

Written, serialised, re-read, rendered

This page was not rendered from the document that built it. It was built, written out as PinataLayout DDL with DdlWriter, parsed back with DdlReader, and the copy that came out of the parser is what the renderer was given. Anything the round trip lost would be missing from this page.

Formatting survives too: **bold**, *italic*, underlined, and a colour set on a run rather than on the paragraph. **Firebrick.**

What	Written by	Read by
A whole document	DdlWriter.WriteString	DdlReader.DocumentFromString
One object	DdlWriter.WriteString(obj)	DdlReader.ObjectFromString
To a file	DdlWriter.WriteToFile	DdlReader.DocumentFromFile

The DDL it went through

257 lines, 3,814 characters, and 0 error(s) reported by the reader. The first eighty lines follow. The format is PinataLayout's own: braces nest, an attribute is name colon value, and a paragraph's text is written between its braces.

```
\document
[
Info
{
Title = "Ddl"
Author = "PdfPinata SampleApp"
}
]
{
\styles
{
Normal
{
Font
{
Name = "Liberation Serif"
Size = 10.5
}
}

Heading1
{
ParagraphFormat
{
OutlineLevel = Level1
}
}

Heading2
{
ParagraphFormat
{
OutlineLevel = Level2
}
}

Heading3
{
ParagraphFormat
{
OutlineLevel = Level3
}
}

Heading4
{
ParagraphFormat
{
OutlineLevel = Level4
}
}

Heading5
{
ParagraphFormat
{
OutlineLevel = Level5
}
}

Heading6
{
ParagraphFormat
{
OutlineLevel = Level6
}
}

Heading7
{
ParagraphFormat
```

```
{
OutlineLevel = Level7
}
}

Heading8
{
ParagraphFormat
{
... and 177 more lines.
}
```

What survived

Sections	3
Elements in the first section	4
Styles defined	18
Does the Listing style survive	yes
Its font	Source Code Pro
Table columns	3
Table rows	4
Is the first row still a heading	yes
Border width	0.5
Document title	Ddł
Reader errors	0

Every number above was read off the document the parser produced, not off the one that was written - so this page is a round-trip test somebody can look at. DDL is worth knowing about for two reasons beyond serialisation: a document dumped to it is readable, which makes it the fastest way to see what a document object model actually holds, and a report template can be kept as text and filled in at run time rather than being written out in C#.