RTF2IPF Overview

This section describes some background information that may be useful when determining if RTF2IPF is the right utility for you.

Purpose of RTF2IPF

This utility has been created as the result of my desire to maintain both printed and on-line documentation for an internally developed utility at work from a single document. The general goal of the RTF2IPF.EXE program is to take a word processing document saved as *Rich Text Format* and re-tag the file with the IPFC compiler tags to allow the creation of an OS/2 .INF file. RTF2IPF is currently a command line DOS program (sorry but I run DOS/Windows at home where this was developed) that should be able to integrate well to a development make file.

Although this utility is being distributed as 'Freeware' I would like to know what you think of it. After trying it please send me a note to my CIS ID (74101,413) and let me know what you think. I hope my efforts will be useful for you too.

What you need to use RTF2IPF

All you need to use RTF2IPF is a word processor capable of saving its document (styles included) in Rich Text Format (RTF). All of my testing has been with Ami Pro v2.0 for Windows and Ami Pro v3.0 for OS/2. The document you are receiving was created using Ami Pro v2.0 for Windows. Any word processor that can save it's documents in RTF format should work. If you use a different word processor and develop a style sheet that you find useful please send it to me (see the section *Obtaining support* for how to reach me) and I will include it in the next version to be uploaded. To compile your on-line document you will need the OS/2 IPFC help compiler.

Disclaimers

RTF2IPF is really still a work in process. As you read through here you will see several places where I mention that a feature is planned or where I will even solicit your ideas about how a feature should be implemented. As a result of this utility still evolving, the current version uploaded here still has some debug code compiled in. As such it is possible that you may run into an assertion or other bugs. Although I believe everything to be correct I cannot guarantee it. If you do have a problem or a suggestion, please contact me and I will see what I can do to fix it.

Creating The Documentation in your Word Processor

This section describes how to prepare you documentation in your favorite word processor.

Styles

RTF2IPF is dependent upon your word processor correctly identifying style information in the RTF file when you save your document. The styles are used to identify how the text in your document should be tagged for the IPFC compiler. The most important use of styles is to identify which 'paragraphs' in your document are intended to be used as headings in the .INF file. Since the IPFC compiler requires that you make an orderly progression through the heading levels it is imperative that you are very diligent in setting up a style sheet that mimics the progression of styles that the IPFC compiler will accept. You must always keep in mind how the styles will affect your on-line document.

For example, this document was created using the Ami Pro style sheet supplied with RTF2IPF. Some of the styles present in the document and their uses are as follows:

Heading 1

The top level headings in the document. The 'paragraph' *Creating The Documentation in your Word Processor* is a *Heading 1* style. That style always gets tagged with the IPFC tags ':h1.'

Heading 2

This is the second level heading in the document. The 'paragraph' *Styles* is marked as a *Heading 2* style. That style always gets tagged with the IPFC tags ':h2.'

Subhead

The Subhead style is used how a style is traditionally used in a word processing document. It is simply used to re-use common formatting attributes, in this case bold text and a larger point size. The Subhead style was used for the *Heading 1, Heading 2,* and *Subhead* headings within the *Styles* heading.

Text attributes

Where possible, RTF2IPF tries to emulate common text formatting characteristics within the constraints of IPFC. Currently, the only attribute supported is **Bold**. In the near future I hope to have it supporting *italics*, underline, as well as all of the combinations of these attributes. Please try not to use different colors for your text as my current thoughts are to use the RTF color tags in defining hyper-links to other topics (although this is still the subject of some debate).

Paragraph formatting

The current version of RTF2IPF does not provide any paragraph formatting beyond the standard word wrap capabilities provided by IPFC. Although IPFC does allow for some formatting it is beyond the scope of the current version of RTF2IPF.

Tables

Tables are currently unsupported by RTF2IPF but are currently very high on the list for planned enhancements (I need them too!). The 'results are unpredictable' for what will happen if you give RTF2IPF a table to process.

RTF2IPF Project Files

This section describes the RTF2IPF project files.

Overview

The RTF2IPF project files contain information for RTF2IPF to use when converting the RTF tagged file to an IPF tagged file.   The file is an ASCII text file that take the format of:

```
Keyword=Value
Keyword=Value
```

By convention the RTF2IPF project files have an .IPJ extension.

Keywords

Each of the valid keywords are described in their own section below.   Although the keywords are shown in mixed case they are processed in a case insensitive manner.

ExampleStyle

Purpose

The ExampleStyle keyword is used to define the style to be interpreted as example tagged information.   The ExampleStyle keyword is required if you need the ':xmp.' tag to be created in the .IPF file.

