



# Preservation Handbook

## [Marked-up Textual Data]

---

Author	Alan Morrison and Martin Wynne
Version	2
Date	22/08/05
Change History	Written by AM 11/10/04; revised MW 22/08/05, 9/11/05 and 7/03/06

---



## Definition

*Markup* is the name for the electronic tags which may be inserted in electronic text data. These markup tags are labels interwoven in the text to represent one or more of the following:

- descriptive information about the document structure;
- instructions about the appearance or formatting of the text;
- annotations which may be an interpretation or categorisation of certain textual elements.

*Marked-up textual data* is therefore electronic data containing both text and markup.

## Description

There are many forms of marked-up textual data encountered by the AHDS. These forms may be categorised in terms of encoding formats as follows:

- Following open standards designed for describing the structural role and meaning of elements in the document (SGML, XML)
- Following open standards designed for the display of the text (HTML, XHTML)
- Following open standards that combine description and presentation of text (LaTeX)
- Based on standards, but not fully following the syntax or semantics of the standard (non-standard HTML, pseudo-SGML)
- Various types of mark-up included within files encoded in proprietary formats (e.g. Microsoft Word, PDF);
- Open *de facto* standards accepted within a certain community (CHAT);
- Legacy formats, no longer widely used (COCOA, GML);
- Other non-standards-compliant data, often developed for the mark-up of the particular resource.

This handbook provides an overview of how to deal with resources created in these formats. The formats which conform to open standards are preferred for preservation purposes, and this handbook will therefore focus on their use. Saying that a format conforms to *open standards* means that its definition is controlled by a non-profit-making body, is publicly available, is not tied to any proprietary interests and is platform and software independent.

The international open standards relevant to textual markup are associated with *Standardised Generalised Markup* (SGML). SGML was certified as ISO standard 8879 in 1986. *Hypertext Markup Language* (HTML) was developed in the 1980s as an application of SGML for web pages. *Extensible Markup Language* (XML) was developed in the late 1990s as a simple 'dialect', or 'subset' of SGML for general use on the web and in exchanging documents between systems. XML has been designed for ease of implementation, and for interoperability with both SGML and HTML. It is now seen as the universal format for structured documents and data on the web, and for many other situations in which documents are encoded and exchanged.

Although SGML was used in arts and humanities computing, it often proved too complex for many scholars to implement, and coupled with a lack of user-friendly software packages, it did not have the immediate impact that was hoped for. As a result, various alternative, non-standard markup languages continued to be widely used after the establishment of SGML as a formal standard.

It is also important to note that although there exist open standards for ways to insert markup, and these standards are accepted in the domain of computer science, there is no formal, agreed acceptance of these (or any other) standards for the ways in which markup should be applied in the relevant academic fields in the humanities.

The Text Encoding Initiative (TEI) was formed to develop and promote guidelines for the consistent encoding of text using SGML for scholarly purposes, and has more recently promoted the use of



XML. It has a strong worldwide following and is used extensively in arts and humanities projects within the UK and abroad. The TEI produces the publication *Guidelines for Electronic Text Encoding and Interchange* (currently in transition from the fourth edition, known as P4, to P5) which explains how electronic texts can be marked up using the TEI system. This document describes SGML and XML data which was marked up following these guidelines as TEI-conformant, and others as non-TEI-conformant. The TEI does not in general propose only one way of implementing a particular encoding. It is possible to adapt and extend the guidelines in ways which are recommended by the TEI, and data which follows these guidelines may also be considered to be TEI-conformant.

The TEI guidelines cannot however be considered an international standard in the same way that XML is, as they have not been approved as a standard by any relevant body, although their work has been important for the development of the XML standard. For this reason the AHDS does not insist that marked up textual data is in TEI-conformant XML, although it is recommended for many types of data.

## Additional Information

TEI Home page (UK)

< <http://www.tei-c.org.uk/> > Last checked 06/03/2006

The Cover Pages, a searchable reference collection of SGML/XML resources

< <http://xml.coverpages.org/> > Last checked 06/03/2006



## Technical Environment

XML, HTML and XHTML are part of the SGML family of mark-up languages. SGML is a 'meta-language', which means that it is a language for defining markup languages. XML, HTML and XHTML are particular implementations of SGML-based mark-up languages. When SGML is referred to below, this refers to all of these formats which are part of the SGML family.

The main components of any valid SGML documents are:

- a Document Type Declaration (DTD), or an XML Schema, which define the syntax of the markup;
- a header, which can contain metadata describing the electronic text and its original source;
- content data, which, in the case of marked-up textual data, is the text interwoven with markup.

