EAD : Enabling Armchair Delivery

Approaches to Encoding Finding Aids at the University of Liverpool

Julie Allinson

Special Collections and Archives, University of Liverpool

1 Jun 1999

Abstract

EAD is increasingly being selected as the primary data format for constructing archival finding aids in the British Archive Community as the new technologies and know-how required to encode lists are being embraced in many repositories. One major problem facing archivists, though, is how to convert finding aids held in a variety of formats (including databases, word processed documents and paper lists with no machine readable form) into EAD. This article will discuss the methods used in Special Collections and Archives at the University of Liverpool Library in converting finding aids into EAD. Two main examples will be discussed: firstly, designing database output styles which automatically generate EAD tags to wrap around database fields using the ProCite bibliographic database and secondly, offshore keying of paper lists with the addition of basic EAD tags following a rigorous template designed by Special Collections and Archives staff. Both methods have proved effective and have facilitated the generation of EAD encoded lists for a number of our largest collections. Finally, there will be a brief discussion of our use of native EAD generation using AdeptEdit software and our continuing use of conversion methods.

Special Collections and Archives at the University of Liverpool

The Special Collections and Archives (SCA) division of the University Library was formed in 1996 when Special Collections and the University Archives merged. The purpose of this convergence was to draw together the University's heritage collections of historical manuscripts, archives and printed materials. The merger has not only created a more streamlined service which avoids the confusion the original two-party structure often caused to users but has also provided the perfect opportunity to build a more developed national and international profile for SCA. This has been aided by the finding aids and access tools created in the last 4 years and the pioneering implementation of EAD.

Prior to this merger several listing projects had commenced both in Special Collections and also in the University Archives funded by HEFCE, the Higher Education Funding Council for England, under its Non-Formula Funding Initiative. These projects straddled the merger and as a result different practices were employed. In Special Collections the Gypsy, Rathbone, Glasier and Science Fiction projects employed a bibliographic database called ProCite for the creation of item and piece level records. The projects based in the University Archives utilised word processing, in the case of archives relating to Social Welfare, and for the Cunard Archive and related deposits OCR scanning and word processing.

The HEFCE initiative allowed the employment of a total of 17 people over a period of 4 years. The work undertaken improved access to some of our most important collections, heightening the profile of these collections, and of SCA as a whole, through web presence, word of mouth and general promotion. It is this funding which allowed SCA to embrace EAD through staff availability, expertise and the accessibility of electronic lists for conversion.

As with any rationalisation of resources, the merger brought together differing listing practices and methodologies which are gradually being assimilated by the implementation of EAD. EAD has encouraged the development of a basic finding aid structure, a structure which still adheres to general MAD principles (Manual of Archival Description) yet at the same time moves listing away from MAD and its insistence on a rigid tabular display. This process has drawn SCA towards a unity of listing practice and electronic list generation which helps us provide a better service for more people in more places.

ProCite Pre-EAD

ProCite is a proprietary bibliographic database package selected for use on several of the HEFCE-funded projects which commenced in 1995. It was used to create item level listings for the Gypsy, Glasier, Rathbone, Cunard photographic and Science Fiction collections. The motivating forces behind the decision to utilise ProCite were cost and the need for expediency in the limited funding and short-term environment of the HEFCE projects. Cost effective, easy to install and straightforward to use with staff training time kept to a minimum and basic data input possible with around 5 function keys meant ProCite was a perfect way of kick starting the HEFCE initiative. Already familiar to members of the HEFCE team made training in use of the database and customising workforms and output much easier.

In addition to these benefits of economy, ProCite offered an impressive array of functions which were all readily exploited. It is highly a organised tool and easy to navigate around with menus and searches. It provides "off-the-shelf" workforms for a range of materials yet is easily customised to reflect in-house practice. Authority control features are exemplary, automatically generating author, title and keyword lists in indexed fields (7 of the total 45 fields in each record). These lists are available via a single keystroke at data input and it is also possible to create in-house term lists. The authority control lists can be used as a quick search tool on the initial database screen (see fig 1). Complex searching is accessible from a separate search screen which enables the use of Boolean logic and search expressions. Searches can be performed across all database fields or limited to one field and can specify whether fields begin or end with text, contain text strings or even have any or specified text absent. Editing features allow global field and field content manipulation. Printing and output options allow complete databases or marked lists of "hits" to be printed either to file or printer using any of a wide variety of given styles. The bibliographic slant meant that ProCite was already geared to the generation of MARC-AMC records and could, as will be shown with EAD generation, be customised to reflect personal output specifications.[1]

The main drawback of ProCite use for creating archival lists was its flat structure. The design of a bibliographic database is very much library orientated, geared toward basic lists of items, albeit items of any material. This design does not generally reflect the hierarchical, multilevel nature of the archival finding aid as it treats each record as a discreet item, duplicating any series and group level information in each item, rather than reflecting its relationship to the whole. In gearing the databases to the output of EAD, something of a structure was imposed onto the lists, a structure, though, which indicated hierarchies yet added little to the functionality of the databases themselves intended more as a means to an EAD end.

---

[1] Watry, Paul B and Maureen M Watry. 'Automating Archival Collections Using MARC-AMC and Z39.50 at the University of Liverpool : a case study'. *Journal of the Society of Archivists,* Vol 17, no. 2, 1996, pp. 167-173

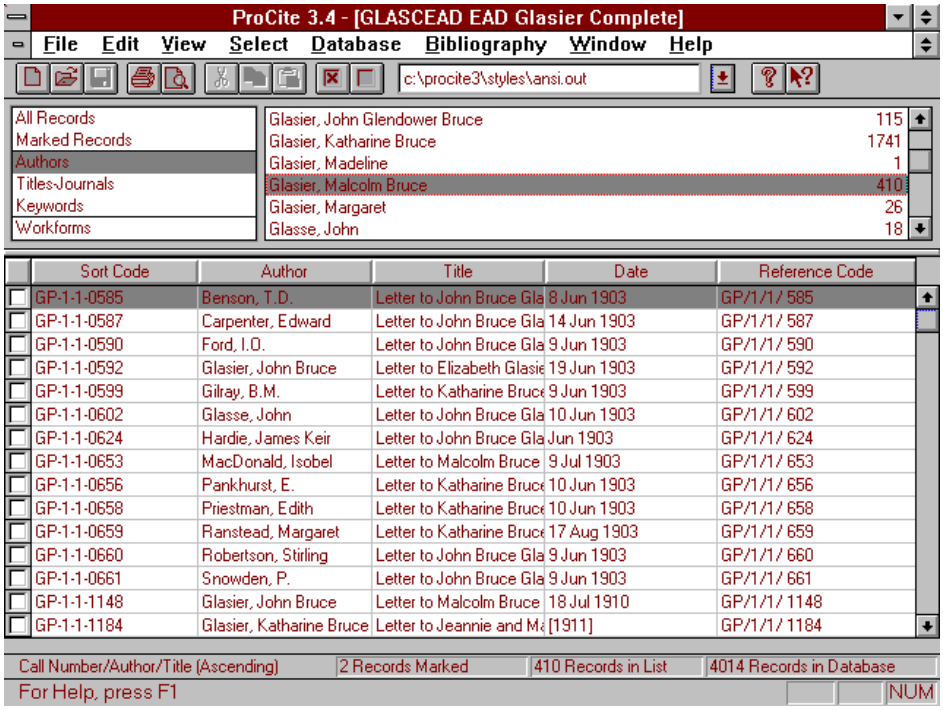


Fig 1 : The ProCite Database showing authority controlled author index

ProCite to EAD

The listing of archives with ProCite at SCA began at a time when EAD was a mere twinkle in Daniel Pitti's eye, and as such was begun without the knowledge that one day it would be necessary, or a least advantageous, to use these databases for the generation of EAD encoded finding aids. With the availability of the beta version EAD DTD, SCA became increasingly aware of the potential importance of EAD and as such began looking at ways to encode existing archival lists. Since the ProCite databases were the most recent additions to our finding aids canon, and with the HEFCE slant towards providing remote access, they were an ideal candidate for conversion to EAD. The customisation of output styles enabled by ProCite could clearly be geared towards EAD generation, and it was a then member of the HEFCE project staff who applied himself to devising the conversion process. Pete Johnston, who is now at the Glasgow Archives and Business Records Centre, was behind the ProCite to EAD conversion and much credit must go to him for the content of this section. Indeed, to a large extent my understanding of EAD's nuts and bolts comes entirely from his invaluable work.

