



Digital Database
Content, API and XML Specification
v 2.0

7/Jul/2019

Client Services
clientservices@nla.co.uk
0207 332 9380

NLA Digital Database – Specification

Table of Contents

Digital Database.....	1
<i>Content, API and XML Specification</i>	1
NLA Digital Database – Specification	2
Table of Contents	2
List of Figures	4
1. Document purpose	5
2. Overview.....	5
2.1. Production and processing	5
2.2. File delivery and PDF access	5
3. Security model	8
3.1. eClips authentication methods	8
3.3. Username and password sign-on	8
3.4. IP address based authentication	8
3.5. MyArchive authentication	9
3.6. Permanent cookies	9
3.7. Authentication for the XML feeds.....	9
4. Application programming interface (API).....	10
4.1. NLA-API.DLL/SearchDatabase?.....	10
4.2. NLA-API.DLL/GetPage?.....	14
4.3. NLA-API.DLL/CheckObject?	15
4.4. NLA-API.DLL/GetObject?.....	18
4.5. NLA-API.DLL/PCAClientrequest.....	20
4.6. NLA-API.DLL/ManageArchive?	21
4.7. NLA-API.DLL/ListArchive?	21
4.8. NLA-API administration calls	22
5. Branding.....	23
6. Character set support.....	26
7. Style sheets.....	26
8. Content feeds	27
8.1. Article feeds.....	27
9. Quality and accuracy levels	31
9.1. XML accuracy levels	31
9.2. Timely content availability	31
9.3. Completeness of content	32
9.4. Clipping presentation	32
9.5. Article delivery performance	32
10. Coverage	35
11. Rights, security and le.g.al implications	36
11.1. Rights	36
11.2. Corrections and libellous materials.....	36
11.3. Adobe security features.....	36

12.	Pictures, images and photos	38
13.	Clipping rules	40
14.	Clipping presentation	41
14.1.	Thumbnails	41
14.2.	Page sizes and numbers	41
14.3.	Metadata presentation	42
14.4.	Highlighting	43
15.	Acceptable usage policy	44
16.	The NLA Digital Database search interface	45
17.	Appendix one: newspaper Three Letter Acronyms (TLAs)	47
18.	Appendix two: sample XML feed.....	48
19.	Appendix three: revisions from the consultation specification documents	52
19.1.	General changes	52
19.2.	Changes to the API and XML specifications.....	52
19.3.	Changes to the content specification.....	52
20.	Appendix four: revisions from the beta specification.....	54

List of Figures

Figure 1: Overview of the NLA Digital Database environment (XML access)	6
Figure 2: Overview of the NLA Digital Database environment (IDS access).....	7
Figure 3: Sample branding PDF template.....	24
Figure 4: Sample branded article	25
Figure 5: Architecture overview	34
Figure 6: Sample low-resolution photo with over-stamping	38
Figure 7: Image with embedded text	39
Figure 8: Vector graphics with text	39
Figure 9: Thumbnails	41
Figure 10: Handling spreads	42
Figure 11: Sample metadata in a clipping.....	43
Figure 12: NLA Digital Database interface search screen.....	45
Figure 13: NLA Digital Database interface search results	46

1. Document purpose

This document outlines how external users of the NLA Digital Database, such as press cuttings agencies (PCAs) or publishers themselves, interact with the system.

There are 4 components of the NLA Digital Database which are described:

1. The XML feed delivered daily which lists the articles and associated metadata for a newspaper
2. The API which allows PCA/customer applications to interact with the NLA Digital Database' database and retrieve content
3. The Content Definition which indicates what articles will be converted into XML and PDFs and how those clippings will appear
4. The User Interface which enables users to interact with the NLA Digital Database and retrieve content

2. Overview

2.1. Production and processing

The NLA receives content directly from publishers as part of their daily printing procedures. These files, which arrive using several methods, in a wide variety of formats and with significant variation in naming conventions, require converting to a common format. The NLA uses Fingerpost software, which is in use at all the national newspapers, to correctly transform and track these files through the process.

These files are then converted to low-resolution versions and clipped (turned into articles) by an outsourced service provider based in India. This processing team consists of many highly experienced staff that are also responsible for escalation of issues to the publishers, verifying all content and ensuring the quality of the clippings and the XML feeds. Custom developed software is used to support these processes.

The output from the production centre is then loaded into a SQL Server database and proprietary search engine, which are maintained and hosted by NLA.

2.2. File delivery and PDF access

The article PDFs are stored within the NLA's environment and are retrieved on demand by end-users following links issued to them by the PCA, usually in the form of an email or via a website page. PCAs can create these links for their customers in the following ways:

2.1. XML feeds

The PCA can receive an XML feed containing all available articles, their metadata and the article IDs needed to retrieve their associated PDFs. These are retrieved by the PCA during the night using FTP pull processes. The PCA must then search or read that feed for articles that match their briefs, utilising their own search technology. They may display to their operators the actual articles or pages for review, and then need to make the selected article PDFs available to their clients by sending them links to those PDFs.

2.2. IDS (Image Data Service)

The PCA can receive a feed containing the XML, article PDF & article thumbnail (jpe.g.) for all. These are retrieved by the PCA during the night using FTP pull processes. The PCA must then search or read that feed for articles that match their briefs, utilising their own search technology. They may display to their operators the actual articles or pages for review, and then need to make the selected article PDFs available to their clients by hosting them on the PCA’s system. PCA’s can co-mingle content from NLA & non NLA sources in this model.

2.3. Search model

The PCA can search the NLA’s XML store directly, entering keyword terms and retrieving a list of matching documents, which they may review for validity. The PCA is then able to select which links are relevant for clients, “package” all such links into client specific collections and then make these baskets of links to the PDF articles available to their clients.

Figure 1: Overview of the NLA Digital Database environment (XML access)

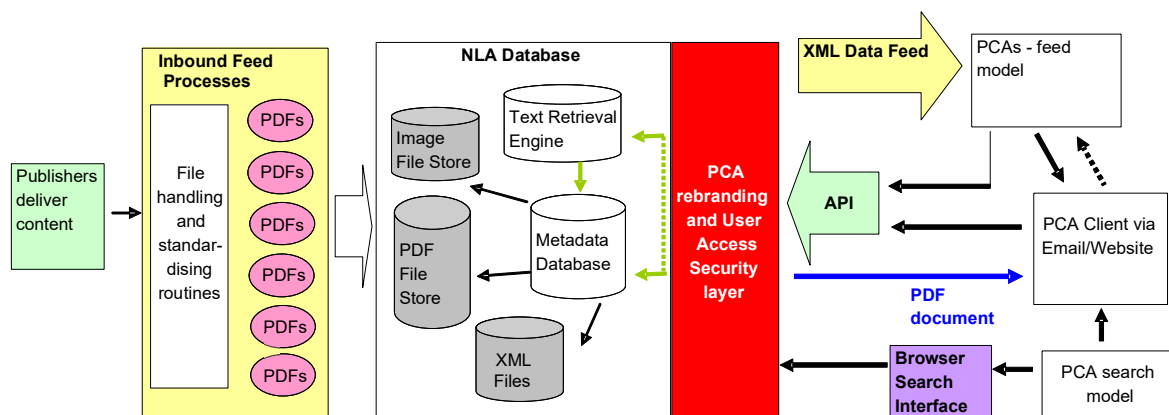
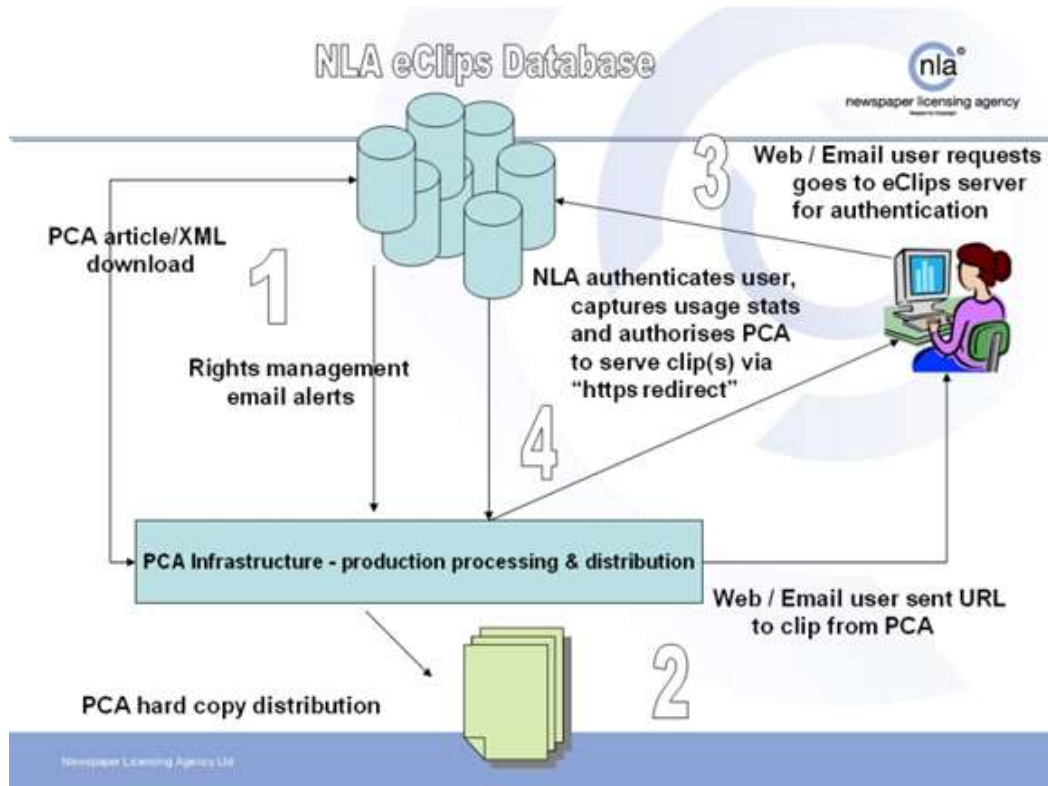


Figure 2: Overview of the NLA Digital Database environment (IDS access)



3. Security model

3.1. eClips authentication methods

There are two methods of authenticating users:

- 3.1.1. Username and password sign-on
- 3.1.2. IP address-based authentication

3.2. Secure HTTPS access

- 3.2.1. NLA only supports HTTPS access.
- 3.2.2. HTTPS access to NLA servers is encrypted between the browser and the server.

