Z39.88-2004: The OpenURL Framework for Context-Sensitive Services

The Key/Encoded-Value (KEV) Format

Implementation Guidelines

dc:title	KEV Implementation Guidelines
dc:creator	NISO AX Committee
dc:contributor	Ann Apps, MIMAS, The University of Manchester, UK. ann.apps@manchester.ac.uk
dc:description	Implementation guidelines for Key/Encoded-Value OpenURL Framework ContextObjects (previously SAP1).
dc:created	2003-04-16
dc:modified	2004-12-09
version	1.7

Contents

1. Introduction

The OpenURL Framework for Context-Sensitive Services Standard [1] provides a means of describing a referenced resource along with a description of the context of the reference. Additionally it defines methods of transporting these descriptions between networked systems. It is anticipated that it will be used to request services pertaining to the referenced resource and appropriate for the requester.

The OpenURL Framework is very general and has the potential to be used in many application domains and by many communities. Concrete instantiations of the various core components within the framework are defined within the *OpenURL Registry* [2]. The *Registry* will be managed by a Maintenance Agency, as defined in Appendix A of the Standard. During the ballot and public review period of the Standard, the content of the *Registry* will be static and has been pre-defined by the NISO AX Committee. There is also an experimental registry where components under development are held. In the future it will be possible to register further items.

There are currently two formats for *ContextObject Representations* defined in the OpenURL Framework, *Key/Encoded-Value* and XML. This document provides implementation guidelines for the *Key/Encoded-Value Format*, concentrating mainly, but not exclusively, on components from the San Antonio Level 1 *Community Profile* (SAP1).

2. Purpose and Scope

Within the scholarly information community the major application of the OpenURL Framework is to enable context-sensitive linking from a reference in a scholarly information system to resources relevant to the referenced item. Prior to the publication of the OpenURL Framework Standard, applications have been based on the draft OpenURL (now know as version 0.1) [3]. The traditional use of the OpenURL Framework is when a user clicks a link or button in an HTML page, typically within an 'abstracting and indexing' database application or the reference list of an electronic journal article. The result of activating this link is to transport a description of a scholarly resource, such as a journal article, to a linking server along with information about the dynamic context of the reference. Passing the dynamic context enables the provision of a list of relevant services appropriate for the user. The *Transport* is based on HTTP(S) GET or POST and is referred to as "an *OpenURL*". Using *OpenURL* version 0.1, the information, or "payload" of the *OpenURL*, is transported inline as the "query string" of a URL. The *Key/Encoded-Value Format* provides a similar means of transporting a description of a reference and its context. The XML Format allows XML descriptions to be transported.

This document provides implementation guidelines for the *Key/Encoded-Value Format*. It is not an integral part of the Standard and is for information only. These guidelines will evolve over time as implementation experience is gained. Although it is intended that the OpenURL Framework Standard and the pre-defined content of the *Registry* will remain static, these guidelines may be updated when deemed appropriate. But note that there were some changes to the *Registry* during the 'draft standard for trial use' period.

This document follows the notational convention used in the Standard in that terms defined in the Glossary of the Standard are shown in *italics font*.

2.1 Community Profiles

A selection from the *Registry* of a consistent core set of components appropriate to a particular application domain is a *Community Profile*. The definitions of *Community Profiles* are also included in the *Registry*. The San Antonio *Community Profiles,* Level 1 (KEV) and Level 2 (XML), provide support for the scholarly information community, the community in which the OpenURL Framework originated. The San Antonio Profiles are defined in Appendices C and D of the Standard. They have machine readable definitions and identifiers in the *Registry*. There is also a Simple Dublin Core *Community Profile* under development.

3. The ContextObject, its Entities and their Descriptors

The description of a referenced resource, and the descriptions of the associated resources that comprise the context of the reference, bundled together are called a *ContextObject*. It is a *Representation* of a *ContextObject* that is transported when a user makes a request by clicking a link. A *KEV OpenURL* may contain only one *ContextObject*.

The *ContextObject* may contain up to six *Entities*. One of these, the *Referent*, conveys information about the referenced item. It must always be included in a *ContextObject*. The other five entities – *ReferringEntity*, *Requester*, *Resolver*, *ServiceType* and *Referrer* – hold information about the context of the reference and are optional.

The following scenario is used for illustration (this is the same scenario used in Section 5, Part 1, of the Standard):

> Jane Doe, a University student at Caltech, reads the following article in the Elsevier ScienceDirect® collection:

McArthur, James G. et al. 2001. "p27-p16 Chimera: A Superior Antiproliferative for the Prevention of Neointimal Hyperplasia." Molecular Therapy. 3(1) 8-13. <doi:10.1006/mthe.2000.0239>

In the reference list of that article she finds a reference to the following article and would like to view its full text:

Bergelson, J. 1997. "Isolation of a common receptor for coxsackie B viruses and adenoviruses 2 and 5." Science. (275) 1320-1323. <doi:10.1126/science.275.5304.1320> <pmid:9036860>

The *Entities* of a *ContextObject* and the constraints on their maximum (Max) and minimum (Min) occurrence are given in Table 3.1. Although the OpenURL Framework allows multiple Resolvers and ServiceTypes their maximum is constrained to one for current KEV community profiles.

Table 3.1 ContextObject Entities

Entity	Definition	Min	Max	Example
Referent	The *Entity* about which the *ContextObject* was created – a referenced resource	1	1	The article by Bergelson
ReferringEntity	The *Entity* that references the *Referent*	0	1	The article by McArthur
Requester	The *Entity* that requests services pertaining to the *Referent*	0	1	Jane Doe
ServiceType	The *Entity* that defines the type of service requested	0	1	Jane requests the full text of the Bergelson article
Resolver	The *Entity* at which a request for services is targeted	0	1	The Caltech linking server
Referrer	The *Entity* that generated the *ContextObject*	0	1	Elsevier's ScienceDirect ®

Information about an *Entity* is detailed by a *Descriptor*. There are four types of *Descriptor*, defined in Section 5.2, Part 1 of the Standard.

- *Identifier*. For example: A Digital Object identifier (info:doi/10.1126/science.275.5304.1320); a PubMed identifier (info:pmid/9036860); an email address (mailto:jane.doe@caltech.edu); a http URI (http://links.caltech.edu/menu)
- *By-Value Metadata*. (Examples are given in Section 5 below)
- *By-Reference Metadata*. (Examples are given in Section 5 below)
- *Private Data*.

In *KEV*, an *Entity* may be detailed by multiple *Identifier Descriptors*. But it may have at most one each of *By-Value Metadata*, *By-Reference Metadata* and *Private Data Descriptors*.

4 Entity Description

Each of the six *Entities* of a *ContextObject* may be detailed by: several *Identifier Descriptors*; a *By-Value Metadata Descriptor*, a *By-Reference Metadata Descriptor*, a *Private Data Descriptor*. When a *ContextObject* contains more than one *Descriptor* for an *Entity* all the *Descriptors* must refer to the same *Entity*. Guidelines are given below for the description of each *Entity*.

Keys for *ContextObject* component *Entities* are defined by the Z39.88-2004 Matrix in the *Registry* at http://www.openurl.info/registry/docs/info:ofi/fmt:kev:mtx:ctx . The Keys available for use in *KEV* for each *Entity* are listed in the following subsections.

The values available for *Identifier Descriptors* are the SAP1 selection from the *Registry*, listed in Section 5.

Private Data Descriptors are not defined by the OpenURL Framework Standard. Private data is specific to the providing *Referrer*, and thus its comprehension requires a prior understanding between a *Referrer* and a *Resolver*. Implementers should be aware that the use of *Private Data Descriptors* could compromise more general interoperability.

By-Value Metadata and *By-Reference Metadata* descriptions are both based on similar metadata descriptions of an *Entity* using metadata keys from the *Metadata Formats* described in Section 5.4. The difference is that for a *By-Value Metadata* description the metadata is contained within the *ContextObject*, whereas for a *By-Reference Metadata* description the metadata is held elsewhere at a network location identified in the *ContextObject*. Also *By-Value Metadata* keys are prefixed by an indicator of the *Entity* to which they refer. In the guidelines below only *By-Value Metadata* is described, but with the above provisos the guidelines also apply to *By-Reference Metadata*. Note that an Entity may be described by only one KEV *Metadata Format*, thus it is not possible to mix *Metadata Formats*, for example using Keys from both 'journal' metadata and Simple Dublin Core metadata is illegal.

4.1 Referent

Every *ContextObject* must have a *Referent*, the referenced resource for which the *ContextObject* is created. Within the scholarly information community the *Referent* will probably be a document-like object, for instance: a book or part of a book; a journal publication or part of a journal; a report; etc.

The Keys listed in Table 4.1 may be used when describing a *Referent*.

Table 4.1 KEV Referent Keys

Key	**Description**
rft_id	*Referent Identifier*
rft_val_fmt	*By-Value Metadata Format* (info:ofi/fmt:kev:mtx:*)
rft_ref_fmt	*By-Reference Metadata Format* (info:ofi/fmt:kev:mtx:*) [**rft_ref** also required]
rft_ref	*By-Reference Metadata* Location [**rft_ref_fmt** also required]
rft_dat	*Private Data*
rft.	Metadata Key Prefix

A *Referent* may be specified using *Identifier Descriptors*. *Identifiers* from most of the *Namespaces* listed in Table 5.4 would be suitable for detailing a scholarly information *Referent* depending on the particular type of the resource: Astrophysics Bibcode; Digital Object Identifier; ISBN; ISSN; OAI identifier; PubMed identifier; SICI. From the *Namespace* listed in Table 5.3, a National Bibliographic Number would be a suitable *Identifier* for a scholarly resource. A *Referent* may be detailed by multiple *Identifiers*, in which case all the *Identifiers* must identify the same resource. Example 5.3 shows a *Referent* detailed by *Identifier Descriptors*.

A *Referent* may be specified using a *Metadata Descriptor*. Any of the *Metadata Formats* listed in Section 5.4.1-4 would be suitable for describing a scholarly information *Referent* depending on the particular type of the resource: Book; Dissertation; Journal; Patent. If a *Referent* is described by both metadata and *Identifiers*, these must refer to the same resource. Examples 5.5 (a journal article) and 5.6 (a book) show a *Referent* described by a *Metadata Descriptor*.

4.2 ReferringEntity

The *ReferringEntity* is the *Entity* that references the *Referent*. It is optional in the *ContextObject*. Within the scholarly information community the *ReferringEntity* could be a journal article that cites the *Referent*. Or it could be a record within an 'abstracting and indexing' database.

The Keys listed in Table 4.2 may be used when describing a *ReferringEntity*.

