Giftware
             %TYP: Charityware

```

See also:

```
%REG: (Registration fee)
```

1.42 Section 4.7.19 -- %VER: (Version string)

4.7.19 %VER: (Version string)

This keyword should reflect the name, version and date of the software, if applicable. Most programs should make use of this keyword. You must always enter the information according to this format;

```
<name> <version>.<revision> (<d>.<m>.<y>)
```

Note also that <version>.<revision> must NOT include a leading "v".

```

Examples:    %VER: Program 1.4 (6.11.94)
             %VER: FaceIt 3.37 (12.1.1995)

```

See also:

```
%REV: (Revision)
```

1.43 Section 4.8 -- List of languages

4.8 LIST OF LANGUAGES

This section contains a list of language-strings that you should use in the %DES: and %LAN: keywords. Do NOT use anything else than the strings specified here, unless the language isn't IN the list.

```

====[ LANGUAGES ]=====
Default          (English)      Use "Default", not "English"
Locale           Only used in the %LAN: keyword
Dansk            (Danish)
Deutsch          (German)
Espanol         (Spanish)
Francais       (French)
Italiano        (Italian)
Nederlands      (Dutch)
Norsk           (Norwegian)
Portugues       (Portuguese)
Suomi           (Finnish)
Svenska         (Swedish)
=====

```

As you might have noticed, the strings used are the NATIVE form for the language. So instead of "Swedish", use "Svenska". The only changes are that accented characters has been replaced by their non-accented cousin.

If YOUR native language isn't in the list, please tell me about it, and I will include it in the next release. The listed languages should more or less be the most common, though.

There is also a special language, named Elite, which only purpose is to provide a way for people to use ASCII logos and similar descriptions. I also recommend that converted file_id.diz descriptions be placed in this "language". Note that an implied %EXT: NOWRAP is used for this language because of its nature.

1.44 Section 5.0 -- Developer addendum