3.3. Username and password sign-on

- 3.3.1. Unless IP authenticated, eClips users are prompted for their username and password to access eClips.
- 3.3.2. Each individual user must have their own unique username. These usernames may not be shared.
- 3.3.3. This username must be unique across all eClips users
- 3.3.4. The username and password are stored encrypted in the user's cookie file, and provided this file exists, the user will only be prompted the first time they access eClips
- 3.3.5. Client administrators can create and manage usernames and passwords in real-time.
- 3.3.6. Usernames and passwords must be a minimum of 4 characters. There are no restrictions on length, formatting, or complexity though clients are encouraged to use a minimum 8-character password incorporating numbers, letters, and other characters.
- 3.3.7. Usernames and passwords do not expire and do not need to be periodically changed though clients are encouraged to do so.

3.4. IP address-based authentication

- 3.4.1. eClips supports single IP addresses, a range (e.g. 192.168.2.1 to 192.168.2.20) and wildcards (e.g. 192.168.2.*).
- 3.4.2. These may be used for NLA option 2 clients only. In each instance NLA will verify that a valid Option 2 licence exists and will require the client to confirm the IP address range will be used in context of the relevant licence only.
- 3.4.3. The PCA is able to verify their IP address by connecting to the following URL:<https://nla.co.uk/nlaapi.dll/DispIP>
- 3.4.4. eClips supports clients having multiple IP addresses

- 3.4.5. With IP based authentication users are not presented with a login screen
- 3.4.6. IP authenticated users can only access eClips outside of their office provided they are using a VPN or other mechanism that allows them to appear from a recognised IP address.
- 3.4.7. The supported IP addresses must be registered on eClips in advance and cannot be configured in real-time.

3.5. MyArchive authentication

- 3.5.1. Regardless of authentication method, MyArchive clients must create a unique username and password for each user that will be accessing the archive

3.6. Permanent cookies

- 3.6.1. All eClips usage requires the use of permanent cookies
- 3.6.2. Using cookies allows users to access eClips without requiring them to enter their login details once they have already done so

3.7. Authentication for the XML feeds

Customers receiving the XML or IDS feeds will need to poll the FTP site to determine if new content is available. To obtain the actual feeds, customers will connect to the NLA's FTP site. Rather using encryption or a secure VPN, this connection is via the public internet and requires a previously configured username and password set up by the NLA customer services team. Customers will then use FTP pull and these username/password details to connect to their unique secure areas on the NLA host, where they may access and download those files they are configured to receive.

4. Application programming interface (API)

All external communication with the NLA database is via API calls, which are URL based requests including parameters (rather than complex Web services or alternative approaches). An example of such a call would be:

<https://www.nla-eclips.com/nlaapi.dll/getpage?source=FTN&date=31122004&pagenumber=2>

In the above example, the [getpage](#) function is one of the possible functions accessible from the NLA-API.DLL.

These calls are of two major types:

- Content-oriented calls (allowing users to search or retrieve content objects).
- Administration-oriented calls (handling user, licence, and rights records)
 - Only two of these administration calls are described in this document – functionality supporting reviewing and modifying user and licence records will be documented when appropriate extranet facilities are made available.

The following core content NLA-API functions are supported:

Page	Function
SearchDatabase	Searches the article text for a match
GetPage	Retrieves the PDF or thumbnail JPG of a given page
CheckObject	Returns licence, rights, or metadata information about an object
GetObject	Retrieves a specific object (article) or list of objects
ManageArchive	Allows items to be added or deleted to an archive
ListArchive	Displays the archive contents for a given user
SearchArchive	Searches within a user's MyArchive for a match
PCAClientRequest	Provides authentication services to IDS PCAs

Each of these URL calls is described in detail below. Additionally, the following two administrative calls are described:

Page	Function
GetVersion	Displays the API version
DispIP	Displays the IP address for a requesting browser

4.1. NLA-API.DLL/SearchDatabase?

This call searches the entire article text (including headline, sub-headline, and by-line) for a match on a supplied text string. It returns a table of contents listing all the matching articles. Matching is done on a case-insensitive exact pattern match.

Search terms must be two or more characters in length (e.g. 3i or O2 is supported, but X is not). A “Stop Word” list also exists which prevents certain words from being searchable. These include restricted words (AND, OR, NOT) which are used by the application and a limited number of common words (THE, OF, IN, HE etc). The NLA client services team can discuss these should they present an issue.

4.1.1. Parameters Supported

Parameter	Mandatory?	Default	Example	Notes
SearchTerm	Yes	None	Iraq	URL encoded search string. Includes support for parentheses and Boolean operators AND, OR and NOT
ObjectType	No	Article	Article	Object types that should be searched (provides future support for photos, adverts etc This initially will only support “Article”)
Source	No	None	GDN	The 3 letter code for a specific newspaper (see Appendix for lists)
SourceSection	No	None		A URL encoded free text string referencing a specific newspaper section
SearchArea	No	Article		May be Article, Body, Headline, SubHeading, Byline
StartDate	No	7 Days ago	25122003	Format DDMMYYYY
EndDate	No	Tomorrow	01042004	Format DDMMYYYY
Edition	No	0	1	0 is all editions, 1 is the 1st, 2 is the 2nd and so on
Format	No	XML	ASCII	Defines the return type as XML, HTML or ASCII

Note: if a search term features an ampersand (&), it must be enclosed in quotes, e.g. <https://www.nla-eclips.com/nlaapi.dll/SearchDatabase?SearchTerm=Tony Blair&SourceSection=Leaders & Letters>

4.1.2. Examples

Example 1: [https://www.nla-eclips.com/nlaapi.dll/SearchDatabase?SearchTerm=Tony Blair OR War OR \(Saddam Hussain\)&StartDate=10052005](https://www.nla-eclips.com/nlaapi.dll/SearchDatabase?SearchTerm=Tony Blair OR War OR (Saddam Hussain)&StartDate=10052005)

Returns all stories published after 10-May-2005 featuring the search terms Tony Blair, War or Saddam Hussein

Example 2: <https://www.nla-eclips.com/nlaapi.dll/SearchDatabase?SearchTerm=Microsoft&SearchArea=Headline&Source=TIM&format=ASCII>

Returns all stories from The Times featuring Microsoft in the headline, returning the content in ASCII format

4.1.3. Output

The results from this call include the following fields for each found article:

- ArticleID
- Headline
- Source (e.g. Guardian, The Sun etc)
- Newspaper Section (e.g. Lex, Letters Page, War in Iraq etc)
- Newspaper SubSource (i.e. the Supplement) if appropriate
- Date
- Byline
- Edition (e.g. 1st, Last Edition)
- Word count

XML Example:

```
<NLAML>
<ARTICLE>
= <SourceMetaData>
  <Property FormalName="Publication_Name" Value="The Guardian" />
  <Property FormalName="Publication_SubSource" Value="G2" />
  <Property FormalName="Edition" Value="1" />
  <Property FormalName="Publication_Date" Value="20041225" />
  <Property FormalName="Day" Value="Monday" />
  <Property FormalName="Page_Section" Value="" />
  <Property FormalName="NLAArticleID" Value="450" />
</SourceMetaData>
= <DescriptiveMetaData>
= <HeadLine>
  <strong>Doubt cast on PM's 'nuclear threat' claim</strong>
</HeadLine>
= <ByLine>
  <strong>Richard Norton-Taylor</strong> in London
</ByLine>
  <Property FormalName="Wordcount" Value="635" />
</DescriptiveMetaData>
</ARTICLE>
</NLAML>
```

HTML Example:

```
<p>
<span class="toc_articleid">Article ID 1233: </span>
<a href="NLA-API.DLL/GetObject?objectid=1233"
class="toc_headline">Another night to forget for troubled Benitez</a>
<br>
<span class="toc_source">Daily Express</span>,</p>
```

```
<span class="toc_edition">1<sup>st</sup> edition</span>,  
<span class="toc_section">Sports Section</span>,  
<span class="toc_page">page 84</span>,  
<span class="toc_date">Thursday 20-Jan-2005</span>  
<br>  
<span class="toc_author">Paul Joyce</span>,  
<span class="toc_words">1,412 words</span>  
</p>
```

Which might be rendered in a browser as:

Article ID 1233: [Another night to forget for troubled Benitez](#)
Daily Express, 1st edition, *Sports Section*, page 84, Thursday 20-Jan-2005
Paul Joyce, 1,412 words

ASCII Example:

```
ArticleID: 1233  
Headline: Another night to forget for troubled Benitez  
Source: Daily Express  
Edition: 1st  
Section: Sports Section  
Page: 84  
Date: Thursday 20-Jan-2005  
Author: Paul Joyce  
Wordcount: 1412
```

4.1.4. Error Handling

Error messages are returned as plain text strings. The following results are returned:

- **“No results found”**: no matching articles have been found
- **“No search key specified”**: no valid search term has been supplied
- **“Invalid (start or end) date specified. Format is DDMMYYYY”**: invalid start or end dates supplied
- **“Invalid source specified”**: an unrecognised 3 letter newspaper source has been entered

Note that there is no error checking on the searcharea, edition or format fields (valid values are defined above) nor on the subsource field (which may have many possible values)

SearchArchive – works similar to SearchDatabase, but only searches within the user’s MyArchive. If no Start Date is specified, it defaults to one week before the current date.