Table 4.2 KEV ReferringEntity Keys

Key	Description
rfe_id	*ReferringEntity Identifier*
rfe_val_fmt	*By-Value Metadata Format* (info:ofi/fmt:kev:mtx:*)
rfe_ref_fmt	*By-Reference Metadata Format* (info:ofi/fmt:kev:mtx:*) [**rfe_ref** also required]
rfe_ref	*By-Reference Metadata* Location [**rfe_ref_fmt** also required]
rfe_dat	*Private Data*
rfe.	Metadata Key Prefix

A *ReferringEntity* may be specified using *Identifier Descriptors*. *Identifiers* from most of the *Namespaces* listed in Table 5.4 would be suitable for detailing a scholarly information *ReferringEntity*, as described above for a *Referent*. A *ReferringEntity* may be detailed by multiple *Identifiers*, in which case all the *Identifiers* must identify the same resource. Example 10.3.2 shows a *ReferringEntity* detailed by *Identifier Descriptors*.

A *ReferringEntity* may be specified using a *Metadata Descriptor*. Any of the *Metadata Formats* listed in Section 5.4.1-4 would be suitable for describing a scholarly information *ReferringEntity* depending on the particular type of the resource: Book; Dissertation; Journal; Patent. If a *ReferringEntity* is described by both metadata and *Identifiers*, these must refer to the same resource. Example 5.6 (a journal article) and 10.8.2 (a book) show a *ReferringEntity* described by a *Metadata Descriptor*.

When a *ReferringEntity* is a record in an ‘abstracting and indexing’ database it is better to describe it by the record *Identifier* rather than by metadata, because the metadata for the *ReferringEntity* will be the same as that for the *Referent*. Examples 10.3.2 and 10.4.2 show *ReferringEntities* that are described by public and private information service record data respectively. In some cases a *ReferringEntity* may be a Web page. For instance a researcher may cite published articles on a Home page. A *ReferringEntity* that is a Web page can be described by an *Identifier* that is the URL of the Web page, as in Example 10.6.2.

4.3 Requester

The *Requester* is the *Entity* that requests services pertaining to the *Referent*. It is optional in the *ContextObject*. Within the scholarly information community the *Requester* is generally a human end-user who clicks a link within a digital library application.

The Keys listed in Table 4.3 may be used when describing a *Requester*. The *KEV ContextObject* definition also includes Keys for metadata description of a *Requester*, but the *Registry* does not currently include any appropriate *Metadata Formats*.

Table 4.3 KEV Requester Keys

Key	Description
req_id	*Requester Identifier*
req_dat	*Private Data*

A *Requester* may be specified using *Identifier Descriptors*. *Identifiers* from the *Namespaces* listed in Table 5.3 that would be suitable for detailing a *Requester* are: LDAP; mailto. A *Requester* may be detailed by multiple *Identifiers*, in which case all the *Identifiers* must identify the same resource. Example 5.2 shows a *Requester* detailed by an *Identifier Descriptor*.

Some applications may wish to pass *Requester* authentication related information. This may be the authentication system used. Or it could be the location information such as the institution or company to which the *Requester* belongs. Or it could be an *Identifier* for the *Requester* that would indicate their preference profile within some system. At present this information would be described using *Private Data* and will depend on prior understanding by both *Referrer* and *Resolver*. It would be inadvisable to include authentication details such as passwords within a *Requester Descriptor*. As well as concerns about the security of the

OpenURL Transport, including such detailed authentication data would probably contravene the terms and conditions of the authentication system in use.

4.4 ServiceType

The *ServiceType* is the *Entity* that defines the type of service requested. It is optional in the *ContextObject*. Within the scholarly information community the *ServiceType* could be a request for; the full text of an article; the abstract of an article; an inter-library loan request, etc.

The Keys listed in Table 4.4 may be used when describing a *ServiceType*. The *KEV ContextObject* definition also includes a Key for an *Identifier* description of a *ServiceType*, but the *Registry* does not currently include any appropriate *Identifiers*.

Table 4.4 KEV ServiceType Keys

Key	**Description**
svc_val_fmt	*By-Value Metadata Format* (info:ofi/fmt:kev:mtx:*)
svc_ref_fmt	*By-Reference Metadata Format* (info:ofi/fmt:kev:mtx:*) [**svc_ref** also required]
svc_ref	*By-Reference Metadata* Location [**svc_ref_fmt** also required]
svc_dat	*Private Data*
svc.	Metadata Key Prefix

A *ServiceType* may be specified using a *Metadata Descriptor*. The *Metadata Formats* described in Section 5.4.5 would be suitable for describing a scholarly information *ServiceType*. Example 10.2.4 shows a *ServiceType* detailed by a *Metadata Descriptor*.

4.5 Resolver

The *Resolver* is the *Entity* at which a request for services is targeted. It is optional in the *ContextObject*. This need not be the same *Resolver* as that specified as the base URL for an *OpenURL Transport* and does not replace that base URL.

The Keys listed in Table 4.6 may be used when describing a *Resolver*. The *KEV ContextObject* definition also includes Keys for metadata description of a *Resolver*, but the *Registry* does not currently include any appropriate *Metadata Formats*.

Table 4.6 KEV Resolver Keys

Key	**Description**
res_id	*Resolver Identifier*
res_dat	*Private Data*

A *Resolver* may be specified using *Identifier Descriptors*. An *Identifier* from the *Namespaces* listed in Table 5.3 that would be suitable for detailing a *Resolver* is http. A *Resolver* may be detailed by multiple *Identifiers*, in which case all the *Identifiers* must identify the same resource.

4.6 Referrer

The *Referrer* is the *Entity* that generated the *ContextObject*. It is optional in the *ContextObject*, but its inclusion is strongly encouraged. Within the scholarly information community the *Referrer* will be an information provider such as an electronic journal application or an ‘abstracting and indexing’ service.

The Keys listed in Table 4.7 may be used when describing a *Referrer*. The *KEV ContextObject* definition also includes Keys for metadata description of a *Referrer*, but the *Registry* does not currently include any appropriate *Metadata Formats*.

Table 4.7 KEV Referrer Keys

Key	**Description**

rfr_id	*Referrer Identifier*
rfr_dat	*Private Data*

Referrer Identifiers are defined in the source identifier *Namespace* ‘info:ofi/nam:info:sid:’. They are identified using the ‘info:sid/’ scheme for the identification of collections of information assets defined in the *Registry* at http://openurl.info/registry/docs/pdf/info-sid.pdf. Such a collection could be an organization, a website, a publisher, or a database. Within this scheme a *Referrer Identifier* consists of a DNS identifier (Internet domain name, sub-domain or host name) for the *Referrer*’s collection, followed by an optional name providing further details about the nature of the collection, for example a particular service or database. Example *Referrer Identifiers* are: ‘info:sid/firstsearch.oclc.org:inspec’; ‘info:sid/wiley.com’.

It is strongly recommended that genuine *Referrer Identifiers* are included in *ContextObjects*. In particular a genuine *Referrer Identifier* is necessary if *Referrer*-specific *Private Data* is included in a *ContextObject*, in order to define the provenance of that private data. All the examples in Section 10 show *Referrers* detailed by *Identifier Descriptors*. *Resolvers* make use of *Referrer Identifiers* to avoid circular linking and to compile usage statistics.

[Note that the *Referrer* scheme changed from in version 1.4 of this document from version 1.3. The ‘ofi/rfr:db’ scheme is no longer available.]

5. Registry Selections

The following selection of KEV OpenURL Framework components from the *Registry* would be appropriate within the scholarly information community.

5.1 ContextObject Format

ContextObjects are represented using the *Key/Encoded-Value (KEV) Format*, as a string of ampersand-delimited pairs. Each pair consists of a label (Key) and an associated Value that is URL-encoded, separated by an equals (‘=’). All values of *KEV* pairs must be URL-encoded so that the *ContextObject Representation* is ‘transport ready’. The *KEV Format* is defined in Part 2 of the Standard. URL-encoding is explained below in Appendix B.3. The *Registry Identifier* of the *KEV Format* is ‘info:ofi/fmt:kev’.

Keys for a KEV *ContextObject* are defined by the Z39.88-2004 MTX *Constraint Definition*. The *Registry Identifier* of the MTX *Constraint Definition* is ‘info:ofi/fmt:kev:mtx’ and the *Registry Identifier* of the *KEV ContextObject Format* is ‘info:ofi/fmt:kev:mtx:ctx’.

As well as defining Keys used to describe the component entities of a *ContextObject* (see Section 4 above), the *KEV ContextObject* also includes optional administrative keys for the *ContextObject*.

Table 5.1 KEV ContextObject Administrative Keys

Key	**Description**
ctx_ver	Version of the *ContextObject*. Fixed value: ‘Z39.88-2004’ (‘Z’ must be uppercase)
ctx_enc	*Character Encoding* used (see Section 5.2 below)
ctx_id	*Identifier* for the *ContextObject*
ctx_tim	Timestamp for the *ContextObject*, a date and time to the seconds level of the W3CDTF profile of ISO 8601, of the form ‘YYYY-MM-DDThh:mm:ssTZD’ or ‘YYYY-MM-DD’

5.2 Character Encoding

The following *Character Encodings* from the IANA character sets (http://www.iana.org/assignments/character-sets) to be used for Values within a KEV *ContextObject* are included in the initial *Registry*, where further description of these character sets may be found.

Table 5.2 KEV Character Encodings

Encoding	**Registry Identifier**

UTF-8 Unicode	info:ofi/enc:UTF-8
ISO Latin 1	info:ofi/enc:ISO-8859-1

The *Character Encoding* used within a *ContextObject* is specified using the **ctx_enc** Key. The default *Character Encoding* is UTF-8. Thus if UTF-8 is used the **ctx_enc** Key may be omitted. For *By-Reference Metadata* UTF-8 *Character Encoding* is the default, but it can declare its own encoding.

Example 5.1 Character Encoding

Specification of ISO Latin 1 *Character Encoding* **&ctx_enc=info%3Aofi%2Fenc%3AISO-8859-1**

When forming an OpenURL encoded using Unicode UTF-8, best practice is to use Unicode Normalization Form C. The practice of reducing Unicode characters to ASCII is discouraged.

Other encoding schemes may be used in the future, provided they are registered, but they will not be included in the scholarly information profile SAP1.

5.3 Namespaces

The following sub-sections list the *Identifier Namespaces* appropriate for the scholarly information community. *Namespaces* have *Registry Identifiers* within the 'info:ofi/nam:' *Namespace*. But for an *Identifier Descriptor* just the URI is used. For example, the Registry Identifier for 'mailto' is 'info:ofi/nam:mailto:' but an *Identifier Descriptor* would be 'mailto:jane.doe@caltech.edu' as shown in Example 5.2.

