To: I&T Group
From: Charles E. Gray
Subject: SPSS enhanced RIFF

Introduction

A number of files used in the integration and test of the SPSS are required to conform to the RIFF (Resource Interchange File Format) standard, as modified for SPSS. I refer to these as RIFF files.

The RIFF standard is extremely flexible. As a result, it is replete with pitfalls for the unwary. This note is intended to help avoid some of them by, first discussing salient aspects of the RIFF standard, then, illustrating these by means of a sample SPSS enhanced RIFF file.

Chunks

The notation used is that of [1], pp 11-21. A RIFF file is a specific instance of a data structure called a *chunk*. A *chunk* consists of exactly the following three elements:

1. A four-byte chunk identifier, denoted *ckID*
2. A 32 bit unsigned integer, denoted *ckSize*, whose value is equal to the value of the size of the third element (see 3. Below)
3. An element of binary data, denoted *ckData.* As noted in [1], *ckData* is word-aligned with respect to the start of the RIFF file; in other words, the first byte of *ckData* occurs an even number of bytes after the first byte of *ckID*. Moreover, if *ckData* contains an odd number of bytes, it is padded with a byte with value 0. This byte, if present, is not counted in *ckSize.*

The notation for a chunk used in [1] is:

<*ckID*> (<*ckData*>)

Neither the *ckSize* nor the pad byte (if present) are represented in this notation; nevertheless, they are present in the file (see Figure 3.).

Forms

In addition to the generic chunk described above, the RIFF standard defines a specific kind of chunk called a *form.* A form meets the requirements 1-3 above. In addition, the first four bytes of the *ckData* of the form are ASCII characters (padded on the right with blanks, if necessary) which specify the form type.

A RIFF chunk is a chunk with the *ckID* 'RIFF'. A RIFF form has, in addition, one of several form types registered with Microsoft Corporation.

As the RIFF standard is presently defined, a RIFF chunk is one of only two chunks whose *ckData* can consist of one or more chunks (called subchunks, naturally enough). The other kind of chunk whose *ckData* can consist of subchunks is the LIST form. This is a form with *ckID* 'LIST' and various form-types, some of which are registered (their form-type consists of all capital letters), others which can be defined by the user (their form-type consists of all lower-case letters). Thus it can be represented as:

'LIST' (<list-type> [<chunk>...])

where [<chunk>...] denotes the presence of 0 or more chunks.
An important LIST form is the one with form-type INFO. This LIST form is discussed in [1] pp 24-25. It is worth mentioning that the chunks that it contains are null terminated text strings. Also note that, since the ICMT chunk is intended to store comments about the file, it may be several sentences in length; these sentences should be ended with a period and **not** with a newline.

The SPSS enhanced RIFF specification contains a LIST chunk with form-type 'spss' (note the lower case) which is defined in Appendix B of [2].

Note that, when an application encounters a chunk whose *ckID* it does not recognize, it can (and should) skip over it using the information contained in the *ckSize*.

Example of an SPSS enhanced RIFF file

This section presents an example of an SPSS enhanced RIFF file.

Figure 1. represents the file in the Backus-Naur form. This is used to construct RIFF file definitions (for an example, see [1] p 21, <GOBL-form> definition). It defines each element of the file in terms of elements that are immediately subordinate to it. These are then defined in like manner.

This form of representation is well suited to the task of defining the file; however it does not exhibit the file hierarchy explicitly so that it is difficult to construct the binary file directly from it. For this reason, a tree representation is used (see for example, Figure 2 or [1] p. 22) which does exhibit the file hierarchy explicitly.

In Figure 2, the entire file is contained in the RIFF chunk which starts at the top of the Figure. The scope of this chunk is indicated by a vertical line descending from its *ckID* and ending with a right parenthesis. Nested within this chunk, with their scopes indicated in like manner, are a LIST 'INFO' chunk, followed by a LIST 'spss' chunk and a WAVE data chunk, as required by the definition of <Sample SPSS data file> in Figure 1. Notice that the WAVE data chunk is a RIFF form with the registered form-type 'WAVE'.

In order to facilitate the comparison of like elements in Figures 1, 2 and 3, the start of each chunk is enclosed in a box with light grey shading; the form-type ID is enclosed in a box with darker grey shading.

From this it can be seen that chunk beginnings and form-type ID's in Figure 2 occur on successive lines down the page in the same order as they do in the binary file depicted in Figure 3.