Example: [https://www.nla-eclips.com/nlaapi.dll/SearchArchive?SearchTerm=Tony Blair or Gordon BrownWar](https://www.nla-eclips.com/nlaapi.dll/SearchArchive?SearchTerm=TonyBlair%20or%20Gordon%20BrownWar)

4.2. NLA-API.DLL/GetPage?

This call serves to display a full page in order for PCAs/others to review a full page to help determine a story's relevance to a customer brief, or for quality assurance/workflow purposes. It also allows the thumbnail of the full page to be retrieved/displayed alongside a clipping or in a user interface.

4.2.1. Parameters Supported

Parameter	Mandatory?	Default	Example	Notes
Source	Yes	None	DEX	A 3 digit code for the newspaper (as defined in the Appendix)
SubSource	No	Core	G2	Allows a named supplement/magazine to be requested
PageNumber	Yes	None	1, A	Each page must be called separately
Date	Yes	None	25122003	Format DDMMYYYY
Format	No	PDF	PDF	Allows either the PDF or the thumbnail JPG (parameter = JPG) to be retrieved
Edition	No	1	3	1 is the 1st edition, 2 is the 2nd and so on (0 is all editions, and is not supported externally)
Re.g.ion	No	Null	Eire	The re.g.ional version of the paper. E.g Eire, Ulster, etc.

4.2.2. Examples

Example 1:

<https://www.nla-eclips.com/nlaapi.dll/GetPage?Source=TIM&Date=18012005&PageNumber=1>

Gets the Times' front cover from the 18th January 2005

Example 2:

<https://www.nla-eclips.com/nlaapi.dll/GetPage?Source=TIM&Date=17012005&PageNumber=1&SubSource=T2&Edition=1>

Get the front cover of the Times' T2 supplement 1st Edition on the 17th January 2005.

4.2.3. Error Handling

Error messages are returned as plain text strings. The following error results are returned:

- **“No page found”**: no matching page has been found
- **“Invalid date specified. Format is DDMMYYYY”**: an invalid publication date has been supplied
- **“Invalid source specified”**: an unrecognised 3 letter newspaper source has been entered

Note that there is no error checking on the edition or format fields (valid values are defined above) nor on the subsource or paganumber fields (which may have many possible values)

4.3. NLA-API.DLL/CheckObject?

This call allows the PCA to find out relevant data about an object, its rights status, or metadata or licence terms, prior to display.

4.3.1. Parameters Supported

Parameter	Mandatory?	Default	Example	Notes
ObjectID	Yes	None	987654	The NLA ID for the object (i.e. the article, picture, page etc). This is a unique, incremented number in the range 1 to 999999999
InformationType	No	0	1	Identifies what data about the document to provide (0 = All, 1 = rights, 2 = licence, 3 = metadata).
Format	No	XML	ASCII	The returning format

4.3.2. Examples

Example 1:

<https://www.nla-eclips.com/nlaapi.dll/CheckObject?ObjectID=412&InformationType=1>

Return the rights information for object 4132 (in default, i.e. XML) format

Example 2:

<https://www.nla-eclips.com/nlaapi.dll/CheckObject?ObjectID=424&InformationType=2&Format=ASCII>

Returns the licence information for object 4132 and returns in ASCII format.

4.3.3. Output

The output information includes the following fields:

- Rights data:
 1. Rights_Status
 2. Restricted_To
 - a. Values include 0 (no one), -1 (publishers only), -2 (this publisher company only), -3 (this newspaper only). -4 (this paper’s staff (i.e no

- freelance etc)), -5 (this paper’s admin/authorised staff only)
- 3. Restricted_ReasonCode
 - a. This code value is only shown to authorised users (NLA staff etc)
- Licence data:
 - 4. User_Licence_Status
 - a. Values include 0 (user is not licensed for this content) and 1 (user is licensed for this content)
 - 5. Organisation_Licence_Status
 - a. Values include 0 (organisation is not licensed for this content) and 1 (organisation is licensed for this content)
- Object Meta data:
 - 6. ArticleID
 - 7. Headline
 - 8. Source (e.g. Guardian, The Sun etc)
 - 9. Newspaper Section (e.g. Lex, Letters Page, War in Iraq etc)
 - 10. Newspaper SubSource (i.e. the Supplement) if appropriate
 - 11. Page Numbers
 - 12. Date
 - 13. Byline
 - 14. Edition (e.g. 1st, Last Edition)
 - 15. Word count

XML Example:

```

<NLAML>
- <RightsData>
  <Property FormalName="Rights_Status" Value="0" />
  <Property FormalName="Restricted_To" Value="0" />
  <Property FormalName="Restricted_ReasonCode" Value="" />
</Rightsdata>
- <LicenceData>
  <Property FormalName="User_Licence_Status" Value="1" />
  <Property FormalName="Organisation_Licence_Status" Value="1" />
</Licencedata>
- <SourceMetaData>
  <Property FormalName="Publication_Name" Value="The Guardian" />
  <Property FormalName="Publication_SubSource" Value="G2" />
  <Property FormalName="Edition" Value="1" />
  <Property FormalName="Publication_Date" Value="20041225" />
  <Property FormalName="Page_Section" Value="" />
  <Property FormalName="NLAArticleID" Value="450" />
</SourceMetadata>

```



```

= <DescriptiveMetadata>
=   <HeadLine>
=     <strong>Doubt cast on PM's 'nuclear threat' claim</strong>
=   </HeadLine>
=   <ByLine>
=     <strong>Richard Norton-Taylor</strong> in London
=   </ByLine>
=   <Property FormalName="Wordcount" Value="635" />
= </DescriptiveMetaData>
</NLAML>

```

HTML Example:

```

<Body>
<p>
Rights status: 0<br>
Restricted to: 0<br>
Restricted code: null<br>
</p>
<p>
User licence status: 1<br>
Organisation licence status: 1<br>
</p>
<p>
NLAArticleID: <a href="GetObject.asp?objectid=1233"
class="toc_articleid">1233</a><br>
Headline: <span class="toc_headline">Another night to forget for
troubled Benitez</span><br>
Source: <span class="toc_source">Daily Express</span><br>
Sub Source: <span class="toc_subsource">Null</span><br>
Edition: <span class="toc_edition">1<sup>st</sup> edition</span><br>
Section: <span class="toc_section">Sports Section</span><br>
Page Numbers: <span class="toc_page">84</span><br>
Publication Date: <span class="toc_date">">20-Jan-2005</span><br>
Byline: <span class="toc_byline">Paul Joyce, London
Correspondent</span><br>
Word count: <span class="toc_words">1,412 words</span><br>
</p>
</body>

```

ASCII Example:

```

Rights status: 0
Restricted to: 0
Restricted code: null
User licence status: 1
Organisation licence status: 1
NLAArticleID: 1233
Headline: Another night to forget for troubled Benitez
Source: Daily Express
Sub Source: Null
Edition: 1st edition
Section: Sports Section
Page Numbers: 84
Publication Date: 20-Jan-2005

```

Byline: Paul Joyce, London Correspondent
 Word count: 1412

4.3.4. Error Handling

No object specified
 Invalid object specified

Error messages are returned as plain text strings. The following error results are returned:

- **“No object specified”**: no objectID was supplied
- **“Invalid object specified”** : no valid objectID has been supplied

Note that there is no error checking on the informationtype or format fields (valid values are defined above).

4.4. NLA-API.DLL/GetObject?

This call retrieves objects (articles) from the database.

4.4.1. Parameters Supported

Parameter	Mandatory?	Default	Example	Notes
ObjectID	Yes	None	987654	The NLA ID for the object (i.e. the article, picture, page etc)
BrandingID	No	None	EDS401	Identifies which PCA branding layer package to apply to the clipped PDF (only applies if format is PDF)
PCAID	No/Yes	None	413	A unique ID assigned to each PCA – must be used if branding is required
Format	No	PDF	PDF	Supported options are PDF and JPG
P1...P5	No	None		These parameters are for branding templates that PCAs define that require them

This call supports up to 100 objects being requested simultaneously and returned in one PDF. In such circumstances, each value for each parameter is separated by a hyphen.

4.4.2. Examples

Example 1:

<https://www.nla-eclips.com/nlaapi.dll/GetObject?ObjectID=456&Format=JPG>

Retrieves the clipped article JPG of the object with ID 456

Example 2:

<https://www.nla-eclips.com/nlaapi.dll/GetObject?ObjectID=456&BrandingID=TNS213&PCAIID=4>

Retrieves the PDF of the clipped article 456 with the PDF including any branding (i.e. logos / metadata etc) as previously defined in the branding package TNS213.

Example 4:

<https://www.nla-eclips.com/nlaapi.dll/GetObject?ObjectID=412-413-414>

Retrieves the PDFS for objects 412, 413 and 414

Example 5:

<https://www.nla-eclips.com/nlaapi.dll/GetObject?ObjectID=456-513&BrandingID=13&PCAIID=4&P1=Clip One-Clip Two>

Retrieves the PDF of the clipped articles 456 and 513, with the PDFs including any branding (i.e. logos / metadata etc) as previously defined in the branding package 13 and with the value of “Clip One” inserted into the %P1 field for the first object, and “Clip Two” for the second object.

Example 6: mobile support

<https://www.nla-eclips.com/nlaapi.dll/GetObject?ObjectID=412&platform=Blackberry>

Since a few mobile devices do not support PDFs natively, the above parameter causes the API to create a PNG-HTML version of the article, making it easier to display on mobile devices. The article also has a navigation bar to help the user to move from one page to the next.