[Note that this section changed significantly from versions 1.1 to 1.2 of this document. All *Namespaces* now follow URI schemes and the 'uri:' prefix has been dropped. The previous ORI *Namespaces* are now URI 'info:' *Namespaces*. The private XRI *Namespace* is no longer available and a *Referrer*-specific identifier should be encoded within a *Private Data Descriptor*.]

5.3.1 Namespaces

Table 5.3 lists the *Namespaces* suitable for scholarly information from URI schemes other than the 'info:' scheme. *Identifiers* from these *Namespaces* may be used as *Identifier Descriptors*. They may also be used to specify network locations for *By-Reference Metadata* descriptions. The following example, 5.2, shows a *Requester* detailed by a mailto *Identifier*.

Example 5.2 Requester URI Namespace Identifier Descriptor

&req_id=mailto%3Ajane.doe%40caltech.edu

Table 5.3 URI Namespaces

URI Namespace	Registry Identifier	URI
FTP	info:ofi/nam:ftp:	ftp:
HTTP	info:ofi/nam:http:	http:
LDAP	info:ofi/nam:ldap:	ldap:
mailto	info:ofi/nam:mailto:	mailto:
ISBN	info:ofi/nam:urn:ISBN:	urn:ISBN:
ISSN	info:ofi/nam:urn:ISSN:	urn:ISSN
National Bibliographic Number (NBN)	info:ofi/nam:urn:NBN	urn:NBN:

5.3.2 'info' Namespaces

Table 5.4 lists the *Namespaces* suitable for scholarly information from the 'info' URI scheme. *Identifiers* from these *Namespaces* may be used as *Identifier Descriptors*. The following example, 5.3, shows a *Referent* detailed by both a PubMed *Identifier* and a Digital Object *Identifier*.

Example 5.3 Referent 'info' Namespace Identifier Descriptors

(Not URL-encoded and with line breaks for readability) **&rft_id=info:pmid/9036860** **&rft_id=info:doi/10.1126/science.275.5304.1320**
(URL-encoded) **&rft_id=info%3Apmid%2F9036860&rft_id=info%3Adoi%2F10.1126%2Fscience.275.5304.1320**

Table 5.4 'info' Namespaces

Namespace	**Registry Identifier**	**URI**
Astrophysics Bibcode	info:ofi/nam:info:bibcode:	info:bibcode/
Digital Object Identifier	info:ofi/nam:info:doi:	info:doi/
CNRI Handle	info:ofi/nam:info:hdl:	info:hdl/
LCCN	Info:ofi/nam:info:lccn:	info:lccn/
Open Archives Initiative (OAI)	info:ofi/nam:info:oai:	info:oai/
OCLC WorldCat	info:ofi/nam:info:oclcnum:	info:oclcnum/
PubMed	info:ofi/nam:info:pmid:	info:pmid/
SICI	info:ofi/nam:info:sici:	info:sici/
Source *Identifier*	info:ofi/nam:info:sid:	info:sid/

5.4 Metadata Formats

Entities within a *ContextObject* may be described by metadata. These metadata descriptions may be contained within the *ContextObject* itself, known as *By-Value Metadata*. Alternatively the metadata descriptions may be held elsewhere, in which case, known as *By-Reference Metadata*, the network location of the metadata is contained in the *ContextObject*. When an *Entity* is described by *KEV By-Value Metadata* the metadata keys must have a metadata prefix indicating the *Entity*. This metadata key prefix should not be used for *KEV By-Reference Metadata*.

The initial KEV Metadata Formats are listed in Table 5.5. These *Metadata Formats* provide a means to describe resources within the scholarly information community. They are described in more detail in the following sub-sections.

Table 5.5 Key/Encoded-Value Metadata Formats

Metadata Format	**Registry Identifier**
Book and Book component	info:ofi/fmt:kev:mtx:book
Dissertation	info:ofi/fmt:kev:mtx:dissertation
Journal and Journal component	info:ofi/fmt:kev:mtx:journal
Patent	info:ofi/fmt:kev:mtx:patent
Scholarly Service Type	info:ofi/fmt:kev:mtx:sch_svc

5.4.1 Journal Metadata

The *Metadata Format* to represent a journal publication is defined by a Z39.88-2004 Matrix in the *Registry* at: http://www.openurl.info/registry/docs/info:ofi/fmt:kev:mtx:journal. This Matrix includes a description for the use of each of the metadata items. That information is not reproduced here but guidelines for a few of the metadata items are given below. The Matrix defines the permissible occurrences of the metadata keys.

When a *Referent* (or *ReferringEntity*) is described with *By-Value Metadata* using the journal *Metadata Format*:

- The **rft_val_fmt** (or **rfe_val_fmt**) Key in the *ContextObject* has the value: 'info:ofi/fmt:kev:mtx:journal'
- The metadata keys must be prefixed with '**rft.**' (or '**rfe.**')

When a *Referent* (or *ReferringEntity*) is described with *By-Reference Metadata* using the journal *Metadata Format*:

- The **rft_ref_fmt** (or **rfe_ref_fmt**) Key in the *ContextObject* has the value: 'info:ofi/fmt:kev:mtx:journal'

- The metadata at the location specified by the **rft_ref** (or **rfe_ref**) Key is in KEV journal *Metadata Format*
- The metadata keys must not be prefixed
- *Character Encoding* default is UTF-8, but it can declare its own encoding

Genre. The journal *Metadata Format* is a general purpose *Format* to describe all levels within a journal or serial publication. It may also be used to describe a conference proceedings or paper where these are published in a journal, or a serial publication preprint. The specific genre of the resource within the journal publication type may be indicated by the **genre** metadata key. The possible values for the genre metadata key are shown in Table 5.6. It would be expected that the granularity of metadata provided would be consistent with the specified genre. For example a resource described with a genre 'issue' would have appropriate **volume** / **part** / **issue** metadata.

Table 5.6 Journal Metadata Genre Values

Genre Value	Description
journal	Serial publication issued in several parts
issue	One instance of a serial publication
article	Document published in a journal
conference	Record of a conference published in a journal
proceeding	Single conference presentation published in a journal
preprint	Single paper or report published prior to its publication in a journal
unknown	Unknown genre

Author. The Journal *Metadata Format* provides several keys to detail the author of an article or paper. For the purpose of matching citations it is generally the first author's name that is significant. The first author should be specified using **aulast** to indicate their family name and one or more of **aufirst, auinit, aunit1, auinitm** for their given name or initials. An **ausuffix** key may also be used if a name suffix is significant. If the main creator of a document is an organization the **aucorp** key should be used instead. A general **au** key is also provided. This may be used to contain the full name of an author where splitting the author's name is not possible. But it is recommended that **aulast**, with separate family name or initials, be used to specify the first author rather than ***au*** whenever possible. The **au** key is repeatable so may be used to list the co-authors of an article.

The following examples show the use of the Journal *Metadata Format*. Example 5.5 shows *By-Value Metadata* for a *Referent*. Example 5.6 shows *By-Reference Metadata* for a *ReferringEntity*.

Example 5.5 By-Value Journal Metadata for a Referent

(Not URL-encoded and with line breaks for readability)

```
&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&rft.genre=article
&rft.atitle=Isolation of a common receptor for coxsackie B
&rft.jtitle=Science
&rft.aulast=Bergelson
&rft.auinit=J
&rft.date=1997
&rft.volume=275
&rft.spage=1320
&rft.epage=1323
```

(URL-encoded *KEV Format*)

```
&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rft.genre=article&rf
t.atitle=Isolation+of+a+common+receptor+for+coxsackie+B&rft.jtitle=Science&rft
.aulast=Bergelson&rft.auinit=J&rft.date=1997&rft.volume=275&rft.spage=1320&rf
t.epage=1323
```

Example 5.6 By-Reference Metadata for a ReferringEntity

Within the *ContextObject*:
(Not URL-encoded and with line breaks for readability)

&rfe_ref_fmt=info:ofi/fmt:kev:mtx:journal
&rfe_ref=http://www.example.org/temp/1234.txt

At http://www.example.org/temp/1234.txt :
(Not URL-encoded and with line breaks for readability)

&genre=article
&atitle=p27-p16 Chimera: A Superior Antiproliferative
&jtitle=Molecular Theory
&aulast=McArthur
&aufirst=James
&date=2001
&volume=3
&issue=1
&spage=8
&epage=13

Within the *ContextObject*:
(URL-encoded *KEV Format*)

&rfe_ref_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rfe_ref=http%3A%2F%2Fwww.example.org%2Ftemp%2F1234.txt

At http://www.example.org/temp/1234.txt :
(URL-encoded *KEV Format*)

&genre=article&atitle=p27-p16+Chimera%3A+A+superior+Antiproliferative&jtitle=Molecular+Theory&aulast=McArthur&aufirst=James&date=2001&volume=3&issue=1&spage=8&epage=13

5.4.2 Book and General Document Metadata

The *Metadata Format* to represent a book, book component, report or general document is defined by a Z39.88-2004 Matrix in the *Registry* at: http://www.openurl.info/registry/docs/info:ofi/fmt:kev:mtx:book. This Matrix includes a description for the use of each of the metadata items. That information is not reproduced here but guidelines for a few of the metadata items are given below. The Matrix defines the permissible occurrences of the metadata keys.

When a *Referent* (or *ReferringEntity*) is described with *By-Value Metadata* using the book *Metadata Format*:

- The **rft_val_fmt** (or **rfe_val_fmt**) Key in the *ContextObject* has the value: 'info:ofi/fmt:kev:mtx:book'
- The metadata keys must be prefixed with '**rft.**' (or '**rfe.**')

When a *Referent* (or *ReferringEntity*) is described with *By-Reference Metadata* using the book *Metadata Format*:

- The **rft_ref_fmt** (or **rfe_ref_fmt**) Key in the *ContextObject* has the value: 'info:ofi/fmt:kev:mtx:book'
- The metadata at the location specified by the **rft_ref** (or **rfe_ref**) Key is in KEV book *Metadata Format*
- The metadata keys must not be prefixed
- *Character Encoding* default is UTF-8, but it can declare its own encoding

Genre. The book *Metadata Format* is a general purpose *Format* to describe books, book components, and similar documents. It may be used to describe a conference proceedings or paper, or a report. Book metadata may also be used to describe a general document where some bibliographic information, such as author and title, are known, but the specific type of the resource cannot be determined. It could possibly be used to describe non-text items as described in Section 5.4.7. The specific genre of the resource within the book metadata type may be indicated by the **genre** metadata key. The possible values for the genre metadata key are shown in Table 5.7.