```

=====
DEVELOPER ADDENDUM
=====

```

5.0

This section's one and only intention is to provide some additional, or complementing, rules and tips applying to the File Description Standard.

As a developer you are required to FULLY understand the specifications! If there is something unclear, drop me a line and we'll discuss it!

As a developer you are allowed to use excerpts from section 4.7.1 up to 4.7.19 as on-line help in your software.

5.1 Creating a file description

Things to consider if your software creates/alters a file description. ↔

-- -- -- --

5.2 Adding the description

How do you go about adding the description to the different types of files? Find out about it here... ↔

-- -- -- --

5.3 Extracting the description

This section explains how you should go about extracting the file description, and how you find it in the first place. ↔

-- -- -- --

5.4 Parsing the description

How do I parse the description then? Well, this section should explain the main things you have to consider. ↔

-- -- -- --

5.5 Keywords in general

Some general rules and hints on keywords.

-- -- -- --

5.6 Keywords in specific

Rules and hints on specific keywords

-- -- -- --

5.7 Keyword recognition

How can you tell it really IS a keyword?

-- -- -- --

5.8 Keyword processing

What is involved when you process the keyword?

-- -- -- --

5.9 Parsing 'Old Style' descriptions

If you want to support release 2 of FDS, read this.

-- -- -- --

5.10 Converting FILE_ID.DIZ

How to convert a FILE_ID.DIZ description to FDS format ↔

-- -- -- --

5.11 Date formatting

This section shows you how you should format the dates ←

1.45 Section 5.1 -- Creating a file description

5.1 CREATING A FILE DESCRIPTION

If your software allows users to create descriptions please inform them that they should refrain from using graphic or international characters. Not that it is impossible to handle, but translating the graphic borders from IBMPC to LATIN-1 is not standardized in any way...

When saving the description, try to make sure that no single line in it exceeds the 80 chars "limit". There is no limit to the line length, but for cosmetic and practical reasons 80 chars is better. I don't want you to discard anything, but simply do not allow the user to enter more than 80 chars. Exception; the %DES: keyword, which you should wrap to fit in 80 chars. See

Section 4.5
for more information on wrapping.

If your software have created a description you should warn the user if there is no %DES: Default (DEFAULT description) present. The DEFAULT is optional, but it should really be present.

When you create a file description, please make sure that you save the file in uppercase. Case sensitive filesystems do exist you know.

1.46 Section 5.2 -- Adding the description

5.2 ADDING THE DESCRIPTION

When you add the description to a file, binary or ASCII, you should try to add it in the first or last part, preferably in the first or last 4K. This will make it easier to find, and also allow faster scanning on very large files.

When you add a file description to an archive please make sure that the file is uppercased before adding it. Case sensitive filesystems exists, so please think carefully about this.

1.47 Section 5.3 -- Extracting the description

5.3 EXTRACTING THE DESCRIPTION

Parsing software, which extracts and parses descriptions, should handle all filetypes; i.e. ARCHIVE files, TEXT files and BINARY files. The two latter types are handled in the same manner, while the ARCHIVE type need access to the archiver (in most cases at least).

When you scan a file, binary or ASCII, for a description, you should do a case insensitive search which allows your software to find HEADERS and FOOTERS which might be written in lowercase. When writing a description to a textfile, ALWAYS write the HEADER and FOOTER in uppercase!

Also, when adding the HEADER, try to precede it with a newline. That's for making it easier to find, as it will begin fresh on a new line. The newline sequence is CR+LF, or LF alone.

For more "hands-on" experience, check out the FindDesc source.

When scanning a file, binary or ASCII, you might want to scan the first and last 4K (4096 bytes) before scanning the bit in between. This could be used especially for very large files, which otherwise would take some time to scan. This is optional, but I recommend developers to add their descriptions to the first or last part of the file, so...

Also note that you MUST support CR+LF or LF alone as newline! FDS take other platforms into consideration, so this one is important.

1.48 Section 5.4 -- Parsing the description

5.4 PARSING THE DESCRIPTION

When you encounter a %DES: keyword which has no language string, ignore it completely. Same thing if the following row is another keyword. You should act as if they weren't even there in these two cases.

A description consists of a short description and an optional extended. In no situation could there be an extended description without the short description preceding it.

Remember that the default charset used is LATIN-1 (ISO 8859-1), and you should be aware of the fact that you might have to translate the text in the descriptions. For example, if the description contains %CHR: IBMPC, which is codepage 437, and you want the output in LATIN-1, you will have to translate any international characters. I leave this to you...

If your software is supposed to extract a certain language for a BBS, I recommend that you consider having a list of preferred languages that it could pick from. For example in the nordic countries most languages are similar, so a Danish sysop might want to have a list of;