4.4.3. Error Handling

Error messages are returned as plain text strings. The following error results are returned:

- **“No object specified”**: no objectID was supplied
- **“Invalid object specified”** : no valid objectID has been supplied
- **“User does not have sufficient rights to access the object”**: an object has been requested that this user does not have rights to
- **“Invalid branding package specified”** : an invalid brandingID has been supplied
- **“Access to this article has been restricted by the publisher. It has either been deemed as potentially libellous to the publisher, or the publisher does not own the copyright and therefore we cannot release the clipping for viewing.”**: if the article has been restricted due to Rights or Libel issues.

4.5. NLA-API.DLL/PCAClientrequest

This call is used by IDS PCAs to authenticate users against the NLA digital database and check for licence status. If the client has a valid licence to the article, then the API call redirects the user to a link on the PCA portal. In case the user does not have a licence or the article has been restricted, the API passes that information back to the PCA for appropriate error handling procedures.

4.5.1. Parameters Supported

Parameter	Mandatory?	Default	Example	Notes
ObjectID	Yes	None	987654	The NLA ID for the object (i.e. the article, picture, page etc)
PCA ID	Yes	None	413	A unique ID assigned to each PCA – must be used if branding is required
PCAURL	Yes	None	https://pca.com	The URL to which the API should redirect the user to

4.5.2. Examples

Example:

https://www.nla-eclips.com/nlaapi.dll/PCAClientRequest?PCA_ID=35&ObjectID=456&PCAURL=https://www.google.com/links/9999

On successful authentication, this call will redirect the user to www.google.com/links/9999

In cases where the user authentication does not pass, the user is redirected to www.google.com/links/9999&omit=456

4.5.3. Error Handling

Error messages are returned as plain text strings. The following error results are returned:

- **“No object specified”**: no objectID was supplied
- **“Invalid PCA_ID specified”** : an invalid PCAID has been supplied
- **“No PCA URL specified”** : a redirect URL has not been supplied

4.6. NLA-API.DLL/ManageArchive?

Users may need to be able to add / delete items from their MyArchive, and display the contents of a MyArchive. The ability to use the Archive must be validated against the user’s licence.

4.6.1. Parameters Supported

Parameter	Mandatory?	Default	Example	Notes
ORGID	Yes	None	123	The Org ID of the client
ObjectID	Yes	None	987654	The NLA ID for the object (i.e. the article, picture, page etc) – mandatory unless Action=List
Action	Yes	None	Add	Can be Add or Delete

4.6.2. Examples

<https://www.nla-clips.com/nlaapi.dll/ManageArchive?OrgID=123&ObjectID=456&Action=Add>

Adds article 456 to the MyArchive of Org 123

4.6.3. Error Handling

Error messages are returned as plain text strings. The following error results are returned:

- **“Invalid Org ID”** : no valid OrgId has been supplied
- **“No object specified”**: no objectID was supplied
- **“Invalid object specified”** : no valid objectID has been supplied
- **“User does not have archive rights”**: the user or Org does not have my archive rights

4.7. NLA-API.DLL/ListArchive?

4.7.1. Parameters Supported

Parameter	Mandatory?	Default	Example	Notes
Format	No	XML	ASCII	Can be XML, ASCII or HTML
OrgID	Yes	Cookie	123	OrgID of the requesting client

4.7.2. Examples

<https://www.nla-eclips.com/nlaapi.dll/ListArchive?OrgID=123&Format=HTML>

Displays the MyArchive for Orgr 123 in HTML format

4.7.3. Error Handling

Error messages are returned as plain text strings. The following error results are returned:

- **“Invalid Org ID”** : no valid OrgId has been supplied
- **“User does not have archive rights available”**: this user does not have an archive right
- **“No articles found in the archive”**: this Org’s archive is empty

4.8. NLA API administration calls

4.8.1. NLA API.DLL/GetVersion

This call displays the version information associated with the NLA API being accessed.

Example:

<https://www.nla-eclips.com/nlaapi.dll/GetVersion?>

The typical output would be:

Module: NLA Database API DLL. Version:1.0f.

4.8.2. NLA API.DLL/DispIP

This call displays the IP address from which the request is received. This information is required by the client services team when registering a client/user who is IP authenticated.

Example:

<https://www.nla-eclips.com/nlaapi.dll/DispIP?>

The typical output would be:

Your calling IP address is 212.229.249.100. Which translates to 1694098900

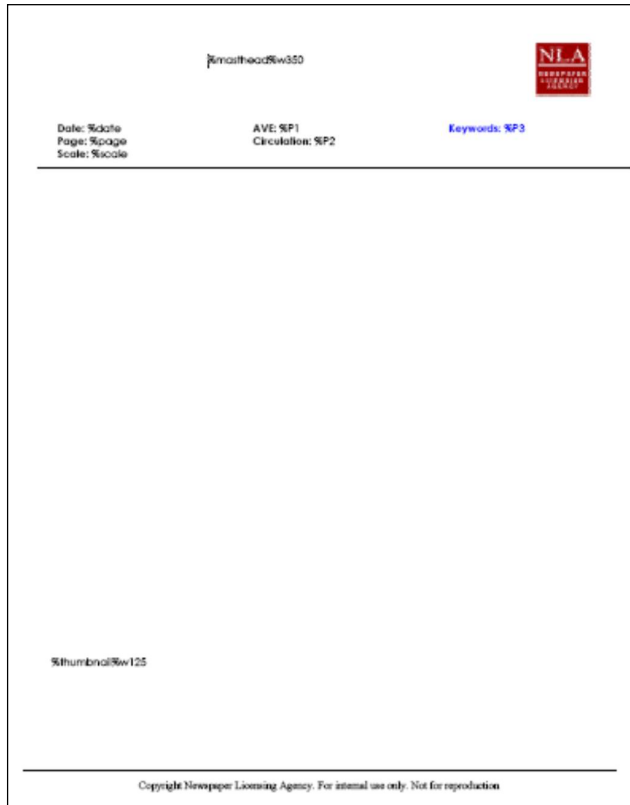
5. Branding

Branding support exists within the NLA Digital Database in order to allow PCAs to present clippings to their users that have some element of presentation defined by the PCA. To utilise branding, the PCA must create a PDF that specifies the branding elements, upload it (and any related images) onto the NLA Digital Database system and include that branding template as part of any API request for an article (as defined in the NLA API.DLL/GetObject? section of this document)

- 5.1. The PCA is able to define up to 1 branding package per client plus 5 additional packages
 - 5.1.1. The PCA maintains these packages themselves
 - 5.1.2. New packages may be added via the Client Services team.
- 5.2. For each branding package, the PCA must submit a valid template PDF and any accompanying logos
 - 5.2.1. The PCA must warrant that these files are virus-free
- 5.3. The branding template allows the PCA to specify the location of the following standard variables
 - 5.3.1. %date (the publication date)
 - 5.3.2. %page (the page number[s])
 - 5.3.3. %thumbnail (the thumbnail for the page)
 - 5.3.4. %edition (the numbered edition)
 - 5.3.5. %bookname (the name of the supplement)
 - 5.3.6. %re.g.ion (the re.g.ion of the title)
 - 5.3.7. %articleid (the eClips article ID)
- 5.4. Image values (logos, thumbnails and mastheads) will be un-scaled unless the parameter string contains a sub-parameter %h or %w. In each case the sub-parameter introduces a value measured in points, representing the finished height or width of the image respectively, e.g. to paste a thumbnail with a height of 1 inch, the parameter string in the template PDF would be %thumbnail%h72.0.
- 5.5. The PCA is able to specify their own parameters
 - 5.5.1. Five additional parameters (%P1, %P2, .. %P5) are defined

In the following example, the template includes a logo and copyright notice and some custom variables (AVE, keywords, circulation) in addition to some standard variables (page, masthead, date, thumbnail).

Figure 3: Sample branding PDF template

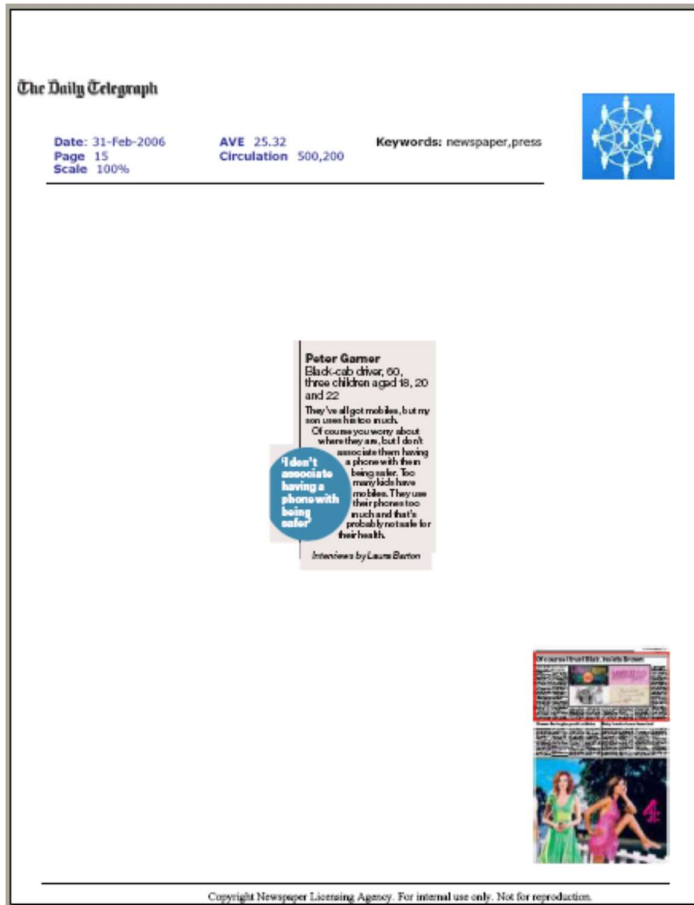


The resultant output combines the template with the story (noting that the thumbnail does not in fact belong to that story) according to the following URL call:

[https://nla-eclip.com/nlaapi.dll/getobject?
ObjectID=1&BrandingID=A1&PCAIID=4&P1=25.32&P2=435,012&P3=World
News](https://nla-eclip.com/nlaapi.dll/getobject?ObjectID=1&BrandingID=A1&PCAIID=4&P1=25.32&P2=435,012&P3=WorldNews)

Figure 4: Sample branded article

The following example includes a possible PCA/client logo, a masthead for the newspaper, metadata for page, date and scale that are from the database, and user supplied variables for AVE and circulation. The thumbnail is intentionally situated bottom right.



