CasketRtfConv

Much modified from HostTextConv.odc by RDC February 2006.

Moved from *CpcRtfConv* to *CasketRtfConv* by RDC February 2010.
Additional capabilities added; eg arbitrary Views are now copied as Bitmaps into the RTF code.

The main documentation file for this Module is Casket/Docu/RtfConv-Quick-Start.

This file explains some of the main changes from the old (BlackBox 1.5) version of HostTextConv.odc.

The main changes (February 2006) **are**:

1	Unicode support added using the	\u	codeword.
2	Charactersets recorded using the	\fcharset	codeword.
3	Code tables used on reading with the	\'nn	codeword.

Various mappings from 'Character Set's to 'Code Table's are hard coded. I suspect that the procedure WinApi.TranslateCharsetInfo should provide a general mechanism, but I can't get it to work.

4 Font 0 colour corrected. Was Ports.black, now is Ports.default.

5 Writing of the \'nn codeword corrected. The Hex number now uses lower case.

6 Problem reading the ruler attributes: \li & \fi.
They were expected in the order \li then \fi.
Word & Wordpad use the opposite order.
Now both orders are interpreted correctly (provided each is only sent once!).

7 Implements the '\field\fldinst SYMBOL nnn \fn' construct that Word 97 uses to output 'Symbol' characters. This is based much more on guesswork than knowledge. But works!

8 The biggest difficulty is the paragraph formatting logic.

The old BlackBox code is complicated, I do not understand it, and it does not work well.

My first attempt was simply as follows:
Replace:

	Rulers	*ruler*	(^J)	with	\pard
	Paragraphs	*para*	(^M ~ 0EX ~ FF)	with	\par
&	Carriage returns	*line*	(CR ~ 0DX)	with	\line.

This worked perfectly for BlackBox -> BlackBox, reasonably for BlackBox -> Wordpad, and badly for BlackBox -> Word.

To find a reasonably compromise ad hoc extra complexity has been added on a trial and error basis as follows:

On writing RTF:

All consecutive 'format control' characters are scanned (these *rulers*, *paras*, & *lines*), except that the scan stops when a page breaking Ruler is found.

The following information is collected:

- No of *line*s before the last *para* or *ruler* (crPreCnt)
- No of*line*s after the last *para* or *ruler* (crProstCnt)
- Attributes of last Ruler scanned.

The output logic is:

If the scan is terminated by a page breaking *ruler*:

- Output MAX(crPreCnt + crPostCnt, 1) '\par's, a '\page', & a '\pard'.

If the scan is terminated by any other character (visible, or space, or tab):

If no ruler was scanned:

If the last character scanned was a *para* (crPostCnt = 0):

- Output MAX(crPreCnt, 1) '\par's.

If the last character scanned was a *line* (crPostCnt > 0):

- Output crPreCnt + crPostCnt - 1 '\par's and 1 '\line.

If at least one *ruler* was scanned:

If the last *ruler*'s attributes equal the those in force before the scan:

- Output MAX(crPreCnt + crPostCnt, 1) '\par's.

If the last *ruler*'s attributes are new:

- Output MAX(crPreCnt, 1) '\par's, a '\pard', & crPostCnt '\line's.

On reading RTF:

- '\lines' are interpreted as *line*s, which are inserted into the output Text.
- '\par's are interpreted as *line*s, which are inserted into the output Text.
- '\page's are interpreted as *ruler*s.

If the last item output is a *ruler*

it's pagebreak option is set.

otherwise

a new (pagebreak) *ruler* is output.

- ► '\pard's are interpreted as *ruler*s.

    If the last item output is a pagebreak *ruler*

        all it's other attributes are cleared to the default state

    otherwise

        a new default *ruler* is output.

- ► *para*'s are inserted when visible text (or space or tab) immediately follows a \par.

- ► The completed text is then scanned:

    repeated, non-pagebreak *ruler*s are replaced by *para*s.

- ► The completed text is then scanned again and unnecessary *para*s are deleted.
  A *para* is considered unnecessary unless the previous *ruler* has non-zero left attribute.

9 Added, and ignore, the code word \loch, which is required to read OpenOffice RTF.

10 Added the code word \ulnone (required to read OpenOffice RTF). It turns off underlining.

11 The reader has a 'Debug' mode.
If the Alt key is pressed the RTF code is displayed untranslated.

12 Some (minimal) support for table formatting. Seems to have some benefit when importing simple tables from XP Wordpad.

13 While the early RTF standard says that all keywords are all lowercase, it now seems to be necessary to also recognize (and currently discard!) some uppercase ones.
Examples are : \trftsWidth2, \clftsWidth, & \clwWidth5000.

14 Links (HYPERLINKS) are imported & exported. Works fine for BlackBox -> BlackBox. Comments please!

The main changes (February 2010) **are**:

15 The scan to remove unnecesary paragraphs was corrected.

16 Arbitrary BlackBox Views are converted to Bitmaps, and embedded in the RTF code as Pictures.

While this module can read these pictures back, and reconstruct Bitmap Views into BlackBox Documents, it cannot import Pictures from Microsoft Word or Wordpad. This is because those programs do not export Pictures as Bitmaps, but rather encode them into Metafiles.

This module cannot read Metafiles.

The following View types are omitted: *StdHeaders.View*, & *StdLinks.Target*.

17 Additional characters supported (eg *enspace*, *emdash*, etc).

18 When TABs are output from BlackBox the *underline* and *strikeout* styles are turned off. They are not turned off for spaces.

Known problems:

1 There are a million other RTF codes ignored by BlackBox.

2 Not very compatible with Outlook express.

3 Binary data (eg embedded objects) is not deliberately skipped; hence can cause arbitrary disruption!

4 Some (not all!) versions of Word get the sizes of Bitmaps incorrect. The solution is to open (or paste) the RTF into Wordpad, then copy it from Wordpad to Word.

5 In links (eg <Dialog.Beep>Beep<>) make sure that either both Links (ie '<Dialog.Beep>', & '<>') are underlined, or neither. If you mix them *CasketRtfConv* can get confused!

6 Word and Wordpad handle Tabs differently to BlackBox. In Word each Tab takes you to the next Tab stop from where you are. In BlackBox the n'th Tab takes you to the n'th Tab stop irrespective of where you are.
In practice this means that some Tabs need to be deleted from Word, or inserted into BlackBox, depending on what direction you are translating.

To install this file: please refer to Casket/Docu/RtfConv-Quick-Start.

Robert D Campbell
robert.campbell_@tiscali.co.uk
25 February 2006
Updated: February 2010