BNC Data Capture:
OUP format definition for text handover to OUCS

Jeremy Clear

Version 1.0 14th July 1992

1 Introduction

Text for the written component of the BNC comes to OUP via three main routes: data supplied already in machine-readable form, text keyboarded from paper original and text OCR scanned from paper original. Text received from these various sources is converted by a mixture of manual and automatic methods into a single target format which is described in this document and which is intended to allow for subsequent conversion by OUCS into the BNC-wide Corpus Document Interchange Format (CDIF).

The primary data capture is carried out over a long period of time and OUP's conventions for encoding features of written text have changed over time to adjust to new demands and to reflect the lessons of experience. In the case of texts which are submitted to OUP in machine-readable form, the BNC project clearly has no control over the encoding of original textual features and certain information about the original cannot be retrieved. Particularly in the case of electronic text sources, the notion of the "original" source material is often weak and evaluative judgements are made in the course of conversion to BNC format about the necessity in relation to the cost and effort of restoring or decoding particular features of individual texts.

This document contains two further sections: first some notes describing general coding conventions and file format of OTF and secondly a description of the use and meaning of all tags and codes used in OUP's target format (hereafter "OTF") arranged thematically.

Encoding follows SGML-like conventions, though it must be admitted that the investment of time and intellectual effort in becoming fully conversant with the detailed technicalities of SGML was not considered to be worth making and the syntax of OTF may occasionally deviate from SGML. In particular, no attempt has been made to describe the OTF with a Document Type Declaration.

2 General File Format Conventions

OTF uses only the 96 printable characters (octal 040 through 176) plus TAB (octal 011) and LF (octal 012) from the ASCII character set.

Multiple contiguous LF (012), SP (040) and TAB (011) chars are not significant and are functionally equivalent to a single space (SP) character.

Non-textual material in the original is not captured and encoded. Tables which contain a significant amount of text may be captured and encoded with the `<table>` tag (see below). Photographs, tables, and other figures are noted as having been present in the original only where such indication is deemed by the data capture staff to be necessary to the continuity and intelligibility of the text.

At the top of each text sample (or collection of text samples in the case where a number of small samples have been gathered within one file) will be a `<text>` tag, immediately followed by a `<head>` which will be some identifying title of the text sample. The content of this `<head>` is free format. An optional `<note>` tag follows which contains a record of the page range captured in this sample. If the whole text is captured, then no page range will be indicated.

3 Code Summary

3.1 Basic Structure: written text

`<text>` (no attributes)

This tag is always present to mark the beginning of a corpus unit. Such a unit is the data contained within a file, having a particular six-character identifier associated with it. It does not designate a discrete and integral textual unit—it merely signals the start of the real data. Only text which is included within the `<text>` tag should be considered part of the language material of this corpus unit. The closing `</text>` tag is used.

`<div0>` (no attributes)

This tag marks a major subdivision of a text. It is used to mark discrete textual units within a composite document. In particular, it is used to mark the extent of individual articles in a newspaper. It is rarely used otherwise. No closing `</div0>` tag is used.

`<div1>` (no attributes)

This tag is the primary marker of first level divisions within a text (e.g. chapter, section, magazine article). No closing `</div1>` tag is used.

`<div2>, <div3>, <div4>` (no attributes)

These tags mark nested divisions subordinate to `<div1>`. They do not have any precise meaning in terms of parts, sections, subsections, chapters, addenda, appendices, etc. since the reality of printed text is vastly more complex than a simple, four-level hierarchy could accommodate. Subdivisions should be consecutive and nested: i.e. a `<div2>` cannot appear without a prior `<div1>` and the appearance of a `<div2>` forces termination of any preceding `<div3>` or `<div4>`. No closing `</divN>` tag is used.

3.2 Paragraph-level elements

`<p>` (no attributes)

This tag marks the beginning of a paragraph-like stretch of text. Usually the orginal paper text signals such units with initial indentation and/or extra white-space between such units. This tag may be used to indicate the beginning of a new stanza within verse passages, to marl boundaries between the cells of tabular structures containing text (see below: `<table>` tag) or for any other text grouping which is analogous to a standard prose paragraph. In some cases the paragraph-like unit is not precisely defined (as when headings, lists, tables and figures are interspersed within the text) and the judgement of the data capture staff is sometimes called upon in the application of this tag. `<p>` are defined to be sequential: they cannot overlap nor nest. No closing `</p>` tag is used.

`<head>` (no attributes)

A title or heading. There is no constraint on the `<head>` tag to appear only at the beginning of a marked structural division. A `<head>` tag may appear only within `<text>`, `<divN>`, `<poem>` `<hi>` or `<q>`. The appearance of `<head>` can be taken therefore to signal the close of any immediately preceding `<p>`. The `<head>` tag cannot be nested. Closing `</head>` tags are used.