Syntax

```
ExampleStyle=<stylename>
```

Parameters

stylename - The name of the style to be tagged as ':xmp.' in the IPF file.

HeadingNStyle

Purpose

The HeadingNStyle keyword is used to define the style to be interpreted as heading level N. Currently N can take the value of 1 - 9.   At least the Heading1Style keyword is required.

Syntax

```
HeadingNStyle=<stylename>
```

Parameters

stylename - The name of the style to use for the heading level N in the .IPF file.

IpfFile

Purpose

The IpfFile keyword is used to define the IPF output file to be created.   This keyword is required.

Syntax

```
IpfFile=<filename>
```

Parameters

filename - The name of the file to be created containing the IPF tagged source for the help file. The file name follows standard naming rules that allow the name to be a fully qualified or relative path name.

IpfTitle

Purpose

The IpfTitle keyword is used to define the title for the .INF file to be created.   This keyword is required.

Syntax

```
IpfTitle=<title>
```

Parameters

title - The title for the .INF file to be created.

RtfFile

Purpose

The RtfFile keyword is used to define the RTF input file.   This keyword is required.

Syntax

```
RtfFile=<filename>
```

Parameters

filename - The name of the file containing the RTF source for the help file.   The file name follows standard naming rules that allow the name to be a fully qualified or relative path name.

Example .IPJ file

The following is the .IPF file used to create this document.

```
RtfFile=rtf2ipf.rtf
IpfFile=rtf2ipf.ipf
IpfTitle=RTF2IPF Reference
Heading1Style=Heading 1
Heading2Style=Heading 2
Heading3Style=Heading 3
ExampleStyle=example
```

Running RTF2IPF

Running RTF2IPF is very simple.   The program is invoked from the DOS (or OS/2) command line as follows:

```
   RTF2IPF <project file>
```

There are no options just a project file on the command line.   RTF2IPF will print a banner line and print a progress '.' at approximately every 500 characters added to the .IPF file until complete.

Obtaining support

If you have a suggestion or problem please send a description along with the .IPJ and .RTF files and the date on the RTF2IPF.EXE file to:

Mark Ault

74101,413 (Comp-U-Serve)

Planned Enhancements

This section contains a list of possible enhancement I am currently considering for RTF2IPF. They are listed in no particular order. If you have some favorites or additional ideas please let me know and I will see what I can do. Currently they are being prioritized base on how badly I need the feature and how long it would take to implement.

Character formatting

As mentioned earlier, one of the first enhancements on tap for this utility is the ability to use text properties other than normal and bold. *Italics* and underline will be added along with the various combinations.

Tables

Currently this utility is un-tested with tables. Table support is planned for the next version I will post.

Margins

One of the things that can make a help files contents more attractive is to control the margins. These will most likely be controlled through style modifiers in the .IPJ files.

Lists

The IPFC compilers supports several flavors of lists. The intention is to support at least the basic list type such as ordered lists, simple lists, and unordered lists. Once again, the various list types will be indicated through the use of styles.

Hyper-links

The ability to create hyper-links from one of the .INF panels to another panel in the same document. Currently I am planning on using a color or footnote tag to identify where the hyper-links occur (although this is still the subject of some debate). If you have any suggestions of a different way to tag it in the word processor please let me know.

Multiple RTF input files

As some documents get larger it can be convenient to break them into several smaller documents. When this is done it requires you to combine the documents as one and save the one RTF file for Rtf2Ipf to process. It would be more convenient to use several RTF files for each of the pieces of the original document.

Re-work the .IPJ project files

Possible enhancements include adding comments, an include directive, getting file names from the command line instead, and getting the title from the document information section of the RTF file.

Add a user interface shell

At the very least the shell would be useful for creating the .IPJ files. However, the ultimate goal of the shell would be to act as an integrated 'workbench' for all of the tools you would use to create your on-line documentation. From the workbench you would start you word processor to create the initial source for the on-line book, run RTF2IPF and view the output, run the IPFC compiler to create the .HLP or .INF file, and even start VIEW.EXE to view the newly created .INF file. Obviously this is the most ambitious of the enhancements listed here and the time to design/create this tool will dictate when/if it appears. Doing this may be mutually exclusive with the previous 'Re-work the .IPF project files' enhancement.

Create panel resource ID's and #defines

The resource ID is required for creating a .HLP file that is accessed from a Presentation Manager application.   This is another of the features that I will be needing within the next few months so look for it to be available relatively quickly (I'm hoping by Sep/Oct 1994 but make no promises!).

Window help file support

As I learn more about the MS Window help compiler I will be adding support for maintaining a MS Window help file in the same way you can use RTF2IPF to maintain an OS/2 help file.