Table 5.7 Book Metadata Genre Values

Genre Value	**Description**
book	Book
bookitem	Section of a book
conference	Record of a conference
proceeding	Single conference presentation
report	Published report
document	General document
unknown	Unknown genre

Author. The recommendations for detailing authors are the same as those for the Journal *Metadata Format* described in Section 5.4.1 above.

The following example, 5.7, shows By-Value Book Metadata for a *Referent*.

Example 5.7 By-Value Book Metadata for a Referent

(Not URL-encoded and with line breaks for readability)

```
&rft_val_fmt=info:ofi/fmt:kev:mtx:book
&rft.genre=book
&rft.btitle=Professional XML Meta Data
&rft.aulast=Dodds
&rft.afirst=David
&rft.date=2001
&rft.isbn=1861004516
```

(URL-encoded *KEV Format*)

```
&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Abook&rft.genre=book&rft.bt
itle=Professional+XML+Meta+Data&rft.aulast=Dodds&rft.afirst=David&rft.date=2
001&rft.isbn=1861004516
```

5.4.3 Dissertation Metadata

The *Metadata Format* to represent a dissertation is defined by a Z39.88-2004 Matrix in the *Registry* at: http://www.openurl.info/registry/docs/info:ofi/fmt:kev:mtx:dissertation. This Matrix includes a description for the use of each of the metadata items, and defines the permissible occurrences of the metadata keys.

When a *Referent* (or *ReferringEntity*) is described with *By-Value Metadata* using the dissertation *Metadata Format*:

- The **rft_val_fmt** (or **rfe_val_fmt**) Key in the *ContextObject* has the value: 'info:ofi/fmt:kev:mtx:dissertation'
- The metadata keys must be prefixed with '**rft.**' (or '**rfe.**')

When a *Referent* (or *ReferringEntity*) is described with *By-Reference Metadata* using the dissertation *Metadata Format*:

- The **rft_ref_fmt** (or **rfe_ref_fmt**) Key in the *ContextObject* has the value: 'info:ofi/fmt:kev:mtx:dissertation'
- The metadata at the location specified by the **rft_ref** (or **rfe_ref**) Key is in KEV dissertation *Metadata Format*
- The metadata keys must not be prefixed
- *Character Encoding* default is UTF-8, but it can declare its own encoding

Author. The recommendations for detailing authors are the same as those for the Journal *Metadata Format* described in Section 5.4.1 above.

The following example, 5.8, shows By-Value Dissertation Metadata for a *Referent*.

Example 5.8 By-Value Dissertation Metadata for a Referent

(Not URL-encoded and with line breaks for readability)

&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation
&rft.title=The effects of the rare earth elements yttrium, gadolinium and dysprosium
&rft.aulast=Apps
&rft.afirst=Peter
&rft.auinitm=J
&rft.date=2001
&rft.co=United Kingdom
&rft.inst=University of Manchester
&rft.degree=PhD

(URL-encoded *KEV Format*)

&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Adissertation&rft.title=The effects+of+the+rare+earth+elements+yttrium%2C+gadolinium+and+dysprosium &rft.aulast=Apps&rft.afirst=Peter&rft.auinitm=J&rft.date=2001&rft.co=United+Kin gdom&rft.inst=University+of+Manchester&rft.degree=PhD

5.4.4 Patent Metadata

The *Metadata Format* to represent a patent is defined by a Z39.88-2004 Matrix in the *Registry* at: http://www.openurl.info/registry/docs/info:ofi/fmt:kev:mtx:patent. That Matrix includes a description for the use of each of the metadata items, and defines the permissible occurrences of the metadata keys.

When a *Referent* (or *ReferringEntity*) is described with *By-Value Metadata* using the patent *Metadata Format*:

- The **rft_val_fmt** (or **rfe_val_fmt**) Key in the *ContextObject* has the value: 'info:ofi/fmt:kev:mtx:patent'
- The metadata keys must be prefixed with '**rft.**' (or '**rfe.**')

When a *Referent* (or *ReferringEntity*) is described with *By-Reference Metadata* using the patent *Metadata Format*:

- The **rft_ref_fmt** (or **rfe_ref_fmt**) Key in the *ContextObject* has the value: 'info:ofi/fmt:kev:mtx:patent'
- The metadata at the location specified by the **rft_ref** (or **rfe_ref**) Key is in KEV patent *Metadata Format*
- The metadata keys must not be prefixed
- *Character Encoding* default is UTF-8, but it can declare its own encoding

5.4.5 Scholarly Service Type Metadata

The *Metadata Format* to represent a scholarly service type is defined by a Z39.88-2004 Matrix in the *Registry* at: http://www.openurl.info/registry/docs/info:ofi/fmt:kev:mtx:sch_svc. This Matrix includes a description for the use of each of the metadata items. That information is not reproduced here but the metadata items are listed below. Possible values for each metadata item are 'yes' or 'no'. The Matrix defines the permissible occurrences of the metadata keys.

Because the OpenURL Framework does not prescribe *Resolver* behavior there is no guarantee that a *Resolver* will provide the requested service when one of these *ServiceType Metadata* Keys is included in a *ContextObject*. Requested *ServiceTypes* should be regarded as a hint to the *Resolver*. In particular it should be noted that requesting 'fulltext' will not override any subscription restrictions.

When a *ServiceType* is described with *By-Value Metadata* using the scholarly service type *Metadata Format*:

- The **svc_val_fmt** Key in the *ContextObject* has the value: 'info:ofi/fmt:kev:mtx:sch_svc'
- The metadata keys must be prefixed with '**svc.**'

When a *ServiceType* is described with *By-Reference Metadata* using the scholarly service type *Metadata Format*:

- The **svc_ref_fmt** Key in the *ContextObject* has the value: ‘info:ofi/fmt:kev:mtx:sch_svc’
- The metadata at the location specified by the **svc_ref** Key is in KEV scholarly service type *Metadata Format*
- The metadata keys must not be prefixed
- *Character Encoding* default is UTF-8, but it can declare its own encoding

Table 5.8 Scholarly Service Type Keys

Key	Description
abstract	Abstract of the *Referent*
citation	Bibliographic citation of the *Referent*
fulltext	Full text of the *Referent*
holdings	Holdings information related to the *Referent*
ill	Inter-library Loan request for the *Referent*
any	Any service type

The following example shows the use of the scholarly service type *Metadata Format*.

Example 5.9 By-Value Scholarly Metadata for a ServiceType

(Not URL-encoded and with line breaks for readability)

```
&svc_val_fmt=info:ofi/fmt:kev:mtx:sch_svc
&svc.fulltext=yes
```

(URL-encoded *KEV Format*)

```
&svc_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Asch_svc&svc.fulltext=yes
```

[Note the scholarly service type *Metadata Format* supersedes the service type identifiers detailed in previous versions (1.3 and below) of this document, which are no longer available.]

5.4.6 Dublin Core Metadata Format

An experimental metadata format for Simple Dublin Core is currently under development [4]. Its eventual Identifier in the Registry will be ‘info:ofi/fmt:kev:mtx:dc’.It provides metadata Keys for the 15 elements of Simple Dublin Core (http://www.dublincore.org) listed in Table 5.9.

Table 5.9 Simple Dublin Core Metadata Keys

Key	Description
title	Name of the resource
creator	Primary creator of the content of the resource
subject	Topic of the content of the resource
description	Description of the content of the resource
publisher	Entity making resource available
contributor	Contributor to content of resource
date	Date of an event in lifecycle of resource
type	Nature or genre of content of resource
format	Physical or digital manifestation of resource
identifier	Unambiguous reference to resource in a given context
source	Resource from which this resource was derived
language	Language of intellectual content of resource
relation	Reference to a related resource
coverage	Extent or scope of content of resource
rights	Rights held in or over resource

The following example shows a use of the simple Dublin Core *Metadata Format* for a *Referent* and a *ServiceType*.

Example 5.10 By-Value Simple Dublin Core Metadata

(Not URL-encoded and with line breaks for readability)

&rft_val_fmt=info:ofi/fmt:kev:mtx:dc
&rft.title=jstor business
&rft.subject=business
&svc_val_fmt=info:ofi/fmt:kev:mtx:dc
&svc.format=text/xml

(URL-encoded *KEV Format*)

&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Adc&rft.title=jstor+business&rft.subject=business&svc_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Adc&svc.format=text%2Fxml

This Simple Dublin Core *Metadata Format* opens the possibility of indicating the ‘subject’ of an entity. But note that all metadata keys for a particular entity must be from the same *Metadata Format*.

5.4.7 Non-text and Other Metadata Formats

At present the *Metadata Formats* described in the above subsections are the only ones in the *Registry* or under development by NISO Committee AX. But there will be requirements within particular application domains for metadata specific to them.

5.4.7.1 Non-text Metadata

There has been some discussion and early development of *Metadata Formats* for non-text scholarly information items, such as images, maps and sound. When development of these is complete they will be added to the experimental *Registry*. In the meantime there are two options for creating *ContextObjects* describing such items.

Some OpenURL implementers have created version 0.1 OpenURLs for non-text items by including the information they can such as ‘title’ and author information but omitting ‘genre’. This option can be taken forward when upgrading to OpenURL version 1.0. Best practice will be to create a *ContextObject* describing the *Referent* using ‘book’ metadata. As previously metadata values such as ‘title’ and author details may be included. The ‘genre’ can either be set to ‘unknown’ or omitted.

Example 5.11 By-Value ‘Book’ Metadata for an Image

(Not URL-encoded and with line breaks for readability)

&rft_val_fmt=info:ofi/fmt:kev:mtx:book
&rft.title=My Phtotgraph
&rft.aulast=Apps
&rft.aufirst=Ann
&rft.genre=unknown

(URL-encoded *KEV Format*)

&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Abook&rft.title=My+Photograph&rft.aulast=Apps&rft.aufirst=Ann&rft.genre=unknown

An alternative approach is to use the Simple Dublin Core *Metadata Format*. This will allow the inclusion of details such as ‘title’ and ‘creator’. It also provides keys for genre (‘type’) and ‘format’. Suggested best practice would be to take values from the DCMI Type Vocabulary (http://dublincore.org/documents/dcmi-type-vocabulary/) for ‘type’ and to use MIME types for ‘format’.

