

# Review of InDesign CS

Wayne Dirks – May 2004

Last year I tackled the challenge of typesetting the Giriyama New Testament trial edition in InDesign version 2 and was impressed with the results. With the release of InDesign CS (and another 30-day trial version) I decided to try typesetting the Pokomo New Testament with it.

This review is based on about 100 hours of using InDesign CS and countless hours of using Ventura Publisher 10. The publication I typeset with ID was a complete New Testament single column, no hyphenation with about 350 footnotes, 40 pictures and 4000 cross references in a side margin. Following is a list of observations and challenges I faced and how I worked around them. Please note that VP stands for Ventura Publisher 10.397, ID2 is InDesign 2 and IDCS is InDesign CS (3.0).

Adobe has released a 3.01 update for InDesign but I only had the trial edition and was not able to use it. Some of this review is the same as my review of InDesign 2. I did not have time to investigate if all the problems with version 2 were solved in version 3.

## Chapter Numbers

IDCS can't control the breaks before and after a paragraph like VP can. Drop chapter numbers were done with a 2 and 3 character drop cap paragraph tag. My cc table would tag the paragraph according to the number of chapter digits and put in a 0x2009 THIN SPACE after the chapter number. I had a "Par Chp 1" tag for single digit chapters (1-9) and a "Par Chp 2" tag for two digit chapters (10+).

The size and drop of the drop cap is determined by the character immediately following the drop cap. This becomes a problem when verse one is printed (or a verse bridge) immediately after a chapter number because the drop cap is screwed up by the size of the vs number. To fix this I inserted a 0x200A HAIR SPACE between the drop cap and the verse bridge. 0x200B ZERO WIDTH SPACE might have been a better choice, though. Or I could have used a character tag for the drop cap... hmmm. We'll try that next time!

One really cool thing with IDCS is that consecutive paragraphs will indent properly around a drop cap from a previous paragraph. So, in Acts chapter one where the first paragraph is only a few words, (Dear Theophilus...) IDCS will indent the second paragraph around the first paragraph's drop cap. Amazing!

## Hyphenation

In ID2, language set to "no language" will allow only discretionary hyphens (0x00AD). The font used must have the same glyph at 0x00AD (soft hyphen), 0x2010 (hyphen) and 0x2011 (no break hyphen) otherwise it will substitute glyphs and the optical kerning gets messed up when printing to a postscript printer. This was very strange until I finally figured it out. I did not have a chance to check this out in IDCS.

## Headers

IDCS can't reference a paragraph tag in the page header like VP. To do headers with chapter numbers I set up a master page with a section marker in it. Then for each book of the NT I would cut the \h from the main text, start a new section on page one of each new

book and paste the \h as the section name. Then, before printing I would manually type in the chapter numbers into the header for each page. With ID when you ctrl+shift click on a master-page frame it is converted to a normal page frame. This may seem like an incredibly time-consuming job but I'm pretty fast with clicking and typing. It would be nice if a section marker could be imported with text but I couldn't figure out a way to do it.

### **Footnotes**

IDCS doesn't do automatic footnotes. I had to place all the footnotes manually. To place threaded frames in InDesign I had to click on the red + sign on the previous frame and then click where I want the next frame to begin. InDesign would then draw a new frame at the click point the full column width, similar to VP. Before placing the frames I turned on text wrapping with nothing selected. This then becomes the default for all new frames.

### **Verse numbers**

I use a special font (a number for each character) for my verse numbers so that they don't get pulled apart by justification. Importing text is a lot faster if character tags are used rather than markup. I chose not to use superscript for anything because the size and jump is universal for all paragraph tags in the document. IDCS can't do different types of superscript for verse numbers or footnote markers. They're all the same.

### **Speed**

Overall, ID is very slow compared to VP. With all the text formatting features turned on IDCS becomes a *slow pig*. Optical kerning seemed to be the biggest culprit. It's a memory hog too. After working on a NT for awhile in IDCS I discovered it to be consuming 200+mb of virtual memory. With only 256mb of RAM I spent a lot of time listening to my hard drive churn away. Using character tags rather than markup makes importing text a lot faster.

### **Special characters - Unicode**

The Pokomo language used a 0x0323 COMBINING DOT BELOW and 0x032F COMBINING INVERTED BREVE BELOW. Even with the latest DoulosSIL 4.01.2004 the combining marks weren't centered properly in all cases. I wanted to use Charis typeface so I ended up making a custom font anyway. (Long live TypeCaster!)

### **Importing/Exporting Text**

- Importing tagged text with markup works as long as a matching tag is already defined in the IDCS document. If not, IDCS will remove all the tags on importing. I setup my cc table to insert a section in each text file that would minimally define the tags used in the document. That way none of my tags were removed in the early stages of design. Later this became unnecessary as my tags were already in the document.
- Autoflow only works when you import text. If you add pictures or footnotes and push text further back it doesn't make more frames and add pages at the end of the book.
- After typesetting was all finished, I exported each book back out to plain text with markup. Then I ran the text through a reverse cc table that converted it back to SFMs. Then I used CSDiff to compare the original SFM file with the reverse SFM to see if

anything was altered.