The initial stage of converting these database finding aids was to gain an understanding of the hierarchical relationships within each archive. The example used for the remainder of this section is that of the Glasier papers although the same processes were carried out with reference to the other lists held in ProCite. At the beginning of the HEFCE listing projects the archive structures were identified by means of a simple pencil and paper drawing of a rough "family tree" diagram. This method, although childish in its simplicity, has proved an excellent means of understanding complex archive structures. For the Glasier papers this specified the following 3 levels of information:

Collection	**Series**	**Sub-series**
Glasier Papers	- General correspondence	
		- Public letters
		- Family letters
		- Related material
	- Diaries	
		- John Bruce Glasier
		- Katherine Bruce Glasier

- Notes
    - JBG notebooks
    - JBG loose notes
    - KBG notebooks
    - KBG loose notes
    - Notes var. authors
- Newscuttings
    - loose
    - newspapers
    - scrapbooks
- Printed Ephemera
    - Reports and minutes
    - Printed Propaganda
    - Meetings propaganda
    - Publications propaganda
    - General propaganda,
- Collected Publications
    - Books
    - Pamphlets
- Personal papers
    - Official documents
    - photos and pictures
    - Misc
- Biographical papers
    - Correspondence
    - Newspapers and cuttings
    - Notes
    - Exhibitions and Memorials

Fig 2 : Glasier Papers archival structure

These series--sub-series--item relationships were expressed consistently throughout the ProCite databases within indexed fields. This information would later be used for the creation of Component Level records and would provide an invaluable means of sorting database records. The keywords field (field 45) was used to express the hierarchy drilled down from collection level e.g. Glasier Papers -- General Correspondence -- Public Letters. Fig 3 is an example of an item level record for correspondence and demonstrates how the fields were used. From the outset ProCite workforms had been used consistently over all collections with workforms having been customised for the major classes of material encountered: artwork, manuscripts, correspondence, sound recordings and books. Thus, the databases field content did follow a fairly consistent pattern, a pattern which was identified and closely mapped to EAD later.

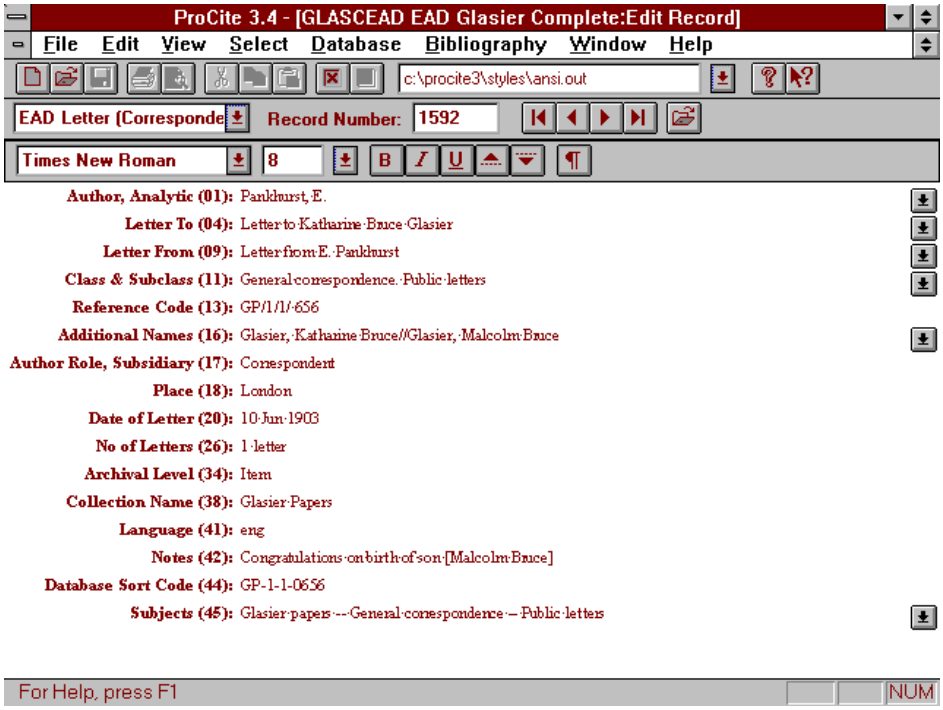


Fig 3 : Sample item level record from ProCite

EAD is structured into archival level specific components. These components nest within each other and require that information be expressed only at its highest level. The nesting means that item level components exist within and closely related to their parent components and thus do not require the repetition of higher level data lower down the hierarchy. Therefore, Series level scope notes, for example, relating to "General Correspondence" which had in ProCite been kept consistently within the Notes field (field 25) of every record of that series would, in EAD, be held in a higher level component area clearly relating to the items following in the list.

To express hierarchies within ProCite, workforms were designed which would wrap group/sub-group level records around each collection of item level records at each particular group or sub-group. (It is perhaps worth noting that the terms class/series/group and their subordinates have been used interchangeably throughout the article to indicate nested levels of archival information). These "component-level" workforms would house information identified as specific to that level. The 2 workforms were entitled "Component Start" and "Component Close" and were designed in close consultation with the EAD tag set. Component Start contained textual information which would be output to EAD whilst component close would be used simply to impose the component close tag (</c>). The following table (fig 4) shows the Component Start workform field names and their intended EAD mappings. In a similar way wraparound FindAid Start and a FindAid Close workforms were created to contain eadheader and initial archdesc elements.

Workform Field	**EAD**
Class & Subclass (#11)	C, DID, HEAD C, DID, UNITTITLE
Reference Code (#13)	C, DID, HEAD C, DID, UNITID
Related Finding Aid (#21)	C, ODD
Note (#25)	C, SCOPECONTENT
Related Materials (#29)	C, ODD
Archival Level (#34)	C [otherlevel]
Arrangement (#35)	C, ARRANGEMENT
Terms Governing Use (#37)	C, ADMININFO, ACCESSRESTRICT

Collection Name (#38)	C, DID, UNITTITLE
Biographical Note (#39)	C, BIOGHIST
Provenance Note (#40)	C, ADMININFO, ACQINFO
Language (#41)	C [langmaterial]
Database Sort Code (#44)	C, DID, UNITID [id]

Fig 4 : Component Start Workform fields and their mapping to EAD

Before an output to EAD could be achieved two more stages had to be completed. Firstly, the database needed ordering to reflect the finding aid structure of fig 2 and secondly an EAD output style had to be designed.