Refer to the first line of Figure 3: the first column lists the memory address, 0A96:0100, of the first byte on that line. The second column lists 8 bytes in the order in which they occur in memory starting with the byte whose address appears in the first column. The third column consists of a hyphen followed by the next 8 bytes. The fourth column consists of 16 characters corresponding to the 16 bytes in columns 2 and 3. If a byte represents an ASCII character, that character is printed; otherwise a period is printed.

In Figure 3, the first 4 bytes of a chunk beginning (light grey box) are the ASCII characters which spell the *ckID*. The next 4 bytes are the *ckSize*, expressed in hex digits in the "Little Endian" byte reversed notation peculiar to Intel machines.

Thus, for example, the *ckSize* for the top RIFF chunk is 182H (expressed in Little Endian style as 82 01) bytes. The decimal equivalent of 182H is 386 which is the decimal number of bytes from the 9^{th} byte (ASCII 'L', 4C Hex) to the last byte in the file (ASCII 'G', 47 Hex) inclusive.

Notice the occurrence of null pad bytes (ASCII null, 00 Hex), as indicated by the darkest shaded box. For example, a null pad byte is appended to the null-terminated string in the ICRD chunk to ensure that the next chunk, ICMT, starts on an even byte, as required by the RIFF standard.

References

[1] Multimedia Programming Interface and Data Specifications 1.0
IBM Corporation and Microsoft Corporation
August 1991

[2] SPSS Integration and Test
Test Software Requirements
25 June 1998
rev 1.2

```
< Sample SPSS data file > ->

    'RIFF'(<LIST 'INFO' chunk> <LIST 'spss' chunk> < WAVE data chunk >)

<LIST INFO chunk> ->

    'LIST'( 'INFO'
            <INAM chunk>
            <ICRD chunk>
            <ICMT chunk>  )

<INAM chunk>  ->

    'INAM'("C:\Program Files\DevStudio
            \MyProjects\cpanalyz\Debug\cpanlz01.in"Z)

<ICRD chunk> ->

     'ICRD'("19980730"Z)

<ICMT chunk> ->

      'ICMT'("Test CP output used in developing CPANALYZ"Z)


<LIST 'spss' chunk> ->

      'LIST'('spss'
            <dtty chunk>
            <tstp chunk>)

<dtty chunk> ->

      'dtty'('CPO ')    // input description

<tstp chunk> ->

      'tstp'("CPANALYZ development")// ditto

<WAVE data chunk>  -> '

     RIFF'('WAVE' <format chunk> <data chunk> )

<format chunk> ->

      'fmt '(
             wFormatTag // 0X0001
             wChannels // 1
             dwSamplesPerSecond // 0
             dwAvgBytesPerSecond // 0
             wBlockAlign // 1
             wBitsPerSample // 32 )

<data chunk> ->
      'data'( <data samples>... // "THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG"
                                // repeated 3 times ( = 132 frames))

                                    Start of chunk ( ckID + ckSize)

                                    Form type
```

Figure 1. Sample SPSS data file (BNF format)