6. Character set support

- 6.1. Within the XML feed and in any XML formatted response to an API call, the data is presented in UTF-8 encoding
- 6.2. Characters are preserved as printed in the PDF. No translation or remapping is done via the search engine or within the XML feed.

The NLA Digital Database renders the content as it was published, and thus a search on “Deutsche Börse” would not find articles mentioning “Deutsche Börse”. Similarly, a search for “naive” would not find articles in newspapers that used “naive”.

It is intended that a later version of the NLA Digital Database will add additional support for searching that will minimise the impact of foreign characters on PCAs, in order to improve hit rates and reduce any translation burden for PCAs.

7. Style sheets

The content returned by the NLA API calls may be delivered in HTML format. While this content would need to be re-formatted by any calling application, there is an inbuilt style sheet and series of styles that are defined.

- 7.1. The default style sheet entry in any HTML is “NLA.CSS”
- 7.2. Style elements are used by the SearchDatabase and ChckObject API calls
- 7.3. There are a several style elements applied
 - 7.3.1. toc_articleid – spans an object ID
 - 7.3.2. toc_headline – spans an article headline
 - 7.3.3. toc_source – spans an article’s newspaper
 - 7.3.4. toc_edition – spans an article’s newspaper edition
 - 7.3.5. toc_section – spans an article’s newspaper section
 - 7.3.6. toc_page – spans an article’s page(s)
 - 7.3.7. toc_date – spans an article’s publication date

8. Content feeds

The types of content that the NLA may publish nightly to its feed customers are:

- **Article Feed**
This is the core feed of articles in XML format that customers will use to review articles for cuttings relevance. Details as above in the delivery section.

- **Rights Update emails**
This is an email sent at regular intervals detailing any articles (or components of articles) for which republication rights are restricted in some fashion.

This is made available to PCAs, whether they are feed, search or other types of NLA customers, in order to ensure that all PCAs have access to the relevant rights information for the material they are reproducing.

- **Rights Update feed**

This is a file that is made available in the PCA's ftp folder as soon as an article has been restricted.

- **Newspaper Flat-plans**

This is an inventory report available each morning to all feed customers summarising the articles available on each page of each newspaper in order to assist with customer processing checklists.

8.1. Article feeds

This is the core feed that customers will utilise in order to determine which articles are of relevance. The content delivery times are as detailed in individual PCA contracts. The PCA is expected to poll the appropriate FTP server/directory and retrieve each file as it becomes available.

- 8.1.1. Each newspaper article (where an article may span multiple pages) will have its own file.
- 8.1.2. Files will become available as soon as a complete article has been processed (subject to any publisher embargo, since the NLA receives content ahead of publication)
- 8.1.3. Each filename will be of the format `yyyymmdd_ttt_RE.G.-null_SUP-null_ED-01_nnn.XML`

- 8.1.3.1. `yyyymmdd` is the publication date in year, month, day format
- 8.1.3.2. `tttt` is the fullname of the newspaper
- 8.1.3.3. `RE.G.` indicates the regional edition with the edition name following the hyphen (NULL is used to represent the default primary edition)
- 8.1.3.4. `SUP` indicates the supplement (or other non-core book component) with the name following the hyphen (NULL is used where the article is from the main newspaper book)
- 8.1.3.5. `ED` indicates the edition with the number following the hyphen
- 8.1.3.6. `nmn` is a number uniquely assigned to each article
- 8.1.3.7. An example would be that an article from the G2 section of the Guardian's Christmas Day 1st London edition might be: `20041225_Guardian_RE.G.-null_SUP-G2_Ed-01_41653.XML`

- 8.1.4. All files for core national content will be available prior to 4am, except under circumstances where newspapers publish final editions after 2am
- 8.1.5. Articles changed in the last edition (or other changes, including rights or le.g.al changes) will be included in later files.
- 8.1.6. Only first edition articles and those articles from pages changed in subsequent editions will be made available (i.e. if the exact same article appears in the first and last editions, only one article – the first - will appear in the feed unless the page it was on changed)
- 8.1.7. These files will contain only the content to which the customer is licensed.
- 8.1.8. These files will be deleted from the FTP server at 23:00 on the day of publication.

For each article within the feed, the following fields will exist:

- 8.1.9. **NLAArticleID**
- 8.1.10. **Publication_Name**
 - 8.1.10.1. This is the full publication name
- 8.1.11. **Publication_Acronym**
 - 8.1.11.1. This is an acronym for the publication
- 8.1.12. **Publication_SubSource**
 - 8.1.12.1. This is the full name of the supplement or similar where appropriate
- 8.1.13. **Edition**
- 8.1.14. **Re.g.ion**
 - 8.1.14.1. Default is null (e.g. Primary edition, such as London edition for the nationals)
- 8.1.15. **Publication_Date**
 - 8.1.15.1. in the form `DDMMYYYY`
- 8.1.16. **Page_Section**
 - 8.1.16.1. This field may be blank
- 8.1.17. **Page_Numbers**
 - 8.1.17.1. Each page number is separated by a comma

- 8.1.18. **Headline**
- 8.1.19. **SubHeadline**
 - 8.1.19.1. This field may be blank
 - 8.1.19.2. There may be multiple instances
- 8.1.20. **Byline**
 - 8.1.20.1. This field may be blank
- 8.1.21. **Author**
 - 8.1.21.1. This field may be blank
 - 8.1.21.2. It is a subset of the byline field
 - 8.1.21.3. Each author is separated by a comma
- 8.1.22. **Wordcount**
- 8.1.23. **Origin_Left**
 - 8.1.23.1. For articles spanning multiple pages, there will be several values delineated by commas
- 8.1.24. **Origin_Top**
 - 8.1.24.1. For articles spanning multiple pages, there will be several values delineated by commas
- 8.1.25. **Height**
 - 8.1.25.1. Defined in millimetres
 - 8.1.25.2. For articles spanning multiple pages, there will be several values delineated by commas
- 8.1.26. **Width**
 - 8.1.26.1. Defined in millimetres
 - 8.1.26.2. For articles spanning multiple pages, there will be several values delineated by commas
- 8.1.27. **Article Area**
 - 8.1.27.1. This is the area in sq. cms, that the article occupies on the page
 - 8.1.27.2. For multi page articles, each value is separated by a comma
- 8.1.28. **Article percentage**
 - 8.1.28.1. This is the percentage area that the article occupies on the page
 - 8.1.28.2. For multi page articles, each value is separated by a comma
- 8.1.29. **Caption**
 - 8.1.29.1. Each picture's caption, if available, will be listed
 - 8.1.29.2. There will be one caption entry for each picture
- 8.1.30. **Body**
 - 8.1.30.1. This is the body text of the story
- 8.1.31. **Info**
 - 8.1.31.1. Info fields contain text not in the body text but nonetheless relevant, such as contents of tables or charts
- 8.1.32. **Rights status**
 - 8.1.32.1. Status codes supported are 0 (full republication), 1 (no republication)

Note that the height & width fields define the largest rectangle required to fully contain the article (thus if the first column is 6 inches and the second column is only 1 inch, the area defined will be 6 inches x 2 inches = 12 inches. Note that these values are given in the database in millimetres)

If items are blank, the field will be included with zero contents, e.g.
<subheadline></subheadline>

Example output is included in the Appendices.

9. Quality and accuracy levels

Quality is defined as the fitness of the application for its purpose: supporting the overnight press cuttings agencies' delivery of cuttings to clients by 7am. Its dimensions are therefore defined as:

- Accuracy of the XML for search purposes
- Timely availability of the content covered
- Completeness of the content
- Clipping presentation
- Article delivery performance

9.1. XML accuracy levels

The XML output from the articles is managed to quality criteria in order to ensure accurate searching and presentation of the data. It is not initially intended to support applications such as online news services that require enhanced reproduction of 100% accurate content. The following levels are defined:

- 9.1.1. Accuracy is measured against the actual contents of the newspapers. Where the original content features spelling mistakes, hyphenation or other typographical errors, these will form part of the benchmark.
- 9.1.2. No efforts are made to ensure accuracy of formatting (i.e. bold or italic markers) within the XML
- 9.1.3. Headlines and authors will be rendered 100% accurately
- 9.1.4. Articles will be reproduced to accuracy levels far in excess of existing OCR solutions
 - 9.1.4.1. Accuracy levels of 5 word errors per 1,000 words (99.5% accuracy) will be sought
 - 9.1.4.2. Accuracy levels of 7 character errors per 1,000 characters (99.3% accuracy) will be sought

9.2. Timely content availability

The content will be made available to PCAs as soon as possible after publication:

- 9.2.1. The NLA will seek to have all 1st editions available by 1am
 - 9.2.1.1. Provided newspapers publish 1st editions before 11pm
- 9.2.2. The NLA will seek to have all last editions available by 4am
- 9.2.3. Provided newspapers publish last editions before 2am
 - 9.2.3.1. In those extreme circumstances where publishers release newspapers "late", the NLA will endeavour to make all pages available within 3 hours of receipt

9.3. Completeness of content

- 9.3.1. The content available will be as defined in the coverage section of this document.
- 9.3.2. For those newspaper titles listed, all ‘books’ (i.e. the core newspaper, each supplement, any magazines etc) will be included except where explicitly noted.
- 9.3.3. Only pages featuring editorial content - as defined in the clipping rules section of this document – are included (i.e. full page adverts will not be featured)
- 9.3.4. Only articles that are not rights restricted or legally restricted will be available for display

9.4. Clipping presentation

- 9.4.1. The articles will be clipped to conform to the clipping presentation section of this document.
- 9.4.2. All articles will be clipped programmatically according to rules but may be subject to manual reformatting if required.