The seemingly simple process of putting the database into order has provided one of the biggest challenges to EAD generation. The use of what Pete Johnston refers to as "heterogeneous numbering conventions"[2] which could not be sorted by ProCite has made it necessary to manually insert a sort code (field 44) alongside the reference code (field 13). The sort code provides a unique number to reflect the collection, group, subgroups and item specific number whilst conforming to the restrictions placed on SGML literals of this kind. In this way a Glasier reference code of GP 1/3/2/5 would have a sort code of say gp-1-03-2-005 and GP 1/12/4/189 would be expressed as gp-1-12-4-189. Each element of the code (separated by a dash) must contain a standard number of digits. This necessitates the addition of extra zeros to correspond to the highest number in that area of the code. For example, if group 1 contains subgroups 1 to 12 (GP 1/1 – GP 1/12) this would have sort codes of gp-1-01 to gp-1-12 (an extra 0 is added because the total number of subgroups exceeds 9) yet group 2 with, say, only 4 subgroups (GP 2/1 – GP 2/4) does not need extra zeros (gp-2-1 to gp-2-4). This applies particularly for item level records where numbers can go into hundreds or even thousands so if subgroup 9 has 100+ items then the sort code would begin at gp-1-09-001 and if it contained 1000+ records it would begin gp-1-09-0001 and so on. In ProCite this code provides nothing more than a sorting service and is, thus, kept invisible to users in any database printout. In EAD it acts as a UNITID id attribute useful for index generation and navigation using <PTR> or <REF> elements. Around 40 000 database records had sort codes manually added, a Herculean task which for future ProCite to EAD use can be avoided by adding this sort code from the database inception.

The final stage towards EAD generation was the design of an output style which would map EAD tags to database fields. Fig 5 shows the output generated for an item of correspondence. Similar styles were designed for artwork, manuscripts, sound recording and books utilising a template that followed a general tagging pattern which, with the addition of FindAid and higher Component level records, looked basically like this :

```
<C LEVEL="OTHERLEVEL" OTHERLEVEL="Item"
LANGMATERIAL="eng">

<DID>

<UNITID ID="GP-1-1-0622">GP/1/1/ 622</UNITID>

<ORIGINATION><NAME SOURCE="LOCAL">Hardie,
James Keir</NAME></ORIGINATION>

<UNITTITLE>Letter to John Bruce Glasier.
<GEOGNAME>London</GEOGNAME>;
<UNITDATE>18 May
1903</UNITDATE>.</UNITTITLE>

<PHYSDESC><EXTENT>1
letter</EXTENT></PHYSDESC>

<NOTE><P>Notes on [Preston candidature] contest:
Hodge's position and Miss Pankhurst's involvement;
agreement to transfer the Labour Leader to the Independent
Labour Party</P></NOTE>

</DID>

<CONTROLACCESS><NAME
SOURCE="LOCAL">Hardie, James
Keir</NAME><NAME SOURCE="LOCAL">Glasier, John
Bruce</NAME><NAME SOURCE="LOCAL">Hodge,
John</NAME><NAME
SOURCE="LOCAL">Pankhurst</NAME><SUBJECT
SOURCE="LOCAL">Glasier papers -- General
correspondence -- Public
letters</SUBJECT></CONTROLACCESS>

</C>
```

Fig 5 : EAD output generated by ProCite

[2] Johnston, Peter. *Configuring a ProCite database for "legacy" archival descriptions - encoded document,* [unpublished, c. 1997]

<EADHEADER>
<FINAID><ARCHDESC>[from findaid start record]

<C LEVEL="SERIES">[from component level record]

<C LEVEL="SUBSERIES">[from component start record]

<C LEVEL= LANGMATERIAL=><DID>
<UNITID ID=>
<ORIGINATION><NAME SOURCE="LOCAL">
<UNITTITLE><UNITDATE>
<PHYSDESC><EXTENT>
<NOTE><P>
</DID></C>

<CONTROLACCESS>
<NAME>
<SUBJECT>
</CONTROLACCESS>

</C>[from Component close record]

</C>[from Component close record]

</ARCHDESC></FINDAID>
[from findaid close record]

The entire ProCite database was then printed to a file called body.sgm within a Glasier directory. A separate document.sgm file was created containing the necessary DTD and entity declarations. The catalog file points the document to the relevant body and index files.

<!DOCTYPE EAD PUBLIC "-//Society of American Archivists//DTD ead.dtd (Encoded Archival Description (EAD))//EN" [
<!ENTITY body PUBLIC "-//University of Liverpool//TEXT Document Body//EN">
<!ENTITY Index PUBLIC "-//University of Liverpool//TEXT Document Index//EN">
<!ENTITY collectn STARTTAG "Glasier Papers">]>

<EAD>
&body;
</EAD>

Using this conversion process it was possible to create a valid EAD encoded document. With another output style it was equally possible to produce a names index within <ADD> using the database sort code as pointers.

Eg:

<ADD><INDEX><HEAD>Names Index</HEAD>
<INDEXENTRY><NAME>Hardie, Keir</NAME>
<REF TARGET="gp-1-3-10-001">GP 1/3/10/1</REF>
<REF TARGET="gp-2-4-034">GP 2/4/34</REF></INDEXENTRY>
</INDEX></ADD>

To view the EAD document the SGML viewing software Panorama was chosen initially due to its relatively low cost. A fairly complex stylesheet was designed which would be able to handle all the database generated EAD lists. Fig 6 show's an extract from a panorama display of the Glasier list with the collapsible navigator on the left hand side and complete finding aid on the right. Fig 7 shows an examples from the generated names index.