`<ct>` (no attributes)

A caption. A stretch of text (usually shortish) which either relates to some table, figure, photograph or other non-textual element or which floats (like a "pull-quote") outside the sequence of the main body of text. Captions are often signalled in printed originals by typeface change and other highlighting. Texts which appear on the printed page to be divided into many levels of subordinate sections (i.e. those which seem to extend below `<div4>`) seem to blur the distinction between `<head>` and `<ct>` and both codes may be used at the discretion of the data capture staff. Captions may not nest. Closing `</ct>` tags are used.

`<poem>` (no attributes)

Delimits a poem or fragment of verse or song. If the fragment is only one line or less, then it is not possible to be sure that the passage is verse and in such a case the `<poem>` tag will not be used. Line breaks within a `<poem>` are marked by the `<l>` tag. No closing `</l>` tag is used. Part lines are not marked as such. No other specific structural features of verse (e.g. stanza, couplet, canto) are encoded, though `<p>`, `<head>` and `<divN>` tags may be used within `<poem>` where appropriate. Closing `</poem>` tags are used.

<list> (no attributes)

This tag delimits a list structure. Lists are composed of one or more sequential elements, each prefaced with a <enum> tag. The use of the <enum> tag in OTF is somewhat bizarre and is a vestige of an earlier confusion over the semantics of the TEI P1 <enum> tag. In OTF the <enum> tag takes two forms: either as an empty element or as a phrase-level element. If the items of the list are labelled with some enumerator then the enumerating label is enclosed within <enum> and </enum> tags. If the list items are not so labelled, then the start of each item is marked by the empty tag <enum>. Since simple sequence is the dominant underlying structure of almost all written documents, the <list> structure (with empty <enum> tags) could be applied to almost any stretch of discourse and there is sometimes no clear justification for the use or omission of the <list> tag. The subjective judgement of the data capture staff is called upon in such cases. Data capture staff are advised to use the <list> tag where the list item labels are apparent and conventional (e.g. the labels are single letters, roman or arabic numerals) or where, depite the absence of labels, the list items are very clearly signalled by the typographical layout. The closing </list> tag is used to terminate the list structure. Lists may be nested.

<table> (no attributes)

This tag encloses text which is laid out in the original in tabular format (and may therefore seem terse, disjointed or incoherent). Though it is normal practice to omit tabular material at the point of data capture, if the table contains a substantial amount of text (words, phrases or continuous prose) it may be captured and marked using this tag. Individuals "cells" of the table are separated from each other by the <p> (paragraph) tag, to aid readability. The <table> tag may not nest. Closing </table> tags are used.

<note> (optional attributes: source, place)

This tag is used primarily to enclose editorial additions (see below). The place attribute is very occasionally used with the <note> tag to mark a note the text of which is taken from the original. Notes may not nest. Closing </note> tags are used.

3.3 Addition, deletion and regularization

<note> (optional attributes: source, place)

This tag is used primarily to delimit extraneous text (i.e. text which is not present in the original). It is used freely to enclose meta-textual information that might be of interest or value to a human studying the corpus text, in which cases it will have an attribute source with the value ed. The <note> tag is also occasionally used to mark original text notes (see above). Notes may not nest. Closing </note> tags are used.

<del> (optional attributes: ed, desc)

The <del> tag explicity marks editorial deletions made during the data capture process. This tag is usually used to signal the excision of personal names, telephone numbers and addresses to preserve anonymity of private individuals. The <del> tag is used, in preference to silent omission, when the deleted text appears as a constituent in sentence structure. In such cases, were the omission to be silent, the encoded text would appear fragmentary and ungrammatical. The `ed` attribute identifies who was responsible for the deletion, and the `desc` gives some indication of the type of text which was deleted. The `<del>` tag is used as an empty element.

`<sic>` (no attributes)

This tag encloses words or phrases which, in the opinion of the keyboarder or post-editor is incorrectly spelled in the original source text. Such words are explicitly marked to ensure that they are preserved in the corpus (since such erroneous spellings may be worthy of study in their own right) and are not silently regularized during subsequent automatic spell-checking and post-editing. Of course, the `<sic>` tag will appear only when the keyboarder or editor *believes* that there is some dubious or incorrect form in the original: such belief may be unfounded in fact so that forms are marked which are correct by all usual standards, and conversely there may be actual errors which are not observed by the data capture staff and are therefore not signalled. No attempt is made to regularize the dubious or incorrect form enclosed with this tag. Sic tags may not nest. Closing `</sic>` tags are used.

`<reg>` (optional attribute: `sic`)

This tag is used to signal a regularized word form or phrase. The editorial regularization is enclosed with this tag, and the optional `sic` attribute records the original form. This tag may not nest. Closing `</reg>` tags are used.

3.4 Quotations and highlighted phrases

`<q>` (no attributes)

This tag encloses a block quotation. A block quotation is one which is set apart typographically from the main body of the text (usually with indentation, extra white space before and after or a different typeface). Quotation which is indicated only by the presence of speech marks will not normally be marked with the `<q>` tag. There are no constraints on the contents of a `<q>`. Quotations which run to multiple paragraphs will have `<p>` tags included within them. It is also acceptable for contiguous `<q>`...`</q>` passages to be used, even if the text thus enclosed form a single continuous quotation. Closing `</q>` tags are used.