9.5. Article delivery performance

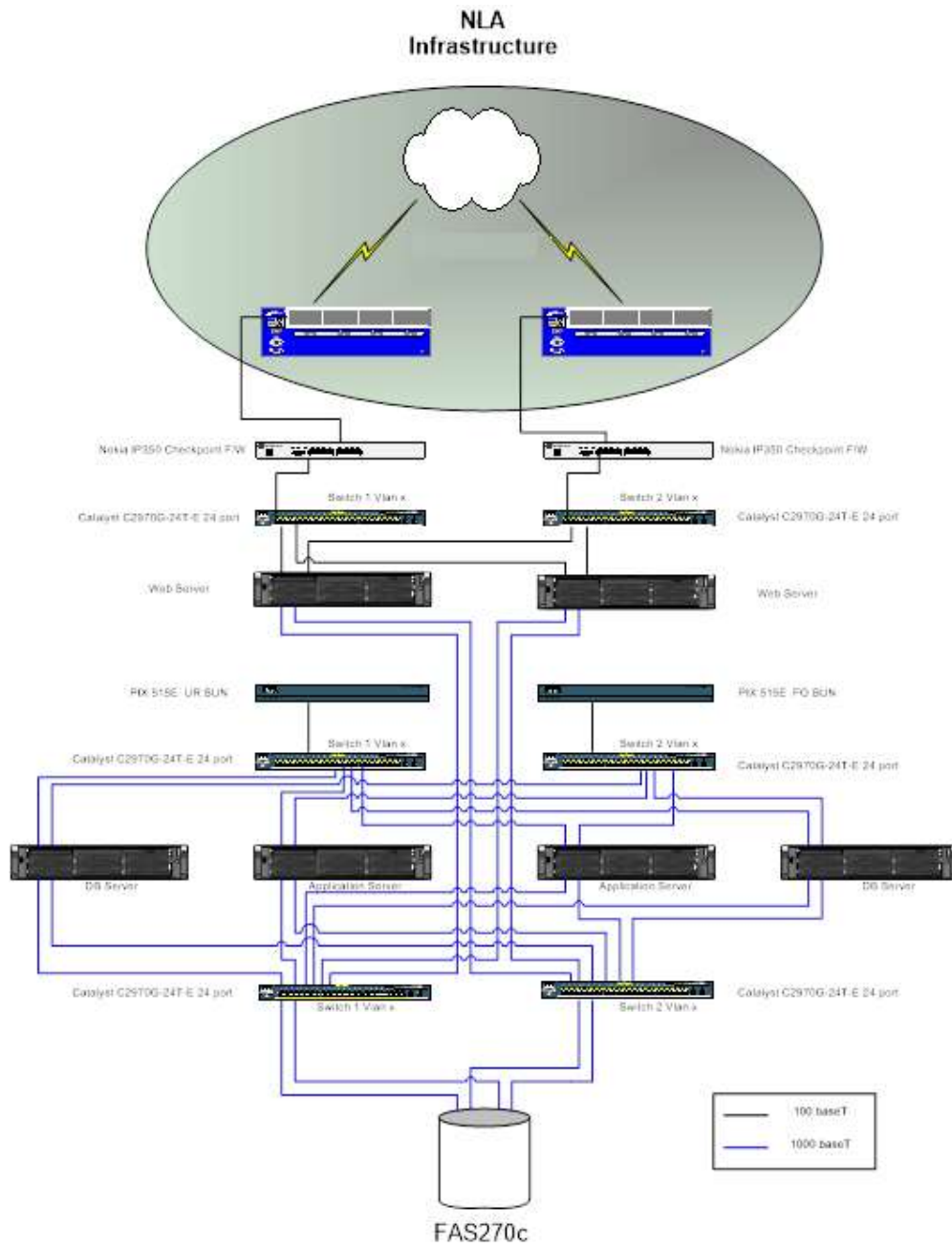
The technical architecture underpinning the NLA Digital Database, within reasonable commercial confidence, can be reviewed upon request. Performance benchmarks can also be discussed.

- 9.5.1. The NLA Digital Database is hosted by BSG, a leading provider listed on the London Stock Exchange and supporting organisations such Barclays and Le.g.al & General. They utilise Level 3’s communications infrastructure and have redundancy through Colt.
- 9.5.2. The NLA application is served by multiple, fully resilient and load-balanced web servers.
- 9.5.3. The application uses industry standard technology and software, including Microsoft SQL Server as the database engine.
- 9.5.4. The environment uses multi-gigabit switches and SAN storage in order to deliver maximum throughput

9.5.5. The system has been architected to support peak-time access of more than 30,000 users

The following diagram is a conceptual outline of the NLA's architecture and infrastructure (for example, the diagram is not an exact representation of how many components are in use for each of the conceptual components, nor does it define the specific specifications and manufacturers of component parts). It demonstrates the system's scalability and resilience/redundancy - there is no single point of failure anywhere in the system.

Figure 5: Architecture overview



10. Coverage

The newspapers will be released in ‘content packages’. A detailed list of content is available on the eClips blog under Documentation → NLA eClips Content List

<https://blog.nla-eclips.com/documentation/NLA%20eClips%20Content%20List%20v8.0.doc>

11. Rights, security and legal implications

11.1. Rights

The system is designed to ensure that copyright sensitive material is not inappropriately accessed by users.

- 11.1.1. All published articles will be available within the XML feed. The XML feed will indicate if articles are not available for onward republication
- 11.1.2. Email notices will be issued to XML customers where articles need to be withdrawn from publication
 - 11.1.2.1. All articles, in whatever format, indicated by such emails must be removed from all database systems within 24 hours of receipt of the email
 - 11.1.2.2. All customers must confirm receipt of the notification email and content removal within 24 hours of receipt
- 11.1.3. Users searching for, or linking to, content that has a restricted rights status will see an appropriate information message and will not be able to access the content

11.2. Corrections and libellous materials

- 11.2.1. Email notices will be issued to XML feed customers where documents need to be corrected or withdrawn
 - 11.2.1.1. All articles, in whatever format indicated by such emails must be corrected or removed from all database systems within 24 hours of receipt of the email
 - 11.2.1.2. All customers must confirm receipt of the notification email and content correction or removal within 24 hours of receipt
- 11.2.2. Emails will be issued to customer organisations where they have accessed libellous material
 - 11.2.2.1. Emails will be sent to the key contacts declared to the NLA
- 11.2.3. Articles that have been corrected will be adjusted within the NLA database to reflect the correction
- 11.2.4. Articles that have been withdrawn for le.g.al reasons will display a message to that effect within the NLA database if that article is subsequently accessed

11.3. Adobe security features

- 11.3.1. Adobe PDF security will be used to prevent copying and pasting content from PDFs
- 11.3.2. Each PDF will be enabled for printing

- 11.3.3. Each PDF will be separately identified to aid tracking onward distribution
- 11.3.4. Adobe’s inbuilt password features will not be enabled
- 11.3.5. Adobe and document control security features may be enabled to support prevention of forwarding, automatic expiration of content and removing the ability to save documents locally. The NLA will advise customers with reasonable notice should it be necessary to activate these or similar DRM features.

12. Pictures, images and photos

- 12.1. Images, pictures and photos will be captured during the NLA processing and may be available in the database subject to rights status.
- 12.2. Captions will be included where available
- 12.3. Photographer and copyright owner will be detailed where identified in the newspaper
- 12.4. Images will be displayed only as part of the clipping – they will not be separately available as searchable or retrievable database objects
- 12.5. Images are captured at full resolution but will be made available at 72 DPI
- 12.6. Images may be over-stamped with a copyright notice
- 12.7. Invisible digital watermarking will not initially be embedded in pictures

Figure 6: Sample low-resolution photo with over-stamping



Copyright Material©

- 12.8. Text within images will not be captured

Figure 7: Image with embedded text

In the following example the Emirates and Carlsberg brands would not be added to the picture’s metadata.



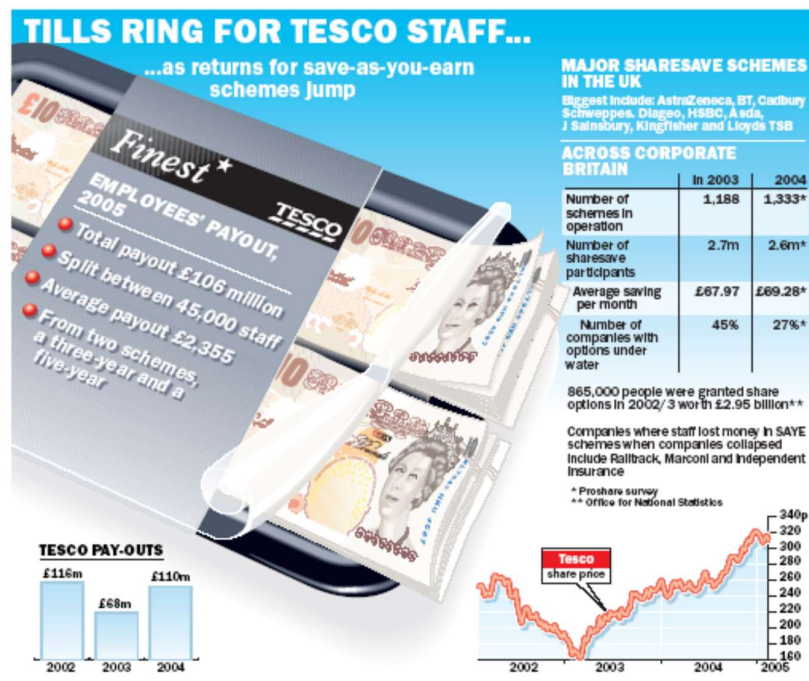
Copyright Material©

12.9. Vector graphics, such as charts, diagrams, tables and so forth, may have text within them and this will be extracted where possible, indexed and presented in the XML feed, but not as body-text (i.e. it will not “read” well though it will be searchable)

In the following example, the XML for the bottom of the graphic might look like:

<p>Tesco payouts £116m £68m £110m 2002 2003 2004, 340p 320 280... etc

Figure 8: Vector graphics with text



Copyright Material©

13. Clipping rules

The NLA recognises that the range of content that customers are interested in is diverse. Consequently, its starting premise is to make available all content that it is legally and technically able to. However, there will be some content components, for example crosswords, that may not be relevant

In general, key concepts the NLA will use when considering an article for clipping include:

- Cover all editorial materials associated with a newspaper (such as supplements and magazines)
- Include all content that is by-lined or clearly an editorial piece or a qualitative/added-value item
- Include all content that features company, person or product names
- Exclude non-unique, non-company content that is comparatively consistent from day to day (such as index levels, currency and commodity tables and similar market data, though share prices are captured)

A detailed list of clipping guidelines is available on the eClips blog under Documentation → Clippings guidelines – general treatment rules

<https://blog.nla-eclips.com/documentation/Clipping%20Guidelines%20-%20general%20treatment%20rules%20v19.pdf>.