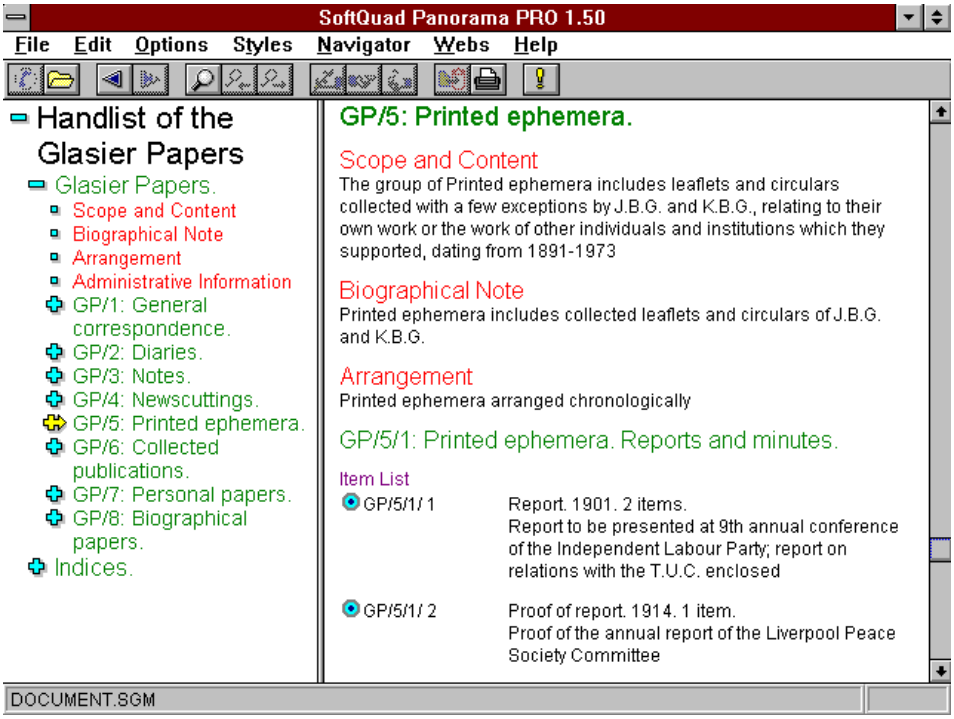


Fig 6 : database generated EAD document displayed using Panorama viewer

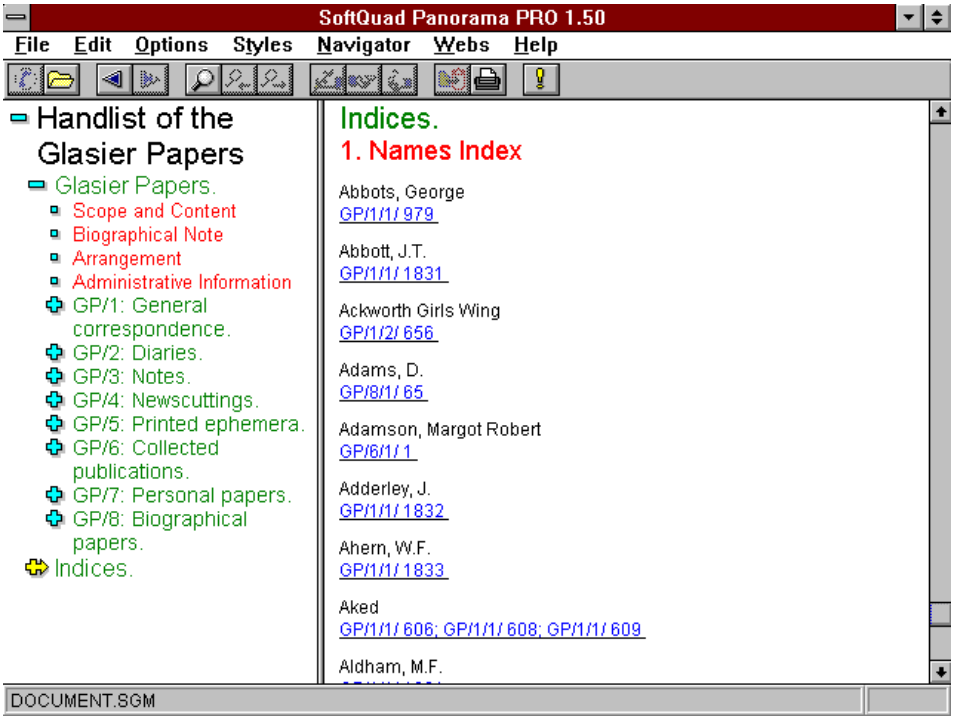


Fig 7 : Panorama display showing names index

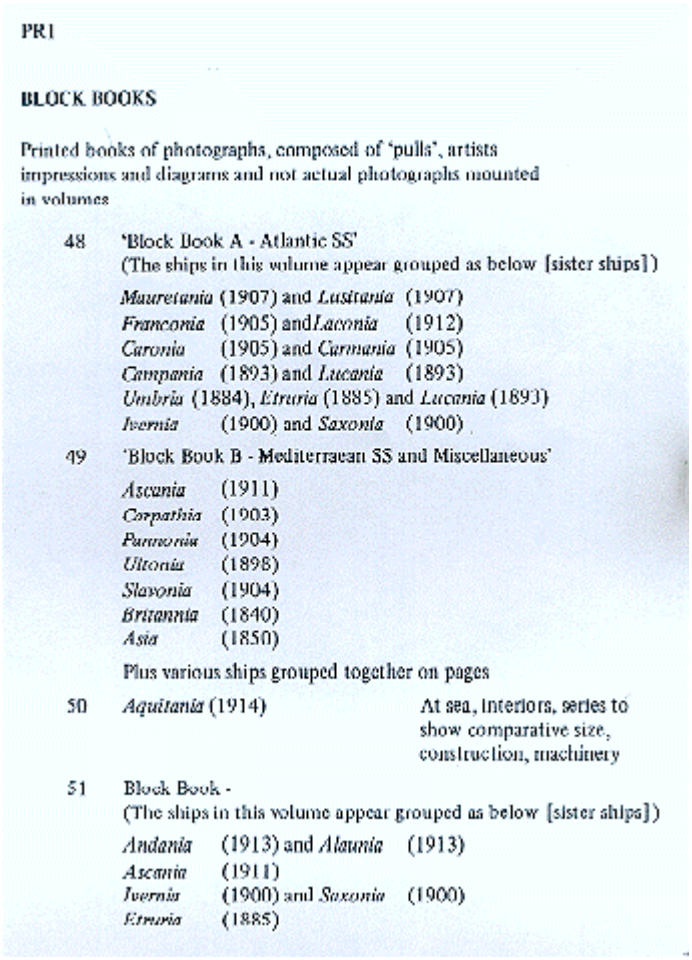
At first glance this conversion process may appear to be complex and to demand much staff input. Initially, yes, this is quite true but many of the initial difficulties came as a result of the EAD/SGML learning curve and the application of this knowledge to a flat database. The main problem of the ProCite to EAD approach was the imposition of a navigable hierarchical structure onto the database with its generically flat structure. By using the full range of functions within ProCite (workform and output style design, authority control, global editing) it was possible to make archival structure more explicit. The procedure used here in SCA with ProCite can also be used with other database software packages. The benefits of using databases for EAD output include the standardisation of tagging which

automatic generation ensures and the infinitely various possibilities of output style designs to apply different structures onto the same information. On the negative side databases are inflexible, limited to a set number of fields and limit the in-depth tagging and attribute use an SGML editor can accomplish. Of course, any encoded document can be additionally manipulated using an SGML editor. This, though, begs questions over what to regard as the basic data, EAD document or database? If changes are made to the database then the whole list will need re-converting. If changes are made only to the SGML encoded text then the database becomes redundant yet to edit both database and EAD wastes time. These questions, though, are minor in the light of it being possible to create valid SGML without any major re-keying of data. With the methodology in place, conversion is now a relatively simple exercise which, with an awareness of EAD, can be made more rigorous. An example of the continued use of ProCite for EAD will be discussed later in the article.

Offshore Keying with partial EAD tagging

One method has already been shown whereby finding aids were converted into valid EAD documents from electronically held databases. Similar conversions are often possible on word-processed documents using word and style templates or programmed conversion scripts. All of these methods depend upon the existence of a digital copy of the archival list in question. For many archives, though, this is not necessarily the case when older lists being have been created by typewriter or outdated proprietary software which is no longer accessible. SCA is no exception and our Archives collections are peppered with such examples. The most painfully obvious example is that of the Cunard Steamship Company Archives, deposited in the 1970's and worked on since deposit by a variety of archivists and volunteers. Listing work led to the creation of the Catalogue of the Archive, published in 1987. This list brings with it a host of problems both for users and staff yet the importance of the Cunard Archive within SCA cannot be underestimated. It is our largest single collection and, covering the popular research areas of shipping and family history, the most used. This use comes in large proportions from remote sources with enquiries on a daily basis from across the globe and frequently from those with Internet access. The Archives' importance within the division made the Cunard list a perfect candidate for an exercise in retrospective conversion and online delivery.