`<hi>` (optional attribute: `rendition`)

This tag encloses text which is highlighted typographically and which is not given special status by virtue of being marked as a heading, caption, page number, or other encoded feature. If, for example, the enumerators of items in a list are printed in boldface, then the `<hi>` code need not be used since these piece of text are already encoded as special features. Only three types of highlighting are encoded: boldface, italic and underlining. The `rendition` attribute may take one of the following values: `italic` `bold` `underline`. Contiguous `<hi>`...`</hi>` phrases with identical rendition attribute values (though redundant) are acceptable. Closing `</hi>` tags are used.

3.5 Miscellaneous elements

`<pb>` (optional attribute: `n`)

This empty element signals that a page break appeared in the original text. The `n` attribute indicates the page number of the original.

`<date>` (no attributes)

This tag is very otional and encloses a date. It is used primarily in the encoding of newspaper text drawn from machine-readable sources in which the date appears at the head of every article. In such circumstances, it is desirable to isolate the date, since it would not appear in print and is so frequent that word frequency counts would be significantly affected if these date strings could not be identified automatically. Dates may not nest. Closing `</date>` tags are used.

`<label>` (no attributes)

This tag is rarely used and has the same meaning as the `<enum>` tag used as a phrase-level element. It encloses some list enumerator or label (typically strings such as "(a)", "xvii" or "2.") which might cause difficulty to an automatic tagging or parsing program and which occurs either within or outside a list structure. The presence of a `<label>` tag does not imply the presence of an enclosing `<list>`. Labels may not nest. Closing `</label>` tags are used.

`<salute>` (no attributes)

This phrase-level element may be used to enclose a formulaic greeting. Typically it is used for the conventional opening phrase of a letter (e.g. "`<salute>` Dear Ms Smith `</salute>`"). Salutes may not nest. Closing `</salute>` tags are used.

4 Special and non-ascii characters

Speech marks

The ascii grave accent (octal 140) is used whenever an opening speech mark occurs in the original. The ascii double inverted commas (octal 42) is used to represent closing speech marks. These codes are used (rather than preserving the original typeset mark) in order that no confusion arises between the apostrophe (occurring in possessives and enclitics, for example) and the single inverted comma (used as a speech mark). The ascii single inverted comma (octal 047) is reserved to represent the apostrophe. These codes are used in a way functionally analogous to SGML entity references for the typographic symbols for opening and closing quote marks—that is, they are not structural units, nor do they delimit structural units, they need not (though they usually do) form mutually complementary pairs and levels of nesting are not indicated in any way.

hyphen

End of line hyphens which split words across a line break on the printed page (apparent "soft" hyphens) are removed, except where the removal of the hyphen would yield an obvious non-word. Hyphens which occur other than at line ends ("hard" hyphens) are retained and the ascii hyphen symbol (octal 55) is used.

other special symbols

The following ascii symbols are used with the meanings shown:

SPACE	any amount of white space equal to or larger than an interword space
!	exclamation mark
"	closing speech mark (inverted commas)
#	unused
$	unused
%	percent symbol
&	SGML-style entity reference initiator
'	apostrophe
(	left round bracket
)	right round bracket
*	asterisk
+	plus symbol
,	comma
-	hard hyphen
.	period
/	oblique stroke
:	colon
;	semicolon
<	SGML tag initiator
=	equals sign
>	SGML tag terminator
?	question mark
@	unused
[	left square bracket
\	unused
]	right square bracket
^	unused
_	unused
`	opening speech mark (inverted comma)
{	unused
|	unused
}	unused
~	unused

Other special symbols are encoded using SGML-style entity reference notation. OTF entity references are initiated with the `&` symbol and terminated with a semicolon (`;`).

The following is a list of OTF-specific entity references:

`&and;`	an ampersand symbol (&)
`&1/2;`	the fraction one half (1 over 2)
`&3/4;`	the fraction three quarters (3 over 4)
	Note that all fractions are represented using this notation: `&numerator/denominator;`
`&yen;`	Japanese yen currency symbol
`&ft;`	the "foot" symbol (Imperial measure) (`'`)
`&ins;`	the "inches" symbol (Imperial measure) (`"`)
`&degree;`	the degree symbol (superscript circle)
`&subN;`	(where N is a letter or number) subscript characters
`&supN;`	(where N is a letter or number) superscript characters
`&formula;`	a mathematical formula of any complexity
`&shilling;`	UK shilling (e.g. `10/-` becomes `10&shilling;` and `10/6d` becomes `10&shilling;6d`)

All other entity references are taken from the publicly declared entity sets as follows:

- Accented Characters
- Diacritical Marks
- Greek Letters
- Numeric and Special Graphic
- Publishing

These sets are reproduced in Goldfarb (ed.) (1990) *The SGML Handbook*, Oxford: OUP, pp. 502ff.