The DTD or schema is essential to any SGML text, as it defines the rules and structure of the data, and is required if the data is to be validated. Validation is carried out with the aid of a parser, which simply checks that the encoded data conforms to the rules as stated in the DTD or schema. Parsers are plentiful and freely available, and most SGML aware software packages have them built into their systems. The DTD employed by a text is usually declared at the beginning of the header or document, for example the TEILite.dtd can be declared as :

```
<!DOCTYPE TEI.2 PUBLIC "-//TEI//DTD TEI Lite 1.0//EN" "teilight.dtd">
<!DOCTYPE tei.2 PUBLIC "-//TEI//DTD TEI Lite 1.0//EN">
<!DOCTYPE TEI.2 SYSTEM "teixlite.dtd">
```

While an SGML file can be validated with a DTD which is stored at a remote location, for preservation purposes the DTD or schema should be archived along with the text, especially if the creator of the text has modified the DTD in anyway. Any modifications should be recorded within the DTD itself.

The header of an SGML text should provide the user of the text with sufficient bibliographical and technical details concerning the text and how it was created. The header **MUST** be kept with the text at all times.

XML is software- and platform-independent and there are a range of open-source and proprietary software tools available to create, process and validate XML encoded text. Large software vendors, such as Microsoft, have incorporated some of the features of XML into their most recent software versions, but these are not necessarily fully compliant with the standard. See "The Cover Pages" (cited above) for an up to date list of available tools.

Documents encoded with HTML markup can be problematic. While there are formal specifications for various versions of HTML, and documents can be tested for well-formedness and validity against a DTD, many HTML files are not well-formed and the extent of conformance to guidelines and specifications varies widely. Many HTML applications (such as web browsers) are designed to be extremely robust in their treatment of HTML, and will render documents with non-well-formed and formally invalid HTML. As a result, users have become accustomed to hacking HTML markup to achieve the desired resulting visual appearance in a particular application, without being aware of issues of conformance or the differences between descriptive and procedural markup.

XHTML is a form of hypertext markup which is XML-conformant. XHTML documents can be interpreted by most web browsers and other HTML software, and can also be validated.

Non-SGML marked-up textual data is more difficult to deal with. It is necessary to evaluate the appropriateness of the encoding, the extent to which it has been documented and the consistency with which it has been applied. It should always be possible to formally differentiate markup from the text of the document, as this is likely to be essential for any users examining the text. Bearing these factors in mind its viability as a preservation format must be assessed. The variety of practice in the numerous communities served by the AHDS make it impossible to describe and assess all formats here.



## Common Formats

Format	File Extension	Notes
XML	.xml	An XML file, validated with DTD or schema specified, is a format suitable for preservation.
SGML	.sgml .sgm	A SGML file, validated, with DTD specified, is suitable for preservation.
HTML	.htm, .html	Hypertext markup language file, which may in principle be validated against a DTD. In practice invalid documents are often produced and used.
XHTML	.xhtml, .htm, .html	XML-conformant HTML file, is required to be well-formed and valid.
DTD	.dtd	Document Type Definition. Defines the rules and syntax applied to a document. To be supplied with an SGML or XML document.
XML Schema	.xsd	An XML schema file. Defines the rules and syntax applied to a document. To be supplied with an XML document.
Pseudo-SGML	.sgm, .sgml, .txt or other	A text file employing some SGML-like formalisms for inserting markup, but not valid SGML. Suitability depends on whether tagging is consistently applied and well-documented, sufficient for later migration.
Various non-SGML encodings in text files	.txt or other	Suitability depends on acceptance as de facto standard in an academic community, plus an assessment of its likely future viability and level of documentation

## Additional Information

Basic information on valid and well-formed XML files

< [http://www.w3schools.com/xml/xml\\_dtd.asp](http://www.w3schools.com/xml/xml_dtd.asp) > Last checked 01/01/2004

A Gentle Introduction to XML, from the TEI Guidelines

< <http://www.tei-c.org.uk/P4X/SG.html> > Last checked 01/01/2004



# Ingest Checklist

## Level 1 (Essential)

For 'SGML family' files (XML, XHTML and HTML, as well as native SGML):

- Ensure that all relevant files are received, including headers, extension files and multiple files encoding a single document or corpus.
- Deposit must contain a copy of the DTD or schema associated with the files. The relevant DTD or schema will be referenced at the beginning of the file, or header.
- All encoded documents should be validated against this DTD or schema. XML files may be validated within several XML editors, such as oXygen XML editor or TEI-Emacs. Various XML editors and some validating programs are available. The validation of multiple encoded pages can be achieved using batch files.
- All changes to the data, such as editing the text, or adding annotations, to be recorded in the header.

For non-XML files:

- Assess the appropriateness of the format and refer back to depositor in the event of problems.
- Ensure that markup can be differentiated from the text by some formal procedure.
- In the event of acceptable non-XML textual data, ensure that markup is well-documented and then apply ingest guidelines for plain text or binary data, as appropriate.

## Level 2 (Preferred)

- Ensure that descriptive and non-procedural markup is rigorously and consistently applied. The markup formalism, procedure and guidelines should be well-documented.