Fig 8 : Sample Page from Cunard List



PR1

BLOCK BOOKS

Printed books of photographs, composed of 'pulls', artists impressions and diagrams and not actual photographs mounted in volumes

48 'Block Book A - Atlantic SS'
(The ships in this volume appear grouped as below [sister ships])

Mauretania (1907) and *Lusitania* (1907)
Franconia (1905) and *Laconia* (1912)
Caronia (1905) and *Carmania* (1905)
Campania (1893) and *Lucania* (1893)
Umbria (1884), *Etruria* (1885) and *Lucania* (1893)
Ivernia (1900) and *Saxonia* (1900)

49 'Block Book B - Mediterraean SS and Miscellaneous'

Ascania (1911)
Carpathia (1903)
Pannonia (1904)
Ultonia (1898)
Slavonia (1904)
Britannia (1840)
Asia (1850)

Plus various ships grouped together on pages

50 *Aquitania* (1914) — At sea, interiors, series to show comparative size, construction, machinery

51 Block Book -
(The ships in this volume appear grouped as below [sister ships])

Andania (1913) and *Alaunia* (1913)
Ascania (1911)
Ivernia (1900) and *Saxonia* (1900)
Etruria (1885)

The Cunard list itself is over 1000 pages long and is printed in an odd A5 format. It is, thus, difficult to copy (even a photocopy must be manually fed rather than sent through automatically) and maintain. The A5 format does not correspond to the corporate image of other archival finding aids in SCA. The listing itself is incomplete, in many instances additions and corrections need to be made with many areas needing a depth of detail not currently present on the list.

Previous attempts to make the list accessible electronically have included OCR text scanning. Figs 8 and 9 show an example of a page from the list and the results of scanning. Because of the tabular arrangement of the list scanning has delivered unsatisfactory results. Even this relatively straightforward example, unable to recognise the

intellectual structure of the document, has detached dates from text and can never be expected to keep consistently to the original pagination. In some scanned examples, 1 page of list became 5 pages of random words and numbers deemed unworthy of reproducing for the purposes of this article. Therefore, the staff input required for both scanning and complex editing of the document post scanning made this option unworkable.

Se~ ~ e,toy- *09 ~o~r*

PR1

BLOCK BOOKS

Printed books of photographs, composed of 'pulls', artists impressions and diagrams and not actual photographs mounted in volumes

(1850) 'Block Book A - Atlantic SS'

(The ships in this volume appear grouped as below [sister ships])

Mauretania (1907) and *Lusitania* (1907)
Franconia (1905)andLaconia (1912)
Caronia (1905)andCannania (1905)
Campania (1893)andLucania (1893)
Umbria (1884), *Etruria* (1885) and *Lucania (1893)*
Ivemia (1900) and *Saxonia* (1900)

(1850) 'Block Book B - Mediterraean SS and Miscellaneous'

At sea, interiors, series to show comparative size, construction, machinery

51 Block Book-

(The ships in this volume appear grouped as below [sister ships])

(1850) *Aquitania* (1914)
Plus various ships grouped together on pages
(1913)andA!aunia (1913)
(1911)
(1900) and *Saxonia* (1900)
(1885)

A scania (1911)
Carp at hia (1903)
Pannonia (1904)
Ultonia (1898)
Slavonia (1904)
Britannia (1840)
Asia (1850)
Andania
A scania
ivernia
Etruria

Fig 9 : Cunard list after text scanning, notice the divorce of dates from ship names and the loss of document structure

With the failure of scanning, a re-think was needed. To re-key in-house would take an enormous input of staff hours and SCA no longer had sufficient staff members to hand with the HEFCE initiative drawing to a close. Looking at the Public Record Office solution of offshore keying for their 400 000 pages of paper lists gave us the answer. With the payment of an initial start-up fee it was possible to have finding aids keying at a relatively low cost (c. £1 per page) with a high level of accuracy and a rapid turnaround. In addition, the company were willing to add basic tagging at input stage. With offshore keying it was felt we had found the perfect answer to creating an electronic Cunard list.

Because the "rekeyers" lack archival knowledge but were able to add basic tag strings at input, it was necessary to mark the pages of the list with tagging instructions. This involved the design of a mark-up template that would then be applied to the entire list before it was sent for keying. It is this template design procedure and subsequent list mark-up which will be fully discussed in the remainder of this section.

The mark-up template needed to be well planned in order to ensure that it would adhere to national and international archival standards (including those of EAD) as well as compliance with in-house standards and practices. It also needed to be general enough to apply to the variety of listing styles and the tabular layout of the existing list. Finally, it must be easy to understand for those carrying out the encoding and for this it was decided to err on the side of caution and assume that those marking up the list were total EAD novices.

The initial stage of the design process was to identify from existing SCA finding aids, the archival elements which were mandatory and those which were used with some frequency. This enabled the drawing up of a tag list that was very similar to those used in the database conversion process. The second step was to perform a detailed survey of the Cunard list in order to understand it's structure, component parts and to draw up a basic set of archival elements for comparison with those identified in other finding aids.

The original intention of EAD is to reflect the various content of documents without losing the value of the information held therein. Therefore, these survey and identification stages were crucial to ensure encoding is carried out to maximise the potential of the content data. The nature of the Cunard list and its variety of listing styles and content necessitated an EAD 'lite' approach as, to a large extent the content fell into a small number of broad areas which in turn were quite easily mapped to the following EAD tags.

<SCOPECONTENT>, <P>
<ARRANGEMENT>, <LIST>
<UNITID>
<UNITTITLE>
<UNITDATE>
<NOTE>, <P>, <LIST>

Naturally, it is only possible to key in data physically present within the list. Thus, information such as <BIOGHIST>, <ADMININFO>, <ORIGINATION>, <PHYSDESC> etc. which had been common features in ProCite were found with such infrequency in this list that it was felt these elements would be more usefully mapped to a closely related tag. Problems with lack of content can only be solved by a major re-listing project.

The template consisted of a number of component specific mini-templates, which would be reflected by the use of a fluorescent marker pen and invoked on encountering each specific colour. Numbered component level tags were selected (<C01> to <C05>) with the addition of level attributes to enable easier navigability and clearer template design. A decision was taken to use the <C04> component level consistently for items and this, in some instances, demanded the use of a "dummy" <C03> (i.e. <C03> with no <DID> elements, used purely to enclose <C04> item lists) where item levels fell as a third level component. The reasoning behind this was to avoid the need for separate item level and subgroup level <C03> templates. In a small number of instances a fifth level component (<C05>) was used. This, though, demanded little change to the <C04> template and was attributed to the MAD inspired <C05 OTHERLEVEL="piece">, a convention consistent in some of our other finding aids.

The template looked something like this:

Highlight colour = Items highlighted are <unitid> and <unittitle>   **</c01><c01 level=series><did><unitid>D42/ (**reference number)**</unitid><unittitle>**(title)**</unittitle>** **</did>** **<scopecontent> **(blocks of text)** </scopecontent>** **<arrangement><list><item>**(lists)**</item><item>**(each list element separated by an item tags)**</item></list></arrangement>**
Highlight colour: NB Cases where the green highlight is followed by orange highlights Items highlighted are <unitid>, <unittitle> and <unitdate> Otherwise Items highlighted are <unitid> and <unittitle>   **</c04><c04 level=item><did>** **<unitid>D42/** (reference number)**</unitid>** **<unittitle>(title)<unitdate>(date)</unitdate></unittitle>** **<note> ** (additional notes)** **(paragraph tags used to separate each note item** </note></did>**

Fig 10 : Cunard List Tagging templates

Each template was illustrated with examples and with some introductory notes the template was complete.

GENERAL NOTES:

- Each highlight invokes the template for that colour, the template is used for all text following the highlight (including highlighted text itself) until a new highlight occurs.
- When there is a sequence of the same colour highlight following each other, then the template is re-used for each
- **D42/** should <u>always</u> be added to the beginning of the unitid
- Tags are not required when there is no text for inclusion, e.g. if there is no text following a pink highlighted <unittitle> then <scopecontent> or <arrangement> tags are not required (see example)
- Manuscript additions in black form part of the keyed text, additions in red are instructions relating to the tagging, black marker crosses out text to be ignored
- The list is arranged in a table form and with text reading in chunks left to right, sometimes running onto several lines.