```
Dansk  
Norsk  
Svenska  
Default  
Deutsch
```

...where he uses the Danish description if it exists. If not, look for a Norwegian description, or a Swedish. If neither of these exists, he'd like to use the default (English) description, or the German, if there's no default description either. If none of the languages in the list are found, use the default description. If there is no description, you are on your own...:)

If you encounter a %DES: Elite, remember that you MUST NOT wordwrap the contents of it. Also, using it for one-line (short) descriptions can be problematic, since it probably isn't formatted with FDS in mind.

1.49 Section 5.5 -- Keywords in general

5.5 KEYWORDS IN GENERAL

Here are some general rules about how to handle the keywords.

- Always put %DES: keywords at the top of the description.
- Always put the DEFAULT description at the very top
- Always write keywords in uppercase.
- Allow parsing of lowercased keywords.
- Parse each keyword, except %DES:, only once (first occurrence).
- Ignore unknown keywords when parsing, but do NOT remove them.
- End all description parsing when you encounter the %END: keyword.
- Keywords are global, and applies to all languages.

1.50 Section 5.6 -- Keywords in specific

5.6 KEYWORDS IN SPECIFIC

%DES:

Note, the contents is LANGUAGE, not the country. ↔

%LEN:

Use only the first "word" of the contents for length. ↔

%REP:

This keyword can be considered a dangerous keyword. It is intended to allow automatic removal of older versions of a software, and thus it involves deleting files. You are aware of the potential danger this causes for sysops who are running a BBS? This means that if your software acts upon this keyword you should provide the sysop with ↔

some way to protect certain files. Two tips;

- ~Don't allow deleting of delete-protected files.
- Use a list of files, which should never be deleted.

Also, this keyword could simply *NOTIFY* the sysop that there is obsolete software in the system. That way he'd be able to authorize any automatic deletions.

1.51 Section 5.7 -- Keyword recognition

5.7 KEYWORD RECOGNITION

When checking for valid keywords, you should check for both the percent sign and the colon. A keyword line ALWAYS begins with percent, ASCII 37 decimal, in the first position. If it doesn't; discard it right away.

To make sure it really is a keyword though, you should check position 5 too, where the colon, ASCII 58 decimal, should be. If there is no colon there, discard the line, it doesn't contain a keyword.

1.52 Section 5.8 -- Keyword processing

5.8 KEYWORD PROCESSING

When you parse a keyword and its contents, remember that there MIGHT be a space between the colon and the contents. This should be removed from the contents before parsing it. In effect; strip away any leading space and tab characters.

If/when your software writes a keyword line with contents, add a single space, ASCII 32 decimal, between the colon and the contents. This makes it easier to read with the human eye.

1.53 Section 5.9 -- Parsing 'Old Style' descriptions

5.9 PARSING 'OLD STYLE' DESCRIPTIONS

If you want to add support for the previous releases of the FDS, please take a look at the FDS-206 specifications, available from the author. I don't think there would be too much hassle to incorporate support for it but that's up to each and everyone.

1.54 Section 5.10 -- Converting FILE_ID.DIZ

5.10 CONVERTING FILE_ID.DIZ

If you convert a file_id.diz description, try to examine the percentage of actual alphanumeric characters. If this percentage is below 60%, you should convert the description into a %DES: Elite description, and add a %EXT: NOWRAP after it. Otherwise, save it as %DES: Default and skip the nowrap option.

1.55 Section 5.11 -- Date formatting

5.11 DATE FORMATTING

You should consider the following rules on date formats;

- Dates always in <d>.<m>.<y> format.
- Year may be in two- or four-digit format, write two-digit.
- Allow leading zeros when reading, but do NOT write leading zeros.

1.56 Section 6.0 -- Miscellaneous

6.0 MISCELLANEOUS

6.0

Where to send your mail if/when you

6.1 The author
want to contact me...

— — — — —

Who was involved in the development

6.2 Acknowledgments
of this standard? Greetings...

— — — — —

What happened during the development

6.3 History
of this standard? Find out!

— — — — —

What do I have to say in defence of

6.4 Last words
my crimes? Not much, probably...

1.57 Section 6.1 -- The author

6.1 THE AUTHOR

You may contact the author at the following address(es);