### **Adjusting Text**

- Tagging text with + or - tags can only be done with ctrl, shift or alt variations of the keypad. This was a bit more inconvenient than VP.
- Tracking adjustment in justified text changes character and word spacing appropriately. VP doesn't do a good job with this. In VP, if you adjusted the tracking the character spacing would increase more than the word spacing. IDCS does both appropriately and it looks nicer. VP tracking also had problems with the last character of a line of justified text not being against the frame. IDCS does it properly.
- IDCS breaks text at : (colon) which was annoying. I fixed this by inserting a `<cNobreak:l>` around the colon with my cc table.
- Optical kerning looks very nice. In ID2 there was one instance where the optical kerning didn't look good (between o and l) but this seems to have been fixed with IDCS. Kerning can be applied with a cc table and imported with the text. ID doesn't offer custom pair kerning within the application like VP.
- Paragraph tags can be forced to the baseline without having to calculate leadings, margins, before and after spacing and all that.
- IDCS doesn't have as much control over superscript attributes as VP. The size and vertical shift is stored with the document and applies to all paragraph tags. For verse numbers and footnote markers I used a character tag with a vertical jump rather than superscript.
- IDCS's vertical justification works better than VP's. ID will apply the same amount of justification to each paragraph tag in the column (as long as the tag allows it).

### **Placing Picture Captions**

With nothing selected click on the picture caption paragraph tag. This sets the default paragraph tag and all pasted text after this will be formatted with that tag.

### **Stability**

During the 100 hours of use, IDCS only crashed twice. Once while doing a save-as and the other when cancelling a print job. Both times I was able to restart IDCS and keep going.

### **Help**

I found that the HTML help text was too small for me. I change style sheet to make it bigger

### **Book Setup**

I chose to setup the NT as one big file. At first it was difficult to navigate until I discovered ID bookmarks. After creating a bookmark for each book of the NT the navigation was much easier. IDCS uses a dockable bookmark window that lists all the bookmarks for the document. Double-clicking on the bookmark takes you to the bookmark location.

### **Screen**

- Anti-aliased text looks much better than VP but it slows down the application a bit. The anti-aliasing is very similar to what Acrobat offers.
- ID's toolbars are intuitive and economical on screen space. All of the toolbars can be grouped and docked on the sides and rolled away similar to VP. You can't create your own toolbars or edit them.
- Story editor shows styles in the left margin. It can use custom fonts but it doesn't show markup codes.
- No status bar that tells special characters like disc hyphens and no-break things.
- No right page or left page specific tags/properties
- Snap to baseline only works when it's visible

### **Printing**

- ID can print spreads (2-up) and other basic printing but nothing compared to VP's powerful impositioning tools. We use FinePrint to do simple booklet impositioning and it worked fine. Both ID and VP can output directly to PDF.
- PDFs made directly from IDCS wouldn't print to our Lexmark Postscript level 2 clone. It only gives "Startdata" errors. I had to make ps files and distill them to pdf.
- Not sure why but when printing choose subset font instead of complete solves some problems (ANSI 48, 49 and ...)

### **Quirks**

Overall there were very few times when IDCS would do something strange. It almost always did exactly what I wanted it to do. But in my time using it I observed the following strange behaviour:

- Justification of first line in a continued frame sometimes got messed up. Fixed by resizing the frame.
- Footnote paragraph tag line-above not the proper length, fixed by reapplying the tag.
- Tab at top of frame of continued frame doesn't always display.

### **Other**

- Hourglass mouse is not used consistently. ID just sits there busy and you think it's crashed or something.
- There's no context-sensitive help.
- Right clicking isn't used very well.
- No keyboard shortcut to override snapping while moving/resizing an object

### **Summary: Things I like better about InDesign CS**

- Paragraph composer, glyph scaling, optical kerning make the end product look beautiful
- Anti alias text looks better on screen
- Baseline can be viewed on screen
- Force paragraph tags to align to baseline without having to do all the math
- Vertical justification applies the same amount to all tags on a page
- InDesign can lock column guides
- SMALL CAPS works like MS Word
- Selective linking and updating of externally referenced files
- Paragraph tags and Master Pages are linked (changing one changes them all)

- More stable and reliable than Ventura 10
- Support for unicode, though I wasn't impressed with the combining diacritic that I needed.

**Summary: Things I like better about Ventura Publisher 10**

- Select and edit multiple paragraph tags
- Customized toolbars
- Automatic running headers (though not very reliable)
- Automatic footnotes (though not very reliable)
- Customized pair kerning
- More control over paragraph breaks
- Copy editor lets you view markup codes
- Underlying page makes adding text and frames easy
- Faster at everything
- Paragraph styles that span multiple columns
- Built in impositioning
- Context sensitive help

**Conclusion**

Overall I was very impressed with InDesign CS. It's more stable and reliable than Ventura. With the use of bookmarks the challenge of navigating a long document is made much easier. The text handling (multi-line composer, glyph scaling and optical kerning) of InDesign is superior to VP and gives paragraphs a more uniform spacing, making it easier to read. These extra features slow down the program compared to Ventura but the end result is worth it.

My test computer was:

- Windows 2000 sp3
- 1.6ghz Athlon XP
- 256mb DDR RAM
- 20gb 7200 rpm hard drive 2mb cache