Fig 11 : Cunard Tagging Template Introductory Notes

One issue which required clarification was how to deal with repetitions. Throughout the list reference codes were split and implied through the page layout rather than given fully for each item. An example can be seen in Fig 9 where the reference code PR1 is given only at the top of the page and the intellectual relationship to the Block Book Numbers 48 and 49 is not specific. To request item 48, the full reference would be PR1/48, yet this can only be ascertained by associating information in different areas on a page. In the marked up list each reference code was keyed in fully and had the deposit number D42, necessary for retrieval, given. The aim of this was to reflect the descriptive mark-up of element content, upon which EAD is based, rather than the procedural method that avoids repeating information, rather gearing to the visual page layout. In this way, information relating to each element is kept at its specific component level so that items can be taken out of context yet still be fully understood.

The second form of repetition encountered was that of text repetitions such as Original file numbers which were given throughout the list as an additional column of data (see fig 12) with the column heading "Original file number". One potential approach to this data was the use of EAD tabular display elements but these were quickly abandoned after reading of the difficulties of their use (e.g. Richard Higgins of Durham University[3]). It was decided that, given the confusion often caused users by this additional reference number in the list, they would be separated from the <UNITID> element into the <NOTE> area. The column header must, thus, be repeated alongside each occurrence (<NOTE><P>Original ref. no.: 56</P><NOTE>). Fig 13 shows one such example. Here, the highlights indicate the use of the <C04 LEVEL="item"> template and the boxes and lines indicate tagging and the repetition of "Original file no." and "C1/". Letters represent tagging instructions and were explained by means of a key provided within the tag template.

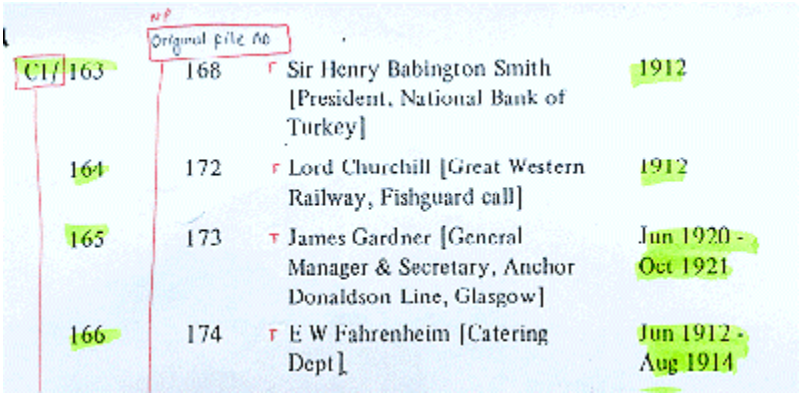


Fig 12 : Example from marked up Cunard list, note repetition of Original file numbers

Fig 13 shows a similar example where duplicate <UNITID>'s were necessary to split unwieldy items into smaller components.

[3] Higgins, Richard. 'Standardised Languages for Data Exchange and Storage : the Encoded Archival Description : using SGML to create permanent electronic handlists', *Business Archives Principles and Practice,* 73, May 1997

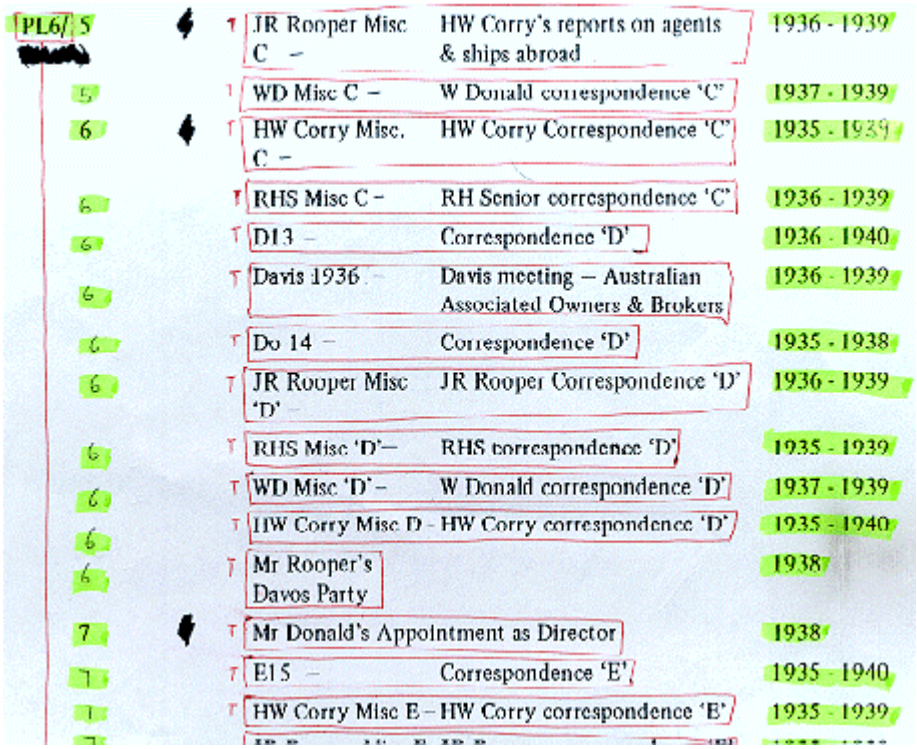


Fig 13 : marked up example fromCunard list, note repeated reference numbering

The actual marking up of the list demanded concentration but was quite swift and completed in a matter of weeks. It took only around a month for the keying process and on the return of the list it was a great relief to discover that the template had been understood completely and used consistently and accurately to great success. With the simple addition of <EAD>, <EADHEADER>, and initial <ARCHDESC> elements this large Finding Aid parsed with no errors. Work on this list is by no means complete. Additions, corrections and annotations to the original list need to be carried out. The encoding itself is very basic in structure and it would be beneficial to improve the depth of tagging and attribute use as well as adding <CONTROLACCESS> elements to aid navigation.

Retrospective Conversion, then, has by no means provided a definitive Cunard Finding Aid but it has created a significant milestone on the road to remote access and given a concrete work in progress to take to potential funding sources. Above all, a validated EAD document is now available for delivery.

One thought on this retro-converson project. I have found it stated in many case studies that although EAD provides an infinite flexibility, its implementation forces Finding Aid structure and listing practice to undergo comprehensive reassessment[4] . The conversion of the Cunard Catalogue is no exception. It was only through trying to map EAD onto the list that I became aware of the idiosyncrasies and inaccuracies therein. It is these which I have attempted to iron out with the creation of a standardised EAD template which takes a less style/table procedure-orientated approach to listing and imposes an ordered, content-orientated descriptive structure. Applying EAD to existing archival helps to develop a more critical eye when appraising archival lists for structure and standardised practice.

Direct EAD Input

This final section of the article will discuss the continuing use of EAD in SCA and the purchase of an SGML editor to generate native SGML. As has already been shown, it is possible to create EAD tagging without SGML specific software but it is worth noting that any of the documents generated by other means can be manipulated, corrected and added to with SGML editors.