Example 5.12 By-Value Simple Dublin Core Metadata for an Image

(Not URL-encoded and with line breaks for readability)

&rft_val_fmt=info:ofi/fmt:kev:mtx:dc
&rft.title=My Phtotgraph
&rft.creator=Ann Apps
&rft.type=Image
&rft.format=application/gif

(URL-encoded *KEV Format*)

&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Adc&rft.title=My+Photograph

```
&rft.creator=Ann+Apps&rft.type=Image&rft.format=application%2Fgif
```

5.4.7.2 Other Metadata

Other KEV *Metadata Formats* may be defined for use in local applications and possibly for eventual use within communities. The name of a new *Metadata Format* ***must not*** begin with the letters ‘ctx’.

To define a KEV *Metadata Format* it is necessary to define a Z39.88-2004 Matrix as defined in Appendix B of the Standard. Four steps are involved in producing a registered *Metadata Format*. At present, until the Registration Agency has been appointed and the registration process defined, only the first and third steps are possible.

- Publish the Z39.88-2004 Matrix defining the new *Metadata Format* on a local website. Its identifier will be its URL. If this *Metadata Format* is for use within a local application or closed community, this is the only action required.
- Request that the new *Metadata Format* be added to the experimental Pre-Registry. This will make it potentially generally available. It will be assigned an identifier by the Registration Agency.
- When it is decided that the *Metadata Format* is suitable for registration it will be moved to the *Registry*, but flagged as being for ‘trial use’. This will eventually require approval from the Registration Authority, but for the initial Registry it requires vetting by NISO Committee AX.
- Following further approval by the Registration Authority it will be moved to ‘official’ status in the *Registry*. A final ‘info:ofi/fmt:kev:mtx:***’ identifier will be assigned. (Again this can be decided by the Committee before the appointment of a Registration Agency.)

5.5 Transports

KEV *ContextObject*s may be transported over the network using the *Transports* listed in Table 5.10. Further details of transporting KEV *OpenURLs* are given in Section 6.

Table 5.10 SAP1 Transports

Transport	**Registry Identifier**
By-Reference OpenURL over HTTP	info:ofi/tsp:http:openurl-by-ref
By-Value OpenURL over HTTP	info:ofi/tsp:http:openurl-by-val
Inline OpenURL over HTTP	info:ofi/tsp:http:openurl-inline

5.6 Profiles

A selection from the *Registry* of a consistent core set of components appropriate to a particular application domain is a *Community Profile*. The definitions of *Community Profiles* are also included in the *Registry*. The San Antonio *Community Profile* Level 1, provides KEV *ContextObject* support for the scholarly information community. There is also a Simple Dublin Core *Community Profile* [5] under development.

The San Antonio Profile Level 1 (SAP1) is defined in Appendix C of the Standard, and identified in the *Registry* as ‘info:ofi/pro:sap1’. SAP1 is defined in the *Registry* at http://www.openurl.info/registry/docs/info:ofi/pro:sap1 . Many of the core components of SAP1 from the *Registry* are listed in the above sections of this document.

The main purpose of *Community Profiles* is to provide a ‘benchmark’ against which a *Resolver* or OpenURL source (ie. *Referrer*) can be measured. Thus they are for supporting compliance claims or requiring conformance when advertising or purchasing Resolvers and Referrers. There is no indication within a *ContextObject* of any *Profile* conformance. There is no requirement that a particular *ContextObject* contains only components from a particular *Profile* unless its creator wishes to claim conformance to that *Profile*. Claims of conformance to a *Profile* could be verified against its definition in the *Registry*, but there is no means provided by the Standard of automatically interrogating a *Resolver* to determine which *Profiles* it supports, the OpenURL Framework not being a protocol.

Metadata Formats exist in the *Registry* independent of *Profiles*. *Profiles* subscribe to a selection of *Metadata Formats*. Thus one *Metadata Format* can be included in several profiles. *Metadata Formats* are indicated in actual *ContextObjects* for the particular Entities described by them, whereas *Profiles* are not.

6 OpenURL Transport

KEV *ContextObjects* are transported over the network using *OpenURL* as *Inline, By-Value,* or *By-Reference*. In all three cases this *Transport* uses the HTTP protocol. The *ContextObjects* may be conveyed using either GET or POST. *OpenURL Transport* is defined in Part 4 of the Standard where there are several examples of using GET and POST and of the three methods

An *OpenURL* includes a single base URL that specifies the internet host, port and path of an HTTP-based service, the *OpenURL Resolver*, that is the target of the *Transport*.

Note that HTTPS, the secure sibling of HTTP, could be supported additionally, but is not required for conformance to current *Profiles* such as SAP1.

OpenURL GET and POST
Table 6.1 lists Keys that are used in all *OpenURLs*. Further Keys used by the different methods are listed in the following subsections (6.1-6.3).

Table 6.1 KEV OpenURL Keys

Key	Min	Max	Description
url_ver	1	1	*OpenURL* version. Fixed value: Z39.88-2004 (‘Z’ must be uppercase)
url_tim	0	1	*OpenURL* creation timestamp

OpenURL GET. In HTTP GET mode the *OpenURL* Key/Value pairs, including the further Keys listed below, are concatenated by ampersand (‘&’) to form the query string of an HTTP GET request. This query string is appended to the base URL of the target *OpenURL Resolver*, separated from it by a question mark (‘?’). The query string including the values of *OpenURL* Keys must be URL-encoded. Example 10.1.6 shows the query string of an *OpenURL* using HTTP GET. Note that best practice is not to include a leading ampersand (‘&’) following the question mark (‘?’), that is there should be no ampersand (‘&’) preceding the first OpenURL Key.

OpenURL POST. In HTTP POST mode the query string, assembled as specified in the previous paragraph, is carried in the message body of the HTTP POST. The Content-Type of the HTTP request must be ‘application/x-www-form-urlencoded’. Thus the message body must be URL-encoded. Example 10.3.8 shows an *OpenURL* using HTTP POST. It should be noted that any whitespace within a POST body, possibly added by a transport agent for compatibility with legacy networks, has to be removed before processing.

6.1 Inline OpenURL

In an *Inline OpenURL* the *ContextObject* is transported as part of the query string. Table 6.2 shows the additional Keys for an *Inline OpenURL*.

Table 6.2 KEV Inline OpenURL Keys

Key	Min	Max	Description
url_ctx_fmt	0	1	*Format* of the *ContextObject*. Fixed value: info:ofi/fmt:kev:mtx:ctx

For an *Inline OpenURL* the query string is assembled by concatenating using an ampersand (‘&’) separator, the *OpenURL* Key/Value pairs of the *OpenURL* and the *ContextObject* (but with no leading ‘&’). Because the query string must be URL-encoded, the *OpenURL* Key/Value pairs must be URL-encoded. The Key/Encoded-Value pairs of the *ContextObject* are already URL-encoded. Examples 10.1.6, 10.4.7, 10.7.7 and 10.8.7 show *Inline OpenURLs*.

6.2 By-Value OpenURL

In a *By-Value OpenURL* the *ContextObject* is transported as the Value of a Key in the query string. Table 6.3 shows the additional Keys for a *By-Value OpenURL*.

Table 6.3 KEV By-Value OpenURL Keys

Key	Min	Max	Description
url_ctx_fmt	1	1	*Format* of the *ContextObject*. Fixed value: info:ofi/fmt:kev:mtx:ctx
url_ctx_val	1	1	*Representation* of the *ContextObject*

In a *By-Value OpenURL* the *ContextObject* is the Value of the **url_ctx_val** Key (with no leading '&'). Along with the other Key/Value pairs in the *OpenURL*, this Value must be URL-encoded. This will result in a double URL-encoding of the *ContextObject* within the *OpenURL*. Examples 10.3.8 and 10.6.8 show *By-Value OpenURLs*.

6.3 By-Reference OpenURL

In a *By-Reference OpenURL* the network location of a *ContextObject*, rather than the actual *Representation* of the *ContextObject*, is transported as the Value of a Key in the query string. Table 6.4 shows the additional Keys for a *By-Reference OpenURL*.

Table 6.4 KEV By-Reference OpenURL Keys

Key	Min	Max	Description
url_ctx_fmt	1	1	*Format* of the *ContextObject*. Fixed value: info:ofi/fmt:kev:mtx:ctx
url_ctx_ref	1	1	Network location of the *ContextObject*

In a *By-Reference OpenURL* the *ContextObject* is at the network location defined by the Value of the **url_ctx_ref** Key. Along with the other Key/Value pairs in the *OpenURL*, this Value must be URL-encoded. The referenced KEV *ContextObject* will by definition be URL-encoded (and should not have a leading '&'). Examples 10.2.8/9 and 10.5.6/7 show *By-Reference OpenURLs*. It should be noted that any whitespace within a *KEV By-Reference* document, possibly added by a transport agent for compatibility with legacy networks, has to be removed before processing.

7 Guidelines for OpenURL Referrers

OpenURL Referrers (aka *OpenURL* Sources) are information providers who include *OpenURL* source links within the Web pages of their applications. They include electronic journal applications, 'abstract and indexing' services, and library OPACs.

OpenURL Referrers are strongly encouraged to provide a genuine *Referrer Identifier* within the *ContextObject* as specified in Section 4.6. This will assist general *OpenURL* interoperability enabling the interpretation of local *Identifiers*, as well as *Resolver* functionality such the provision of usage statistics, and the prevention of circular links. Inclusion of this provenance within an *OpenURL* is an indication of its quality. A genuine *Referrer Identifier* is necessary where a *ContextObject* includes *Referrer*-specific Private Data, in order to define the provenance of that private data.

OpenURL Referrers are encouraged to supply as much information within the *ContextObject* as they have available. There is no minimum recommended information set that *Referrers* should supply, all items being optional, but common sense indicates that sufficient data should be included to produce the required action.

Because the *OpenURL* Framework Standard does not prescribe *Resolver* behavior, there is no guarantee that *Resolvers* that process the *OpenURLs* will take action on all of this information. In particular, *Referrers* cannot rely on *Resolvers* providing the requested service when a particular *ServiceType* is included in an *OpenURL*. Requested *ServiceTypes* should be regarded as hints to the *Resolver* (see section 5.4.5). The use of *Private Data* within a *ContextObject* is discouraged because it reduces interoperability, requiring a prior understanding between a *Referrer* and *Resolver*.

All KEV *OpenURLs* must be URL-encoded according to the rules given in Section 6 and the URL-encoding definition in Appendix B.3.

Implementers of *OpenURL Referrers* should be aware of the problems associated with upgrading to *OpenURL* version 1.0 from *OpenURL* version 0.1. A recommendation on supplying 'interim' hybrid *OpenURLs* is given in Appendix A.2.