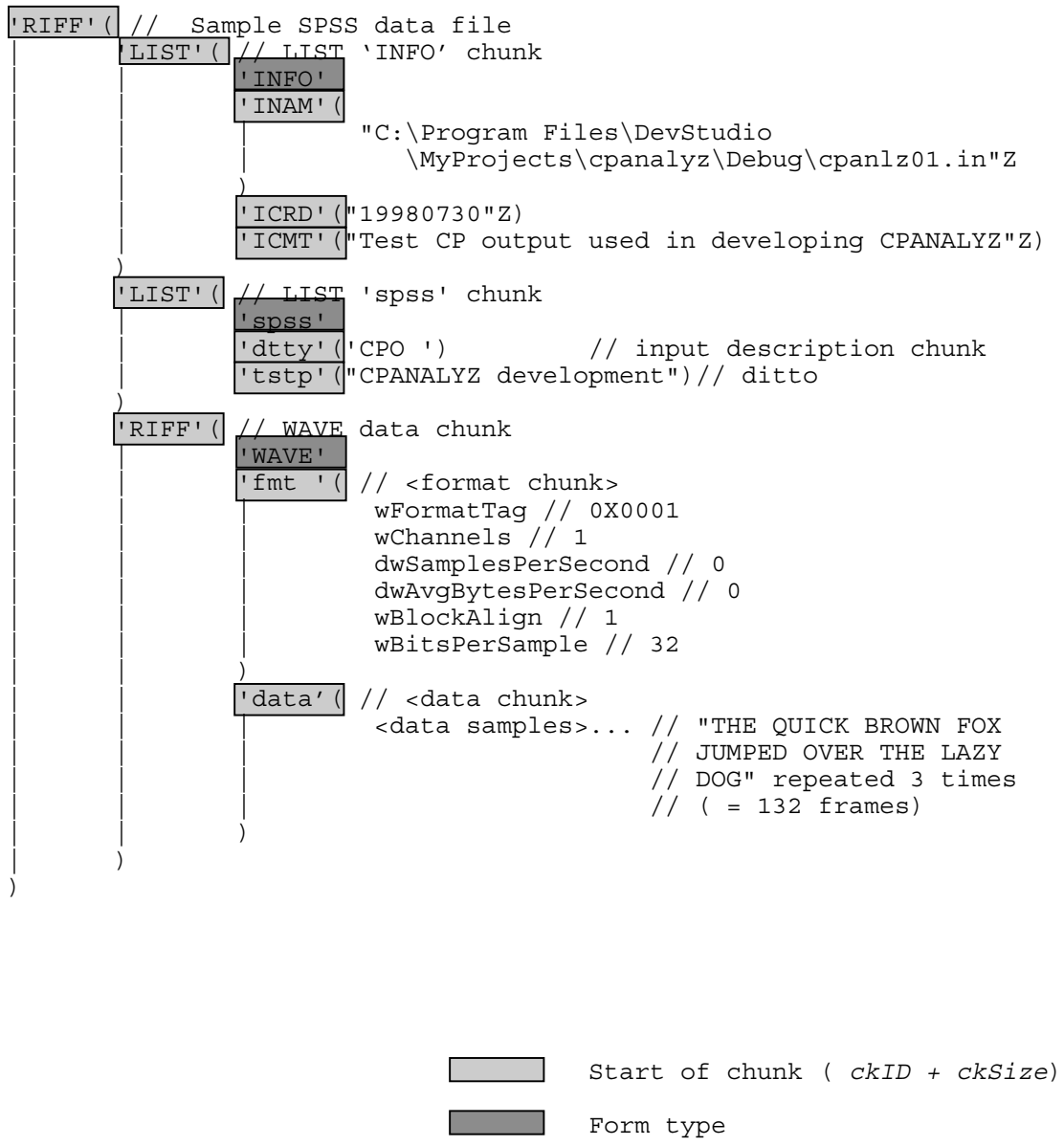

Figure 2. Sample SPSS data file (Tree format)

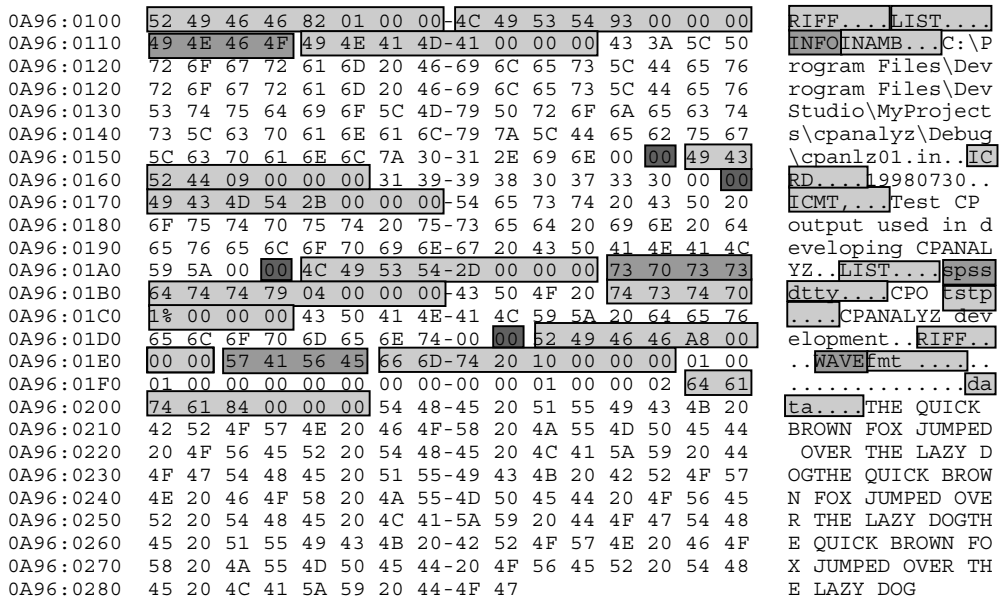

Figure 3. Sample SPSS data file (binary format)