Snail: Peter Bornhall
 Bäck 5, Främmestad
 S-46598 NOSSEBRO
 SWEDEN

FidoNet: 2:201/417.41
 InfoNet: 42:105/106.41
 AmigaNet: 39:164/104.41

email: bornhall@karkis.canit.se

1.58 Section 6.2 -- Acknowledgments

6.2 ACKNOWLEDGMENTS

First and foremost, I would like to thank the members of USG, whom have helped me in developing this standard.

Secondly, I'd like to send some special thanks;

| | |
|----------------------|---|
| - Henrik Lindhe | For being the first supporter of FDS |
| - Mattias Palsson | For further spreading the word on this |
| - John Hertell | For many ideas on the language descriptions |
| - Mattias Ahnberg | For suggestions and ideas. |
| - Robert Bjorn | For slowing the "development" down a bit |
| - Christian Wikander | For supplying me with FDS software |
| - Piergiorgio Ghezze | For being first international contact |
| - Mikael Berglund | For spreading the word |
| - Micke Persson | For your excellent "bughunting" |
| - Marcus Ohlstrom | For dropping everything to do FindDesc |
| - Mats Oscarsson | For the spell-checking |
| - Jörgen Olmats | For suggestions |

General greetings and acknowledgments;

| | |
|------------------------|--|
| - Nico Francois | For Spot, ReqTools and PowerSnap |
| - Stefan Boberg | For LhA |
| - Timm Martin | For SID |
| - William S Hawes | For ARexx and WShell |
| - Magnus Holmgren | For Visage |
| - Thomas Krehbiel | For Viewtek |
| - Holger Kruse | For ReOrg |
| - Dave Haynie | For DiskSalv3 |
| - Olaf "Olsen" Barthel | For Term |
| - Raymond Penner | For PhoneBill |
| - Petri Nordlund | For Executive |
| - Peter Stuer | For BootX |
| - Digital Illusions | For Pinball Dreams/Fantasies/Illusions |
| - Team 17 | For Alien Breed, Alien Breed II, Tower |

- Assault, Superfrog, KingPin Bowling
and Arcade Pool.
- MicroProse For UFO: Enemy Unknown
 - Magnetic Fields For The Lotus Trilogy
 - Michael Bjornberg,
 Kjell Cederfeldt
 & Martin Larsson For all your help with FidoNet...
 - Henrik Fonde,
 Jorgen Fritzsos
 & Mats Oscarsson For being the greatest of friends!

1.59 Section 6.3 -- History

6.3 HISTORY

- 28.7.94 (1.00) - First document, not released.
 - 2.8.94 (1.01) L Second document, limited release.
 - 3.8.94 (1.02) - Third document, not released.
Users Standards Group formed.
 - 3.8.94 (1.03) P Fourth document, public release.
 - 4.8.94 (1.04) - Fifth document, not released.
 - 16.8.94 (1.05) P Sixth document, public release.
 - 18.9.94 (2.00) - Bumped release version to 2.
Began work on rewrite of release 2.
Multiple keywords added to the document.
 - 22.9.94 (2.01) - Decided to be backwards compatible with
the dollar-sign keywords.
 - 27.9.94 (2.02) L Began work on the examples file, USG_Exam.txt.
MAJOR rewrite of most of the document.
Limited release (betatesting)
 - 3.10.94 (2.03) - Rewrote some parts of the document.
 - 17.10.94 (2.04) L Added %REV keyword.
Added multiple file support for %REP.
Limited release (betatesting)
 - 27.10.94 (2.05) L Rewrote and added to the document.
Limited release (betatesting)
 - 18.11.94 (2.06) P Second public release.
 - 26.1.95 (2.10) - FDS now Copyrighted to Peter Bornhall
Removed USG-stuff from document.
Major document cleanup.
 - 21.3.95 (2.20) - Slightly new style to the document.
-

- AmigaGuide version planned.
History now in a separate file.
Document using the Text Encapsulation Standard.
- 22.3.95 (2.21) L Finished rewrite.
Dates in history changed to <d>.<m>.<y> format.
Limited release (betatesting)
- 6.4.95 (2.30) - Another rewrite again.
Separated into USER and DEVELOPER docs.
- 18.5.95 (2.40) L Continued rewriting.
Added %EXT: and %PID: keywords.
Updated %FIL: keyword.
First AmigaGuide document.
- 27.9.95 (3.00) L Joined user and developer specs again.
Finished developer addendum.
Rewrote some parts of the user docs.
Grammar and spell check.
- 1.10.95 (3.01) - Minor changes.
- 27.10.95 (3.10) P Added %DIR: keyword.
Moved history to guide.
Third public release.

P = Public release
L = Limited betarelease
- = Internal use only

1.60 Section 6.4 -- Last words

6.4 LAST WORDS

Finally!! It has been almost a whole year since I began the rewrite of these specifications. Many changes, and many long periods of time where I did absolutely NOTHING creative with it. With this release I have now dropped support for the earlier releases. FDS-310 is the NEW standard!! Well, the format has not changed that much, but some of the old backward compatibility is gone, unless developers really WANT to support it.

Oh, by the way: Sory four anny speling erorr...:)

-Live long and prosper!

...Peter