8 Guidelines for OpenURL Resolvers

An *OpenURL Resolver* is an HTTP-based Internet service that is the target of a request for services when an end-user activates an *OpenURL* source link. Typically the *Resolver* will provide to the end-user a set of services that pertain to the *Referent* within the *ContextObject* of the *OpenURL*. These services will be appropriate for the end-user taking into account the user's location and privileges. Typically *OpenURL Resolvers* include knowledge bases that record subscription, holdings and preference information for their organization.

Conformance to Profiles requires support for all their defined components (see section 5.6). To claim conformance to the San Antonio Level 1 Profile, *OpenURL Resolvers* should support all the selections from the *Registry* listed in Section 5, as defined in Appendix C.5 of the Standard. However, the *OpenURL* Framework Standard does not prescribe *Resolver* behavior, and it is not a protocol. Thus there is no requirement, beyond general expectation, for the actions that a *Resolver* should take on any of these selections. A SAP1 compliant *Resolver* should resolve all of the SAP1 *Identifier Namespaces* and *Metadata Formats* that it is capable of and handle gracefully the ones it cannot resolve.

It is expected that *Resolvers* will deal gracefully with foreign keys (see Appendix B.2). During an interim phase when implementers of *Referrers* are upgrading to *OpenURL* version 1.0 from *OpenURL* version 0.1, *Resolvers* should be aware that they may receive both forms of *OpenURL* as well as some hybrid ones (see Appendix A.2). Although the Standard makes no requirements on *Resolver* behavior, it would seem reasonable that *Resolvers* should continue to accept version 0.1 *OpenURLs* for the time being.

Because the OpenURL Framework does not prescribe Resolver behavior, there is no required action when a received OpenURL results in more than one item. A Resolver may return a list of items, or it may return nothing, dependent on the choice of the particular Resolver.

Implementers of *OpenURL Resolvers* should follow the guidelines in Appendix B.4 for URL-decoding and Appendix B.5 for query string parsing. They are advised to be aware of the security considerations and precautions given in Appendix C, and considerations on the length of URLs in Appendix B.1. If part of a *Resolver*'s operation involves redirecting *OpenURLs* to another *OpenURL Resolver* or target, it should follow the guidelines in Appendix B.3 for URL-encoding, and should pass on all the Key/Value pairs in the *OpenURL* and its payload *ContextObject* without making any changes to them.

9 Guidelines for OpenURL 'link-to' Resolvers

Some *OpenURL Resolvers* just provide an *OpenURL* compliant 'link-to' syntax as an access point to their content. They do not provide further services related to the *Referent*, except for any Web links and *OpenURL* source links they may provide as part of their usual service.

It is probable that such *Resolvers* will implement only part of the functionality implied in an *OpenURL*. They will provide a service where the *OpenURL* maps onto their application and data. Thus it would be expected that they publish the *OpenURL* functionality supported and the expected result of *OpenURL* requests. In the future there may be an *OpenURL* 'link-to' Profile registered suitable for such *Resolvers* to indicate conformance. At present they could claim to be compliant with their published subset of SAP1.

10 Example Scenarios

Following are some example scenarios where *OpenURLs* could be used. The examples show the use of *Inline*, *By-Value* and *By-Reference KEV OpenURLs* within the scholarly information community. The choice of *OpenURL Transport* used in these examples is arbitrary. It is used to illustrate the coding of the *Transports* and does not indicate any relationship between scenario and choice of *Transport*. In all the examples, except 10.3, the *OpenURLs* show only the query string. This would be appended to a base URL with a '?' separator (and no leading '&') to create an actionable *OpenURL*.

10.1 Journal Article, Referenced from an Electronic Journal Service, Inline OpenURL

A journal article in D-Lib Magazine (ISSN 1082-9873):

> Caplan, Priscilla and Arms, William Y. 1999. "Reference Linking for Journal Articles." D-Lib Magazine. 5(7/8). <http://www.dlib.org/dlib/july99/caplan/07caplan.html> <doi:10.1045/july99-caplan>

is cited in the journal article:

> Van de Sompel, Herbert and Beit-Arie, Oren. 2001. "Open Linking in the Scholarly Information Environment Using the OpenURL Framework." D-Lib Magazine. 7(3). <http://www.dlib.org/dlib/march01/vandesompel/03vandesompel.html> <doi:10.1045/march2001-vandesompel>

Referent

The *Referent* is the first article. It could be described by an *Identifier* and also by metadata, as shown in Example 10.1.1.

Example 10.1.1 Referent

(Not URL-encoded and with line breaks for readability)

&rft_id=info:doi/10.1045/july99-caplan
&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&rft.genre=article
&rft.aulast=Caplan
&rft.aufirst=Priscilla
&rft.issn=1082-9873
&rft.volume=5
&rft.issue=7/8
&rft.date=1999
&rft.atitle=Reference Linking for Journal Articles

ReferringEntity

The *ReferringEntity* is the second article. It could be described by an *Identifier* and also by metadata, as shown in Example 10.1.2.

Example 10.1.2 ReferringEntity

(Not URL-encoded and with line breaks for readability)

&rfe_id=info:doi/10.1045/march2001-vandesompel
&rfe_val_fmt=info:ofi/fmt:kev:mtx:journal
&rfe.genre=article
&rfe.aulast=Van de Sompel
&rfe.aufirst=Herbert
&rfe.issn=1082-9873
&rfe.volume=7
&rfe.issue=3
&rfe.date=2001
&rfe.atitle=Open Linking in the Scholarly Information Environment using the OpenURL Framework

Requester

No information about the *Requester* is included in this example.

ServiceType

No information about the *ServiceType* is included in this example.

Resolver

The *Resolver* is the base URL to which the query string will be sent. It is not included explicitly in the *ContextObject*.

Referrer

The *Referrer* is the D-Lib electronic journal application. It can be described by an *Identifier* as shown in Example 10.1.3.

Example 10.1.3 Referrer

(Not URL-encoded and with line breaks for readability)

&rfr_id=info:sid/dlib.org:dlib

ContextObject

The above *Descriptors* would be combined into a *ContextObject* by including the Keys shown in Example 10.1.4.

Example 10.1.4 ContextObject

(Not URL-encoded and with line breaks for readability)

&ctx_ver=Z39.88-2004
&ctx_enc=info:ofi/enc:UTF-8
&ctx_id=10_1
&ctx_tim=2003-04-10T13:56:30TZD

OpenURL Keys

The *OpenURL* Keys for an *Inline OpenURL* are shown in Example 10.1.5.

Example 10.1.5 OpenURL Keys

(Not URL-encoded and with line breaks for readability)

url_ver=Z39.88-2004
&url_tim=2003-04-10T13:57:15TZD
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx

Inline OpenURL

The complete query string for the *OpenURL* is shown in Example 10.1.6.

Example 10.1.6 Inline OpenURL

(URL-encoded)

url_ver=Z39.88-2004&url_tim=2003-04-10T13%3A57%3A15TZD&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx&ctx_ver=Z39.88-2004&ctx_enc=info%3Aofi%2Fenc%3AUTF-8&ctx_id=10_1&ctx_tim=2003-04-10T13%3A56%3A30TZD&rft_id=info%3Adoi%2F10.1045%2Fjuly99-caplan&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rft.genre=article&rft.aulast=Caplan&rft.aufirst=Priscilla&rft.issn=1082-9873&rft.volume=5&rft.issue=7/8&rft.date=1999&rft.atitle=Reference+Linking+for+Journal+Articles&rfe_id=info%3Adoi%2F10.1045%2Fmarch2001-vandesompel&rfe_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rfe.genre=article&rfe.aulast=Van+de+Sompel&rfe.aufirst=Herbert&rfe.issn=1082-9873&rfe.volume=7&rfe.issue=3&rfe.date=2001&rfe.atitle=Open+Linking+in+the+Scholarly+Information+Environment+using+the+OpenURL+Framework&rfr_id=info%3Asid%2Fdlib.org%3Adlib

10.2 Journal Article, Referenced from an Electronic Journal Service, By-Reference OpenURL

Jane Doe, a University student at Caltech, reads the following article in the Elsevier ScienceDirect® collection:

> McArthur, James G. et al. 2001. "p27-p16 Chimera: A Superior Antiproliferative for the Prevention of Neointimal Hyperplasia." Molecular Therapy. 3(1) 8-13. <doi:10.1006/mthe.2000.0239>

In the reference list of that article she finds a reference to the following article and would like to view its full text:

> Bergelson, J. 1997. "Isolation of a common receptor for coxsackie B viruses and adenoviruses 2 and 5." Science. (275) 1320-1323. <doi:10.1126/science.275.5304.1320> <pmid:9036860>

Referent

The *Referent* is the second article. It could be described by two *Identifiers* and also by metadata, as shown in Example 10.2.1.

Example 10.2.1 Referent

(Not URL-encoded and with line breaks for readability)

```
&rft_id=info:doi/10.1126/science.275.5304.1320
&rft_id=info:pmid/9036860
&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&rft.genre=article
&rft.aulast=Bergelson
&rft.auinit=J
&rft.jtitle=Science
&rft.volume=275
&rft.spage=1320
&rft.epage=1323
&rft.date=1997
&rft.atitle=Isolation of a common receptor for coxsackie B viruses and
adenoviruses 2 and 5
```

ReferringEntity

The *ReferringEntity* is the first article. It could be described by an *Identifier* and also by metadata, as shown in Example 10.2.2.

Example 10.2.2 ReferringEntity

(Not URL-encoded and with line breaks for readability)

```
&rfe_id=info:doi/10.1006/mthe.2000.0239
&rfe_val_fmt=info:ofi/fmt:kev:mtx:journal
&rfe.genre=article
&rfe.aulast=McArthur
&rfe.aufirst=James
&ref.auinitm=G
&rfe.jtitle=Molecular Therapy
&rfe.volume=3
&rfe.issue=1
&rfe.date=2001
&rfe.atitle=p27-p16 Chimera: A Superior Antiproliferative for the Prevention of
Neointimal Hyperplasia
```

Requester

The *Requester* is Jane Doe, who could be identified by her email address as shown in Example 10.2.3.

Example 10.2.3 Requester

(Not URL-encoded and with line breaks for readability)

```
&req_id=mailto:jane.doe@caltech.edu
```

ServiceType

Jane would like to view the full text of the *Referent* article. This could be indicated by metadata as shown in Example 10.2.4.

Example 10.2.4 ServiceType

(Not URL-encoded and with line breaks for readability)

```
&svc_val_fmt=info:ofi/fmt:kev:mtx:sch_svc
&svc.fulltext=yes
```

Resolver

The *Resolver* is the base URL to which the query string will be sent. It is not included explicitly in the *ContextObject*.