## Level 3 (Best practice)

- Where possible and relevant, ensure that accepted guidelines or standards (such as TEI) are followed in applying and documenting the markup.

## Inform Depositor

- If the resource is marked-up in a non-standard or legacy format, as this is likely to significantly affect the viability of long-term preservation.
- If the resource arrives without a DTD or Schema being declared.
- If the depositor has modified or created their own DTD/Schema without including it in the deposit package.
- If files are not valid and require small-scale conversion by AHDS.
- If files are not valid and require substantial correction, or if correction is problematic for any other reason. The resource should be returned to the depositor for correction.



# Preservation

## Significant Characteristics

For the purposes of the preservation of the intellectual content of textual data, it is preferable for markup to be descriptive rather than procedural. This means that markup tags should aim to describe the logical structure or function of a textual element, not prescribe computational procedures to be carried out on the element, or to describe the preferred presentation or function of an element in a particular application.

For textual data to be in a form suitable for long-term preservation, it is not sufficient for it to be presented in formally valid and well-documented SGML. It is quite possible to apply completely meaningless or inconsistent mark-up which is formally valid. It is additionally necessary that the text be encoded with descriptive rather than procedural markup, and for consistent and appropriate guidelines to have been applied. This means that it is necessary to look at what is marked up and how, and not just to run an automatic validation procedure.

SGML family files have a hierarchical structure. SGML documents are also software and device independent, so no proprietary software will be required. SGML documents may be stored in more than one file.

## Technique

A parser should be used to validate the SGML family documents against the specified DTD or Schema. Validation is complete when the parser can find no more errors and returns a message such as "validation complete".

XML and XHTML files may be validated by loading them into one of several XML editors. Some validating programs are available, which may be more convenient for validation of multiple files and of native SGML documents.

## Problems and Issues

There are cases where data is deposited in a non-SGML format which is a *de facto* standard for a particular community of researchers. An example of this is the CHAT format for the network of language development researchers making use of the CHILDES database. In this case, the depositor should be warned of the dangers for the sustainability of the data, and encouraged to export the data to a format more appropriate more long-term preservation. AHDS Literature, Languages and Linguistics is exploring the possibilities for exporting data from this format to XML.

In cases where a legacy format, such as COCOA has been used, the depositor should be informed of this problem, and encouraged to convert the data. AHDS Literature, Languages and Linguistics has developed tools and methods for the conversion of some commonly used formats. However it should be noted that there are few if any guidelines for the implementation of many legacy encodings, and automatic conversion is unlikely to be straightforward.

In cases where a completely non-standard technique has been employed, the depositor should be encouraged to convert the data. No guarantees can be made for the preservation of the resource, but a preservation copy of the data as received should be made.

There are frequent problems with overlapping markup elements. Documents marked up using the SGML family of markup languages have (usually) to implement a single hierarchical model of the structure of the document. This is problematic where the data model is not a single hierarchical tree. For example, a document may be divided in structural terms into front matter, the body of the text, and back matter. And within these sections there may be chapters, page numbers, footnotes, paragraphs, etc. The document could include interpretive annotations which apply to stretches of text, such as tags indicating the start and end of passages of direct speech. These elements are likely to overlap with paragraphs, and so cannot be inserted in the text in the most convenient way. Even the structure of a document may be difficult to capture in a single hierarchy. When representing a print edition of a text in electronic form, pages and other elements of the physical structure of a book are likely to overlap with elements of the logical structure such as paragraphs.



For the reasons given above, data is often prepared using pseudo-SGML, or 'broken' SGML (or XML or HTML, etc.). If the markup formalism, procedure and guidelines are well-documented, then such resources should be considered acceptable for preservation. They should be considered as candidates for migration when the technologies for implementing the overlapping elements are more mature. Where possible, a well-formed version of the text without the problematic markup is also likely to be valuable in addition to the marked-up text.

There are technologies which implement multiple hierarchies and other methods of encoding overlapping markup elements, but these are not widely used, well-developed or standardised and are not implemented in widely available software. Where data incorporating such technologies is offered for deposit, then it needs to be treated on a case by case basis, and efforts should be made to accept it for preservation. Until standards become more developed in this area, it is not possible to make recommendations.

Note that it is now possible to export to XML from a variety of applications, including Microsoft Word. However, simply automatically converting the file format to XML does not make textual data viable for long-term preservation. It is necessary to assess the appropriateness of the resulting ways in which various textual phenomena have been encoded, and it is unlikely that an automatic conversion will result in adequate descriptive markup. Furthermore, the XML produced by Microsoft applications is not even necessarily formally conformant to W3C standards.

## Additional Information

TEI Software page

< <http://www.tei-c.org.uk/Software/index.html> > Last checked 07/03/2006

List of projects who are using or have used the TEI Guidelines

< <http://www.tei-c.org.uk/Applications/index.html> > Last checked 07/03/2006

Oxygen XML

< <http://www.oxygenxml.com/> > Last checked 07/03/2006