[4] Dooley, Jackie M (ed.) *EAD Encoded Archival Description : Context, Theory and Case Studies.* The Society of Archivists, Chicago, 1998

The software chosen by SCA was ArborText's Adept Editor and Document Architect. Adept was expensive but offered the ability to handle very large files. This was a major consideration as some of the lists generated from ProCite were very large and other collections due for listing were also of some considerable size. Other good features included the user friendly input screen, split like Panorama for easier navigability with collapsible elements. Input controls were useful, allowing only valid tags to be inserted from any one point with a completeness check pointing out missing and badly used attributes. The creation of tag templates was very handy to ensure tagging was used consistently, in order and not simply overlooked. The creation of file entities meant that sections of the list could be separated into discreet chunks of EAD whilst remaining accessible to the main list. This enables one list to be worked on consecutively by different members of staff.

Behind the relatively easy to manipulate front end of the Adept Editor package are some quite complex and rigorous tools. The Adept Command Language and its ACL Designer add-on give a programming element to Adept and allow for the design of a customised interface, forms driven data entry, dialog boxes and automatic text generation. Using an ACL script and a friendly programmer (increasingly necessary in the electronic age) it has been possible to create Authority files for <GEOGNAME>, <PERSNAME>, <CORPNAME> and <SUBJECT> which pop-up on tag selection. This enables the maintenance of lists which reduce input time for frequently occurring names and create a standardised thesaurus which can later be indexed.

Direct EAD encoding has been carried out in SCA since 1997 with the deposit, in 1997, of the Papers of David Owen, Chancellor of the University. This project has become our flagship project for the use of EAD although concurrent to this 2 year project various other lists have also been encoded. These include the Josephine Butler Collection, Dora Yates Papers, Poverty 3 Community Project Papers and parts of the Dr Barnardo's list.

Direct SGML generation in Adept Editor gives access to the full range of EAD tags and attributes and, thus, allows for the creation of more intelligent, structured and deeply tagged documents. On the downside, an SGML editor restricts the user only as per EAD tagging rules, rules which by their nature enable enormous flexibility. With different members of staff working on EAD listing projects this can lead to the creation of inconsistent finding aids which are very different in structure and content. To avoid this each member of staff embarking on an EAD listing project is recommended to follow a number of guidelines. Initially they are encouraged to draw a simple tree diagram to reflect the structure of their archive and to mark this up with component level EAD elements (<C01 LEVEL=> etc.). This gives a visual aid which helps keep track of the structure of their document, something which can become quite confusing on screen. Secondly, it is suggested that other EAD encoded finding aids are surveyed within Adept Editor as an aid to understanding EAD and at the same time gaining an awareness of the list of tags used consistently throughout SCA. This process is aided by SCA's migration to a unix-based-platfom which allows SCA users read-only access any file from any

GEOGNAME

tagged as they appear in the text. Expanded versions given in controlled access

ATTRIBUTES

SOURCE
- othersource="ICSSD"

alternative sources are atlases, gazetteers etc. and should be specified in othersource as used

ROLES
- subject
- place of creation - place of writing, publication etc.
- venue - used for conferences etc.

Example:
<GEOGNAME>Plymouth
<CONTROLACCESS><GEOGNAME
othersource="gazetteer" role="venue">England : Devon : Plymouth
expanded form should use country and town name for major cities with counties/areas specified for smaller places

Fig 14 : <GEOGNAME> Tagging convention, David Owen project, Sep 1988

computer. Finally, a list of tagging conventions were created, primarily for the David Owen project but applicable to any SCA EAD finding aid, which specified tag and attribute use for the most common tags. Fig 14 shows a sample entry from these guidelines for the <GEOGNAME> tag.

The one major drawback of an SGML editor is its complexity (or perceived complexity) for non-technical staff . Most people are comfortable using word processing software and if Adept is treated as a simple word processing package with added tags then they find it quite easy to use. This, though, ignores much of the functionality of Adept and does not justify its cost. To harness some of the more complex function involves a steep learning curve and demands technical awareness in approach, an approach many of us are unwilling or unable to take. For example, to change the display of Adept one needs to made alternations to the Formatting Output Specification Instance (FOSI), a process which involves a complex network of miniature changes to lists of seeming jargon.

Whilst working on the David Owen Project I had 2 large groups of articles (c. 500) and speeches (c. 1000) to list. These formed part of the Owen Archive (D709 3/18 and D709 3/17 respectively) yet at the same time formed discreet units of flat bibliographic material. On surveying the material I was reminded of ProCite and its bibliographic function. Although the Owen list was in the process of being encoded directly into EAD, I decided to try and combine this native EAD with the organisation of item level data elements in a ProCite database. Coming to ProCite with a knowledge of EAD it was possible generate more complex EAD finding aids. The ProCite workform designed for the Owen articles and speeches was more closely mapped to EAD with field names related specifically to their EAD destination. For example Speech title field became “Unititle”, reference codes were “unitid” and sort-code “attribute id” and the indexed fields were named “Controlaccess (persnames), (geognames) or (subjects)”. This provided a constant reminder to the inputter of the EAD destination of data and, thus, the need for EAD compliance. In addition to tag generation on output, some basic EAD tagging was carried out in the notes via the creation of a ProCite term list, see fig 15.

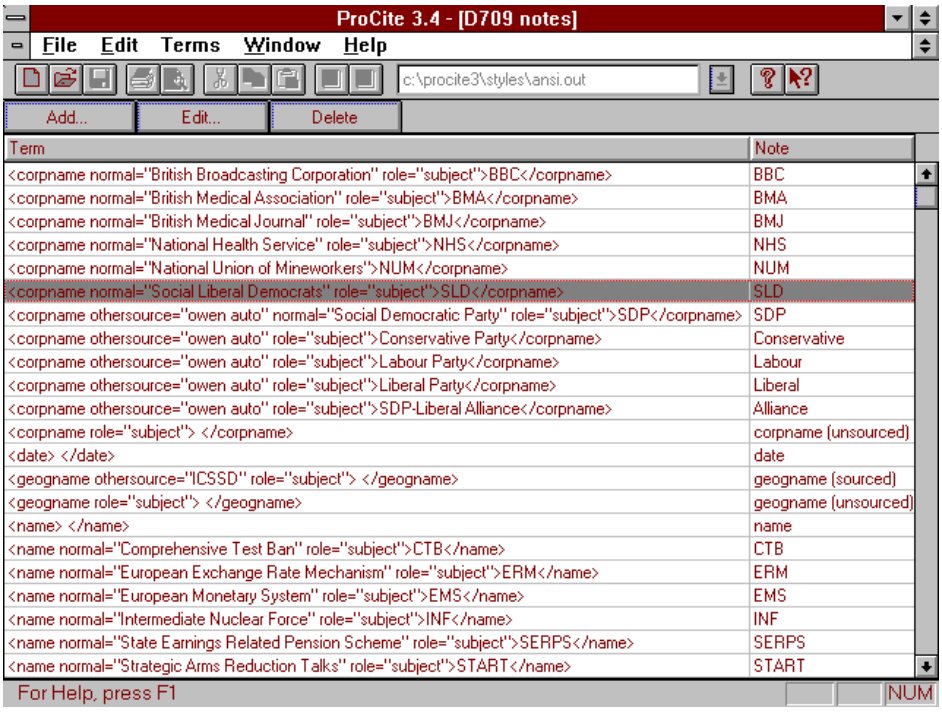


Fig 15 : EAD terms list for ProCite notes field

Fig 16 shows the output generated from this process. Notice the addition of tagging in the notes area and also the addition of -£- and -$$$$- to the <PERSNAME> and <GEOGNAME> elements. This code was devised to indicate attributes which could be applied by a series of global edits on the ASCII output document. For example, <GEOGNAME>-$$$$- would be replaced with <GEOGNAME othersource=“gazetteer” role=“venue”> where dollars signs indicate use of the attribute role=venue and the four dollar signs signifies othersource=gazetteer. These codes, used for role attributes and