14. Clipping presentation

14.1. Thumbnails

- 14.1.1. Adobe’s in-built thumbnails and bookmarks will not be included in the PDF
- 14.1.2. Thumbnails of the page from which a clipping is obtained will be included on all clippings
- 14.1.3. PCAs using the branding package may define the location for the thumbnail
- 14.1.4. The thumbnail will highlight the location of the clip within the page
- 14.1.5. Where an article spans multiple pages, each PDF page will display the page thumbnail appropriate to that clipped article element

Figure 9: Thumbnails



Copyright Material©

14.2. Page sizes and numbers

- 14.2.1. Clippings will be presented in the PDF at the centre top of each page
- 14.2.2. All articles are captured in their full size
- 14.2.3. All clipping PDFs will be optimised for printing in A4 portrait
- 14.2.4. Articles may be reduced up to 85% in order to fit onto a single A4 portrait page
- 14.2.5. For certain titles, spreads and articles that do not fit an A4 page at 85% reduction will be re-formatted

Figure 10: Handling spreads



Copyright Material©

14.3. Metadata presentation

- 14.3.1. PDFs will detail key metadata relating to the article. These will include:
 - 14.3.1.1. Publication name, which may be presented as a masthead image
 - 14.3.1.2. Publication date
 - 14.3.1.3. Page numbers

- 14.3.2. The metadata for a clipping may be stored within PCA systems for as long as the clipping itself may be displayed to users, as defined in the licence (currently 7 days)
- 14.3.3. Circulation data or advertiser value equivalents will not be included.
 - 14.3.3.1. PCAs may include such data using a branding package
- 14.3.4. Using a branding package, PCAs may customise the layout, formatting and labelling of the metadata

Figure 11: Sample metadata in a clipping



14.4. Highlighting

Highlighting of search terms will not initially be available within a PDF. It is anticipated that this may become available, possibly as a premium option, subsequent to further discussion with potential customers. In such an option, the customer will need to pass the ‘search term’ into the NLA database for the highlighting to be applied.

15. Acceptable usage policy

The NLA Digital Database supports legitimate use by PCAs and their clients accessing the service for the press cuttings application. There are particular restrictions surrounding programmatic use of the database:

15.1.1. The NLA Digital Database must not be accessed by any programmatic tool initiating more than 1 API request per second over a 5 second period

15.1.1.1. The NLA Digital Database will disable any account from which it appears to be receiving denial of service type activity

15.1.2. The NLA Digital Database API calls may be initiated by any legitimate application, including custom controls or programmatic and non-browser-based tools.

The above should be read in conjunction with the standard terms of the licence agreement and the terms of beta usage agreement. These include the following elements:

15.1.3. Each individual user must have their own unique user credentials. These may not be transferred between users.

15.1.4. Users may not store PDFs – they must always be accessed directly from the NLA digital database using an approved API call or user interface selection.

15.1.5. Users may not forward on PDFs to users

15.1.6. Users may not copy and paste from PDFs, take screenshots of them or repurpose them.

15.1.7. Users may not be presented with articles to which they are not licensed, nor with articles that do not have republication rights

15.1.8. XML body text may not be stored or repurposed

16. The NLA Digital Database search interface

PCAs who do not have search technology or do not wish to use the XML feed may instead search the NLA Digital Database directly and use it to create their own baskets of links to be sent to customers.

A PCA's principal activity will be to access the NLA Digital Database searching facilities and identify clippings that are relevant to their clients' briefs.

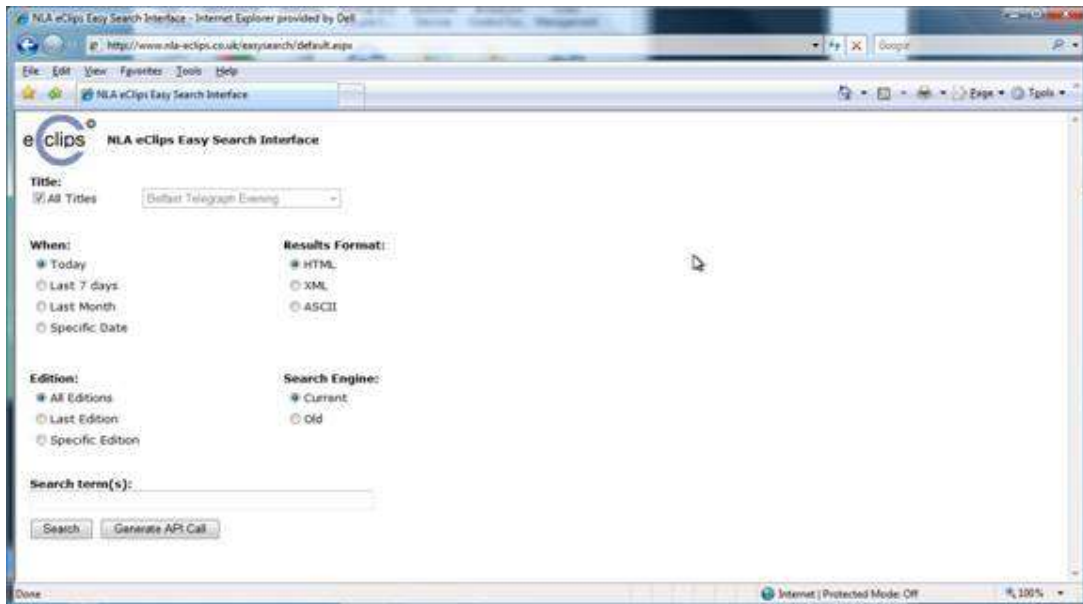
The searching is analogous to the API call SearchDatabase and features the same core selection criteria:

- Keyword
- Searcharea (including headline, byline or all text)
- Publication
- Date range (either absolute dates or relative to the usage date)

The SearchDatabase API call can be used to automate the search function within PCA systems. But for PCAs who would rather use an external interface, the NLA EasySearch interface is available as detailed below & can be accessed at

<https://www.nla-eclips.com/easysearch/default.aspx>

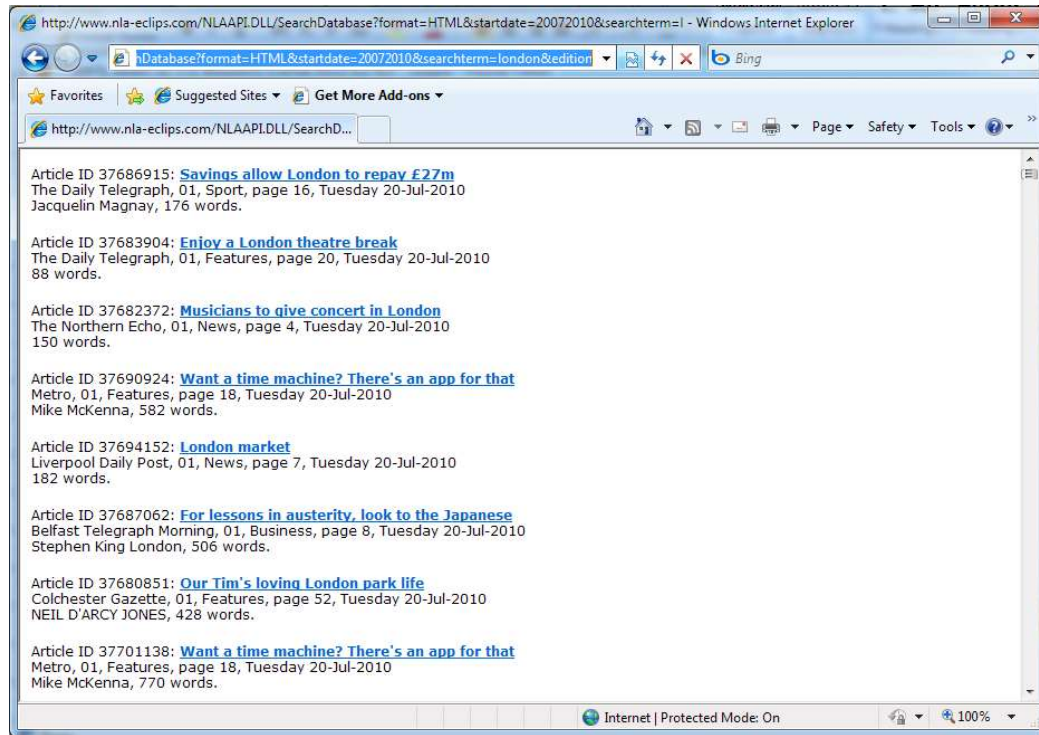
Figure 12: NLA Digital Database interface search screen



The UI can be accessed without passing any credentials. However, executing a search query will require the user to enter eClips credentials.