Referrer

The *Referrer* is Elsevier's ScienceDirect® electronic journal application. It can be described by an *Identifier* as shown in Example 10.2.5.

Example 10.2.5 Referrer

(Not URL-encoded and with line breaks for readability)

```
&rfr_id=info:sid/elsevier.com:ScienceDirect
```

ContextObject

The above *Descriptors* are combined into a *ContextObject* by including the Keys shown in Example 10.2.6.

Example 10.2.6 ContextObject

(Not URL-encoded and with line breaks for readability)

```
ctx_ver=Z39.88-2004
&ctx_enc=info:ofi/enc:UTF-8
&ctx_id=10_2
&ctx_tim=2003-04-10T14:25:30TZD
```

OpenURL Keys

The *OpenURL* Keys for a *By-Reference OpenURL* are shown in Example 10.2.7.

Example 10.2.7 OpenURL Keys

(Not URL-encoded and with line breaks for readability)

```
url_ver=Z39.88-2004
&url_tim=2003-04-10T14:28:15TZD
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
&url_ctx_ref=http://example.org/temp/10_2.txt
```

By-Reference OpenURL

The complete query string for the *OpenURL* is shown in Example 10.2.8. The *ContextObject* that resides at http://example.org/temp/10_2.txt is shown in Example 10.2.9. Note that the Metadata Keys within the *ContextObject* are prefixed, because the entire *ContextObject* is located in one place. Metadata Key prefixes are omitted only when an *Entity* within a *ContextObject* is By-Reference.

Example 10.2.8 By-Reference OpenURL

```
(URL-encoded)

url_ver=Z39.88-2004&url_tim=2003-04-
10T14%3A28%3A15TZD&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx
&url_ctx_ref=http%3A//example.org/temp/10_2.txt
```

Example 10.2.9 Referenced ContextObject

```
(URL-encoded)

ctx_ver=Z39.88-2004&ctx_enc=info%3Aofi%2Fenc%3AUTF-
8&ctx_id=10_2&ctx_tim=2003-04-
10T14%3A25%3A30TZD&rft_id=info%3Adoi%2F10.1126%2Fscience.275.5304.132
0&rft_id=info%3Apmid%2F9036860&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3A
mtx%3Ajournal&rft.genre=article&rft.aulast=Bergelson&rft.auinit=J&rft.jtitle=Sci
ence&rft.volume=275&rft.spage=1320&rft.epage=1323&rft.date=1997&rft.atitle=Is
olation+of+a+common+receptor+for+coxsackie+B+viruses+and+adenoviruses+2
+and+5&rfe_id=info%3Adoi%2F10.1006%2Fmthe.2000.0239&rfe_val_fmt=info%3
Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rfe.genre=article&rfe.aulast=McArthur&
rfe.aufirst=James&ref.auinitm=G&rfe.jtitle=Molecular+Therapy&rfe.volume=3&rfe
.issue=1&rfe.date=2001&rfe.atitle=p27-
p16+Chimera%3A+A+Superior+Antiproliferative+for+the+Prevention+of+Neointi
mal+Hyperplasia&req_id=mailto%3Ajane.doe%40caltech.edu&svc_val_fmt=info
%3Aofi%2Ffmt%3Akev%3Amtx%3Asch_svc&svc.fulltext=yes&rfr_id=info%3Asid
%2Felsevier.com%3AScienceDirect
```

10.3 Journal Article, Referenced from an Abstracts Service, By-Value OpenURL, POST

This article:

> Sturino JM and Klaenhammer TR. 2002. “Expression of Antisense RNA Targeted against Streptococcus thermophilus Bacteriophages.” Appl Environ Microbiol. 2002 Feb; 68(2):588-596

can be found in Entrez PubMed:

> http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?
> cmd=Retrieve&db=PubMed&list_uids=11823195&dopt=Abstract

Referent

The *Referent* is the article. It can be described by metadata, as shown in Example 10.3.1.

Example 10.3.1 Referent

```
(Not URL-encoded and with line breaks for readability)

&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&rft.genre=article
&rft.aulast=Sturino
&rft.auinit=JM
&rft.stitle=Appl Environ Microbiol
&rft.volume=68
&rft.issue=2
&rft.spage=588
&rft.epage=596
&rft.date=2002-02
&rft.atitle=Expression of Antisense RNA Targeted against Streptococcus
thermophilus Bacteriophages
```

ReferringEntity

The *ReferringEntity* is the record for the article in Entrez PubMed. It can be described by an *Identifier*, as shown in Example 10.3.2.

Example 10.3.2 ReferringEntity

(Not URL-encoded and with line breaks for readability)

&rfe_id=info:pmid/11823195

Requester

Via the Cubby system, PubMed may hold information, such as the email address, about the *Requester* activating the *OpenURL*. If *Requester* is Fred Bloggs, he could be identified by his email address as shown in Example 10.3.3.

Example 10.3.3 Requester

(Not URL-encoded and with line breaks for readability)

&req_id=mailto:fred.bloggs@example.org

ServiceType

A PubMed user would probably wish to view the full text of the *Referent* article. This could be identified by metadata as shown in Example 10.3.4.

Example 10.3.4 ServiceType

(Not URL-encoded and with line breaks for readability)

&svc_val_fmt=info:ofi/fmt:kev:mtx:sch_svc
&svc.fulltext=yes

Resolver

The *Resolver* is the base URL to which the query string will be sent. It is not included explicitly in the *ContextObject*.

Referrer

The *Referrer* is PubMed. It can be described by an *Identifier* as shown in Example 10.3.5.

Example 10.3.5 Referrer

(Not URL-encoded and with line breaks for readability)

&rfr_id=info:sid/ncbi.nlm.nih.gov:pubmed

ContextObject

The above *Descriptors* would be combined into a *ContextObject* by including the Keys shown in Example 10.3.6.

Example 10.3.6 ContextObject

(Not URL-encoded and with line breaks for readability)

ctx_ver=Z39.88-2004
&ctx_enc=info:ofi/enc:UTF-8
&ctx_id=10_3
&ctx_tim=2003-04-10T15:35:30TZD

OpenURL Keys

The *OpenURL* Keys for a *By-Value OpenURL* are shown in Example 10.3.7.

Example 10.3.7 OpenURL Keys

(Not URL-encoded and with line breaks for readability)

url_ver=Z39.88-2004
&url_tim=2003-04-10T15:37:15TZD

```
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
&url_ctx_val=
```

By-Value OpenURL

The complete *By-Value OpenURL* is shown using HTTP(S) POST in Example 10.3.8. Note that the values within the *ContextObject* are double URL-encoded.

Example 10.3.8 By-Value OpenURL using HTTP(S) POST

```
POST /myresolver HTTP/1.1
Host: www.example.org
Content-Length: 784
Content-Type: application/x-www-form-urlencoded

url_ver=Z39.88-2004&url_tim=2003-04-
10T15%3A37%3A15TZD&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx
&url_ctx_val=ctx_ver%3DZ39.88-
2004%26ctx_enc%3Dinfo%253Aofi%252Fenc%253AUTF-
8%26ctx_id%3D11_3%26ctx_tim%3D2003-04-
10T15%253A35%253A30TZD%26rft_val_fmt%3Dinfo%253Aofi%252Ffmt%253Ake
v%253Amtx%253Ajournal%26rft.genre%3Darticle%26rft.aulast%3DSturino%26rft
.auinit%3DJM%26rft.stitle%3DAppl%2Benviron%2BMicrobiol%26rft.volume%3D6
8%26rft.issue%3D2%26rft.spage%3D588%26rft.epage%3D596%26rft.date%3D200
2-
02%26rft.atitle%3DExpression%2Bof%2Bantisene%2BRNA%2Btargeted%2Bagai
nst%2Bstreptococcus%2Bthermophilus%2BBacteriophages%26rfe_id%3Dinfo%
253Apmid%252F11823195%26req_id%3Dmailto%253Afred.bloggs%2540example
.org%26svc_val_fmt%3Dinfo%253Aofi%252Ffmt%253Akev%253Amtx%253Asch_
svc%26svc.fulltext%3Dyes%26rfr_id%3Dinfo%253Asid%252Fncbi.nlm.nih.gov%
253Apubmed
```

10.4 Conference Paper in a Journal, Referenced from a Table of Contents Service, Inline OpenURL

This conference paper, which is published in a journal:

> Apps A and MacIntyre R. 2002. “Prototyping Digital Library Technologies in zetoc.” In: ECDL 2002: European Conference on research and advanced technologies for digital libraries. Lecture Notes in Computer Science. 2458:309-323. ISSN: 0302-9743. ISBN: 3540441786.

was found in **zetoc**, the British Library’s Electronic Table of Contents current awareness service (http://zetoc.mimas.ac.uk), with an identifier: CN045159474, by a user at The University of Manchester in the UK.

Referent

The *Referent* is the paper. It can be described by metadata, as shown in Example 10.4.1.

Example 10.4.1 Referent

```
(Not URL-encoded and with line breaks for readability)

&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&rft.genre=proceeding
&rft.aulast=Apps
&rft.auinit=A
&rft.issn=0302-9743
&rft.jtitle=Lecture Notes in Computer Science
&rft.issue=2458
&rft.spage=309
&rft.epage=323
&rft.date=2002
```

```
&rft.atitle=Prototyping Digital Library Technologies in zetoc
```

ReferringEntity

The *ReferringEntity* is the record for the paper in **zetoc**. It can be described by a local identifier within private data, as shown in Example 10.4.2. This data is specific to **zetoc** and will be understood by a *Resolver* only where a prior understanding exists between them.

Example 10.4.2 ReferringEntity

```
(Not URL-encoded and with line breaks for readability)

&rfe_dat=zetocid:CN045159474
```

Requester

The user is at the University of Manchester, where the Athens authentication system is in use. The *Requester Entity* includes information about the user's institution, a three letter prefix of their Athens user name. This is described by a local identifier within private data, as shown in Example 10.4.3, which will not be understood by the *Resolver* unless some prior agreement has been made.

Example 10.4.3 Requester

```
(Not URL-encoded and with line breaks for readability)

&req_dat=athens:man
```

ServiceType

No information about the *ServiceType* is included in this example.

Resolver

The *Resolver* is the base URL to which the query string will be sent. It is not included explicitly in the *ContextObject*.

Referrer

The *Referrer* is the **zetoc** application. It can be described by an *Identifier* as shown in Example 10.4.4.

Example 10.4.4 Referrer

```
(Not URL-encoded and with line breaks for readability)

&rfr_id=info:sid/mimas.ac.uk:zetoc
```

ContextObject

The above *Descriptors* would be combined into a *ContextObject* by including the Keys shown in Example 10.4.5.