othersource attributes only, had the benefit, in ProCite, of not affecting the alphabetical indexing of terms and for EAD were consistent with the Owen list as tagged directly without demanding time spent amending the entire list.

```
<C04 LEVEL="Item" LANGMATERIAL="Eng">

<DID>

<UNITID ID="d709.3.17.1.81.04.13">D709 3/17/1</UNITID>

<ORIGINATION><PERSNAME>-£-Owen, David</PERSNAME></ORIGINATION>

<UNITTITLE>The Wider Responsibilities of the Physician : The Third Lord Cohen Lecture delivered
to the Royal Society of Health Annual Congress, Plymouth<UNITDATE>13 Apr
1981</UNITDATE></UNITTITLE>

<PHYSDESC><GENREFORM>Lecture</GENREFORM>: <EXTENT>ts (13p.) + printed
(4p.)</EXTENT></PHYSDESC>

<NOTE><P>Lecture given 13 Apr 1981 and published in the <title>Royal Society of Health
Journal</title>, Vol 101, no.  3, <date>Jun 1981</date>, pp.  85-88.  Also attached is an issue of the
journal containing the Lord Cohen Lecture from <date>Jun 1979</date> and <corpname
role="subject">Royal Society of Heath</corpname> congress 13-16 Apr 1981
programme.</p><p>On health and the role of the Physician.  Makes particular reference to smoking
and drinking and also to famine in <geogname othersource="ICSSD"
role="subject">Africa</geogname>.</P></NOTE>

</DID>

<CONTROLACCESS>
<?Pub _newline><GEOGNAME>-$$$$-England : Devon : Plymouth</geogname>
<?Pub _newline><SUBJECT OTHERSOURCE="ICSSD">Health</SUBJECT>
<?Pub _newline><SUBJECT OTHERSOURCE="ICSSD">Medical personnel</SUBJECT>
<?Pub _newline><SUBJECT OTHERSOURCE="ICSSD">Tobacco</SUBJECT>
<?Pub _newline><SUBJECT OTHERSOURCE="ICSSD">Alcohol</SUBJECT>
<?Pub _newline><SUBJECT OTHERSOURCE="ICSSD">Famine</SUBJECT>
</CONTROLACCESS>

</C04>
```

Fig 16 : ProCite EAD output for item level David Owen speech

This continued use of database software may appear redundant and complicated but has provided a way of using familiar software (ProCite) to list the material for which it was designed. Familiar to those using it, ProCite enables the swift input of large numbers of item level records consistently and leaves us with a searchable database use both in-house to assist David Owen himself and a fairly thoroughly tagged EAD list. Using ProCite from a networked PC and Adept from Solaris/Linux gives the additional benefit of somewhere to continue working if one or the other system crashes.

Overall this marriage of direct and generated EAD has proved successful in the Owen project and has allowed two members of staff to work on the same document simultaneously. It has also demonstrated that conversion practices and native SGML encoding are not mutually exclusive, indeed it is perfectly feasible to have various elements of the same list encoded by different methods. This could be of use if, say, a word document listing is held for an archival sub-series and needs encoding for addition to its parent EAD list. Future conversion projects will include archival lists held in word processed form using word styles and macros or programmed conversion scripts.

Conclusion

EAD has provided archives with the best method yet of creating digital renderings of complex, multilevel finding aids. For any archive, though, the process of implementing EAD is not as simple as buying a piece of software and starting to encode. With lists held in a variety of proprietary and non electronic formats converting a suite of finding aids is an enormous task in retrospective conversion. Here at Special Collections and Archives we have embraced EAD as fully as possible and are in the process of encoding various finding aids directly into EAD with an SGML editor. Concurrently we have implemented a conversion process to generate EAD files for some of our major collections lists held in database format as well as having one of our major lists sent overseas for re-keying and basic EAD tagging. The article has demonstrated Liverpool University's EAD methodologies and results which facilitate the delivery of encoded archival finding aids in the online arena.

Acknowledgements

- Thanks to Maureen and Paul Watry for support; to all SCA staff and their EAD lists, particularly Emma Challinor, the other half of project Owen and creator of the Glasier database. Also to Pete Johnston.

Bibliography

1. ArborText. *[Adept Software documentation]*, ArborText

2. ArborText. *Native SGML vs Filtered : a white paper from ArborText*, ArborText

3. Bentely Historical Library EAD Finding Aids Project. *Bentley EAD Tag Set*, at URL http://www.umich.edu/~bhl/EAD/bhltags.htm [current 28 May 1999]

4. Cook, Michael. 'The Cunard Archives at Liverpool'. *Business History,* 20, 1978

5. Cook, Michael and Margaret Proctor. *Manual of Archival Description,* Gower, Aldershot, Hants., 1989

6. Coombes, James H, Allen H Renear and Steven J DeRose. *Markup Systems and the Future of Scholarly Text Processing 1*

7. Dooley, Jackie M (ed.). *Encoded Archival Description : context, theory and case studies*, Society of American Archivists, Chicago, [1998]

8. Encoded Archival Description Working Group of the Society of American Archivists. *EAD Encoded Archival Description Tag Library*, Society of American Archives, Chicago, 1998

9. Glasgow University Archives and Business Records Centre. *Encoded Archival Description Project,* at URL <http://www.archives.gla.ac.uk/arcbrc/ead/> [current 28 May 1999]

10. Harvard/Radcliffe Digital Finding Aids Project. *Guidelines for using the EAD v. 1 at Harvard Radcliffe repositories*. Jun 1998, at URL http://findingaids.harvard.edu/dfap/eadgiude.html [current 2 Oct 1998]

11. Higgins, Richard. 'The Standardised Languages for Data Exchange and Storage : The Encoded Archival Description : using SGML to create permanent electronic handlists'. *Business Archives*

Principles and Practice, 73, May 1997, at URL http://www.dur.ac.uk/Library/asc/eadarticle.html [current 28 May 1999]

12. Johnston, Peter. *Configuring a ProCite database for teh generation of an EAD-encoded document*, [1997].

13. Johnston, Peter. *SGML for Archivists : an overview of Encoded Archival Description (EAD) and on approach to implementation*. Presentation to Master of Archives and Records Management Course. 1997 Oct 14.

14. Pitti, Daniel V. *Accress to Digital Representations of Archival Materials : The Berkeley Finding Aid Project*. RLG Digital Image Access Project: Proceedings From an RLG Symposium. 1995; pp. 73-81. At URL http://sunsite.berkeley.edu/FindingAids/EAD/diap.html [current 10 Jul 1997]

15. Public Records Office. *Archives Direct 2001 : Information Pack*, Public Record Office, [c. 1997]

16. Sanderson, Robert. *EAD2HTML,* at URL <http://gondolin.hist.liv.ac.uk/~azaroth/ead2html/> [current at 28 May 1999]

17. Sperbert-McQueen, C M and Lou Burnard. *A Gentle Introduction to SGML : Chapter two of Guidelines for Electronic Text Encoding and Interchange (TEI P3),* at URL http://www-tei.uic.edu/orgs/tei/sgml/teip3sg/index.html [current 28 May 1999]

18. Watry, Paul B and Maureen M Watry. 'Automating Archival Collections Using MARC-AMC and Z39.50 at the University of Liverpool : a case study'. *Journal of the Society of Archivists,* Vol 17, no. 2, 1996, pp. 167-173