This search will return a table of contents listing all matching articles and links to those articles.

Figure 13: NLA Digital Database interface search results



The links can be made available on a PCA portal or sent via email to clients.

17. Appendix one: newspaper Three Letter Acronyms (TLAs)

The updated list of three letter codes required for the “Source” parameter in the API calls can be found here –

<https://blog.nla-eclips.com/documentation/eClips%20title%20list%20external.xls>

eClips Blog → Documentation → eClips Publication Details

18. Appendix two: sample XML feed

18.1. XML DTD

```

<!ELEMENT NLAML (SourceMetaData ,Article ) >
<!ELEMENT SourceMetaData (Property*) >
<!ELEMENT Property EMPTY >
  <!ATTLIST Property FormalName CDATA #IMPLIED >
  <!ATTLIST Property Value CDATA #IMPLIED >
<!ELEMENT Article (DescriptiveMetadata ,DataContent ) >
<!ELEMENT DescriptiveMetadata (HeadLine ,SubHeadLine ,ByLine ,Author
,Property*) >
<!ELEMENT HeadLine (#PCDATA) >
<!ELEMENT SubHeadLine (#PCDATA) >
<!ELEMENT ByLine (#PCDATA) >
<!ELEMENT Author (#PCDATA) >
<!ELEMENT DataContent (body ) >
<!ELEMENT body (#PCDATA) >

```

18.2. XML Schema

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="NLAML" type="NLAMLType"/>
  <xs:complexType name="NLAMLType">
    <xs:sequence>
      <xs:element name="SourceMetaData" type="SourceMetaDataType"/>
      <xs:element name="Article" type="ArticleType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="SourceMetaData" type="SourceMetaDataType"/>
  <xs:complexType name="SourceMetaDataType">
    <xs:sequence>
      <xs:element name="Property" type="PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Property" type="PropertyType"/>
  <xs:complexType name="PropertyType">
    <xs:sequence/>
    <xs:attribute name="FormalName" type="xs:string"/>
    <xs:attribute name="Value" type="xs:string"/>
  </xs:complexType>
  <xs:element name="Article" type="ArticleType"/>
  <xs:complexType name="ArticleType">
    <xs:sequence>
      <xs:element name="DescriptiveMetadata"
type="DescriptiveMetadataType"/>
      <xs:element name="DataContent" type="DataContentType"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="DescriptiveMetadata"
type="DescriptiveMetadataType"/>
  <xs:complexType name="DescriptiveMetadataType">
    <xs:sequence>
      <xs:element name="HeadLine" type="HeadLineType"/>
      <xs:element name="SubHeadLine" type="SubHeadLineType"/>
    </xs:sequence>
  </xs:complexType>

```



```

    <xs:element name="ByLine" type="ByLineType"/>
    <xs:element name="Author" type="AuthorType"/>
    <xs:element name="Property" type="PropertyType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="HeadLine" type="HeadLineType"/>
<xs:simpleType name="HeadLineType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:element name="SubHeadLine" type="SubHeadLineType"/>
<xs:simpleType name="SubHeadLineType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:element name="ByLine" type="ByLineType"/>
<xs:simpleType name="ByLineType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:element name="Author" type="AuthorType"/>
<xs:simpleType name="AuthorType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:element name="DataContent" type="DataContentType"/>
<xs:complexType name="DataContentType">
  <xs:sequence>
    <xs:element name="body" type="bodyType" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="Photo" type="PhotoType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="body" type="bodyType"/>
<xs:complexType name="bodyType" mixed="true">
  <xs:sequence>
    <xs:element name="info" type="infoType" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Photo" type="PhotoType"/>
<xs:complexType name="PhotoType">
  <xs:sequence>
    <xs:element name="Caption" type="CaptionType" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="Agency" type="AgencyType" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="Photographer" type="PhotographerType"
minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Caption" type="CaptionType"/>
<xs:simpleType name="CaptionType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>
<xs:element name="info" type="infoType"/>
<xs:complexType name="infoType">
  <xs:sequence>
    <xs:element name="strong" type="strongType"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="strong" type="strongType"/>
<xs:simpleType name="strongType">

```

```

    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:element name="Agency" type="AgencyType"/>
  <xs:simpleType name="AgencyType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:element name="Photographer" type="PhotographerType"/>
  <xs:simpleType name="PhotographerType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>

```

18.3. XML Sample

```

<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Created by NLADB, Version 1.50, Build 043 (17/01/2005 14:46:04)
-->
- <!--
<!DOCTYPE NLAML SYSTEM ".\NLAML_1.2.dtd">
-->
<NLAML>
= <SourceMetaData>
  <Property FormalName="Publication_Name" Value="The Guardian" />
  <Property FormalName="Publication_Acronym" Value="GDN" />
  <Property FormalName="Publication_SubSource" Value="G2" />
  <Property FormalName="Edition" Value="1" />
  <Property FormalName="Re.g.ion" Value="" />
  <Property FormalName="Publication_Date" Value="20041225" />
  <Property FormalName="Day" Value="Monday" />
  <Property FormalName="Page_Section" Value="" />
  <Property FormalName="Page_Numbers" Value="026,027" />
  <Property FormalName="Origin_left" Value="1,1" />
  <Property FormalName="Origin_top" Value="17,17" />
  <Property FormalName="Width" Value="341,111" />
  <Property FormalName="Height" Value="84,26" />
</SourceMetaData>
= <Article>
  = <DescriptiveMetadata>
    = <HeadLine>
      <strong>Doubt cast on PM's 'nuclear threat' claim</strong>
    </HeadLine>
    = <SubHeadLine>
      <strong>Atomic agency</strong>Evidence inconclusive, say
sources
    </SubHeadLine>
    = <ByLine>
      <strong>Richard Norton-Taylor</strong> in London
    </ByLine>
    <Author>Richard Norton-Taylor</Author>

```

```

    <Property FormalName="Wordcount" Value="635" />
    <Property FormalName="Rights_Status" Value="0" />
    <Property FormalName="NLAArticleID" Value="450" />
  </DescriptiveMetadata>
  = <DataContent>
    = <body>
      <body.head />
      = <body.content>
        = <block lang="en">
          <p>The International Atomic Energy Agency has no evidence that Iraq is developing nuclear weapons at a former site previously destroyed by UN inspectors, despite claims made over the weekend by Tony Blair, western diplomatic sources told the Guardian yesterday.</p>
          <p>After his talks on Saturday with President Bush at Camp David, Mr Blair referred to the "real" threat of Saddam Hussein's nuclear programme. He said: "We only need to look at the report from the International Atomic Energy Agency ... showing what has been going on at the former nuclear weapons site to realise that." </p><info>Alan Frank's movie verdict - See Page 48 </info>
        </block>
      </body.content>
    </body>
  = <Photo>
    <Caption>Richard Reeves . . . 'It will take an awful lot of money to make you a little bit happier. It's very inefficient'
    </strong></Caption>
    <Photographer>Green Matthew Illustration:</Photographer>
  </Photo>
  = <Photo>
    <Photographer>Argles Martin Photograph:</Photographer>
  </Photo>
</DataContent>
</Article>
</NLAML>

```

19. Appendix three: revisions from the consultation specification documents

19.1. General changes

This document now includes sections for:

- The security and access models
- Quality and accuracy levels
- Branding support
- Style sheet support
- Character set support
- The NLA Digital Database search interface
- The acceptable usage policy

19.2. Changes to the API and XML specifications

- 19.2.1. The parameter “FormatType” in SearchDatabase has now been replaced with “Format” for consistency
- 19.2.2. The API calls now document error codes
- 19.2.3. The SearchDatabase API call now supports “NOT”, “OR” and “AND” as Boolean operators
- 19.2.4. Defined the XML metadata storage limit as 7 days only
- 19.2.5. The GetObject call now requires a PCAID parameter if branding is used
- 19.2.6. Added SearchArchive API call definition

19.3. Changes to the content specification

- 19.3.1. Cartoons are now included as elements that the NLA Digital Database will contain, recognising their importance
- 19.3.2. Horoscopes are now included, where they mention individuals’ birthdays
- 19.3.3. All unidentified gossip column stories will be attributed to all the columnists associated with that column

- 19.3.4. Gossip “snippets” will be clipped as part of a larger gossip column area rather than separately
- 19.3.5. “Best Buys” and case studies are now explicitly defined in the content specification
- 19.3.6. TV channel highlights are now supported
- 19.3.7. Clippings from the Financial Times will now be presented without the pink background in order to improving printing/faxing of clips
- 19.3.8. Clipped articles will now be reduced in size by up to 15% in order to fit onto a single A4 page
- 19.3.9. Some clipped articles that are too large, even when reduced to 85%, will be re-formatted to fit

20. Appendix four: revisions from the beta specification

- 20.1.1. The userid is no longer a mandatory field for most API calls
- 20.1.2. The ManageArchive and ListArchive API calls have been defined
- 20.1.3. The %clipreduction parameter for branding has been replaced by the %scale parameter
- 20.1.4. Added Body and Info tags to the XML feed description
- 20.1.5. Company share price pages will not be included
- 20.1.6. XML tags corrected (lower case to Upper Case)
- 20.1.7. Authentication model updated - PCA authentication no longer supported
- 20.1.8. NLA XML schema added.
- 20.1.9. Updated MyArchive details incorporating User level archive to Org level changes.
- 20.1.10. AddUser API call documented
- 20.1.11. Newspaper codes (TLA – Three Letter Acronym) updated
- 20.1.12. %scale parameter dropped from 'Branding' parameters
- 20.1.13. MyArchive service changed from 'user' to 'org' level
- 20.1.14. BlackBerry service details added
- 20.1.15. Search interface changed