Example 10.4.5 ContextObject

```
(Not URL-encoded and with line breaks for readability)

&ctx_id=10_4
&ctx_tim=2003-04-10T16:23:30TZD
```

OpenURL Keys

The *OpenURL* Keys for an *Inline OpenURL* are shown in Example 10.4.6.

Example 10.4.6 OpenURL Keys

```
(Not URL-encoded and with line breaks for readability)

url_ver=Z39.88-2004
&url_tim=2003-04-10T16:24:15TZD
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
```

Inline OpenURL

The complete query string for the *OpenURL* is shown in Example 10.4.7.

Example 10.4.7 Inline OpenURL

(URL-encoded)

```
url_ver=Z39.88-2004&url_tim=2003-04-
10T16%3A24%3A15TZD&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx
&ctx_id=10_4&ctx_tim=2003-04-
10T16%3A23%3A30TZD&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajour
nal&rft.genre=proceeding&rft.aulast=Apps&rft.auinit=A&rft.issn=0302-
9743&rft.jtitle=Lecture+Notes+in+Computer+Science&rft.issue=245&rft.spage=3
09&rft.epage=323&rft.date=2002&rft.atitle=Prototyping+Digital+Library+Technolo
gies+inzetoc&rfe_dat=zetocid%3ACN045159474&req_dat=athens%3Aman&rfr_id
=info%3Asid%2Fmimas.ac.uk%3Azetoc
```

10.5 Conference Paper, Referenced from a Table of Contents Service, By-Reference OpenURL

This conference paper:

> Apps A, MacIntyre R and Morris L. 2002. "Exposing Cross-Domain Resources for Researchers and Learners." In: Dc-2002: Dublin and metadata for e-communities. Firenze University Press. pp 71-80. ISBN: 8884530431.

can be found in **zetoc**, the British Library's Electronic Table of Contents service (http://zetoc.mimas.ac.uk), with an identifier: CN046568515

Referent

The *Referent* is the paper. It can be described by *By-Reference Metadata*, as shown in Example 10.5.1. Note that for *By-Reference Metadata*, the Metadata Keys are not prefixed.

Example 10.5.1 Referent

(Not URL-encoded and with line breaks for readability)

```
&rft_ref_fmt=info:ofi/fmt:kev:mtx:book
&rft_ref=http://www.example.org/myeg.txt
```

(At http://www.example.org/myeg.txt and with line breaks for readability)

```
genre=proceeding
&aulast=Apps
&auinit=A
&isbn=8884530431
&pub=Firenze University Press
&date=2002
&spage=71
&epage=80
&atitle=Exposing Cross-Domain Resources for Researchers and Learners
```

ReferringEntity

The *ReferringEntity* is the record for the paper in **zetoc**. It can be described by a local identifier within private data, as shown in Example 10.5.2. This data is specific to **zetoc** and will be understood by a *Resolver* only where a prior understanding exists between them.

Example 10.5.2 ReferringEntity

(Not URL-encoded and with line breaks for readability)

```
&rfe_dat=zetocid:CN046568515
```

Requester

No information about the *Requester* is included in this example.

ServiceType

No information about the *ServiceType* is included in this example.

Resolver

The *Resolver* is the base URL to which the query string will be sent. It is not included explicitly in the *ContextObject*.

Referrer

The *Referrer* is the **zetoc** application. It can be described by an *Identifier* as shown in Example 10.5.3.

Example 10.5.3 Referrer

(Not URL-encoded and with line breaks for readability)

```
&rfr_id=info:sid/mimas.ac.uk:zetoc
```

ContextObject

The above *Descriptors* would be combined into a *ContextObject* by including the Keys shown in Example 10.5.4.

Example 10.5.4 ContextObject

(Not URL-encoded and with line breaks for readability)

```
ctx_ver=Z39.88-2004
&ctx_enc=info:ofi/enc:UTF-8
&ctx_id=10_5
&ctx_tim=2003-04-10T16:41:30TZD
```

OpenURL Keys

The *OpenURL* Keys for a *By-Reference OpenURL* are shown in Example 10.5.5.

Example 10.5.5 OpenURL Keys

(Not URL-encoded and with line breaks for readability)

```
url_ver=Z39.88-2004
&url_tim=2003-04-10T16:43:15TZD
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
&url_ctx_ref=http://example.org/temp/10_5.txt
```

By-Reference OpenURL

The complete query string for the *OpenURL* is shown in Example 10.5.6. The *ContextObject* that resides at http://example.org/temp/10_5.txt is shown in Example 10.5.7.

Example 10.5.6 By-Reference OpenURL

(URL-encoded)

```
url_ver=Z39.88-2004&url_tim=2003-04-
10T16%3A43%3A15TZD&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx
&url_ctx_ref=http%3A//example.org/temp/10_5.txt
```

Example 10.5.7 Referenced ContextObject

(URL-encoded)

```
ctx_ver=Z39.88-2004&ctx_enc=info%3Aofi%2Fenc%3AUTF-
8&ctx_id=10_5&ctx_tim=2003-04-
10T16%3A41%3A30TZD&rft_ref_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Aboo
k&rft_ref=http%3A%2F%2Fwww.example.org%2Fmyeg.txt&rfe_dat=zetocid%3A
```

```
CN046568515&rfr_id=info%3Asid%2Fmimas.ac.uk%3Azetoc
```

10.6 Book, Referenced from a Shopping Service, By-Value OpenURL

This book is available from Amazon:

Professional XML Meta Data. 2001. By David Dodds, et al. ISBN: 1861004516. <http://www.amazon.com/exec/obidos/ASIN/1861004516>

Referent

The *Referent* is the book. It could be described by an *Identifier* and also by metadata, as shown in Example 10.6.1.

Example 10.6.1 Referent

(Not URL-encoded and with line breaks for readability)

```
&rft_id=urn:isbn:1861004516
&rft_val_fmt=info:ofi/fmt:kev:mtx:book
&rft.genre=book
&rft.aulast=Dodds
&rft.aufirst=David
&rft.isbn=1861004516
&rft.date=2001
&rft.btitle=Professional XML Meta Data
```

ReferringEntity

The *ReferringEntity* is the page within Amazon where the book was found. It can be described by an *Identifier*, as shown in Example 10.6.2. It could be used by Amazon to return the user back to the originating page after going through the “shopping cart” procedure.

Example 10.6.2 ReferringEntity

(Not URL-encoded and with line breaks for readability)

```
&rfe_id=http://www.amazon.com/exec/obidos/ASIN/1861004516
```

Requester

The *Requester* may be the *Identifier* of the session for the user within Amazon. This can be described by a local Amazon-specific identifier within private data, as shown in Example 10.6.3.

Example 10.6.3 Requester

(Not URL-encoded and with line breaks for readability)

```
&req_dat=104-011434-4639158
```

ServiceType

The *ServiceType* is the Amazon-specific service “add to shopping cart” This can be described by a local identifier within private data, as shown in Example 10.6.4.

Example 10.6.4 ServiceType

(Not URL-encoded and with line breaks for readability)

```
&svc_dat=addToCart
```

Resolver

The *Resolver* is the base URL to which the query string will be sent. It is not included explicitly in the *ContextObject*.

Referrer

The *Referrer* is the Amazon application. It can be described by an *Identifier* as shown in Example 10.6.5.

Example 10.6.5 Referrer

```
(Not URL-encoded and with line breaks for readability)

&rfr_id=info:sid/amazon.com
```

ContextObject

The above *Descriptor*s would be combined into a *ContextObject* by including the Keys shown in Example 10.6.6.

Example 10.6.6 ContextObject

```
(Not URL-encoded and with line breaks for readability)

&ctx_id=10_6
&ctx_tim=2003-04-10T17:11:30TZD
```

OpenURL Keys

The *OpenURL* Keys for a *By-Value OpenURL* are shown in Example 10.6.7.

Example 10.6.7 OpenURL Keys

```
(Not URL-encoded and with line breaks for readability)

url_ver=Z39.88-2004
&url_tim=2003-04-10T17:12:15TZD
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
&url_ctx_val=
```

By-Value OpenURL

The complete query string for the *OpenURL* is shown in Example 10.6.8. Note that the values within the *ContextObject* are double URL-encoded.

Example 10.6.8 By-Value OpenURL

```
(URL-encoded)

url_ver=Z39.88-2004&url_tim=2003-04-
10T17%3A12%3A15TZD&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx
&url_ctx_val=%26ctx_id%3D11_6%26ctx_tim%3D2003-04-
10T17%253A11%253A30TZD%26rft_id%3Durn%253Aisbn%253A1861004516%26r
ft_val_fmt%3Dinfo%253Aofi%252Ffmt%253Akev%253Amtx%253Abook%26rft.ge
nre%3Dbook%26rft.aulast%3DDodds%26rft.aufirst%3DDavid%26rft.isbn%3D186
1004516%26rft.date%3D2001%26rft.btitle%3DProfessional%2BXML%2Bmeta%2B
Data%26rfe_id%3Dhttp%253A%252F%252Fwww.amazon.com%252Fexec%252Fo
bidos%252FASIN%252F1861004516%26req_dat%3D104-011434-
4639158%26svc_dat%3DaddToCart%26rfr_id%3Dinfo%253Asid%252Famazon.co
m
```

10.7 Journal, Referenced from a Serials Directory, Inline OpenURL

Ulrich's Serials Directory records the following journal. The record is accessed by Mary Smith via her institution's copy of Ulrich's. She would like to find a list of services concerning this journal, for instance where she can find electronic copies and which libraries hold print copies.

Title: European Journal of Pain
Abbreviate Title: Eur J Pain
Publisher: Harcourt Health Sciences
ISSN: 1090-3801
Accession Number: 14587033

Referent

The *Referent* is the journal. It could be described by an *Identifier* and also by metadata, as shown in Example 10.7.1.

Example 10.7.1 Referent

(Not URL-encoded and with line breaks for readability)

&rft_id=urn:issn:1090-3801
&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&rft.genre=journal
&rft.issn=1090-3801
&rft.jtitle=European Journal of Pain
&rft.stitle=Eur J Pain

ReferringEntity

The *ReferringEntity* is the record in Ulrich's. It could be described by Ulrich's-specific private data, as shown in Example 10.7.2.

Example 10.7.2 ReferringEntity

(Not URL-encoded and with line breaks for readability)

&rfe_dat=14587033

Requester

The *Requester* is Mary Smith. Because she has been authenticated to use her institution's copy of Ulrich's, the *Requester* is described by an *Identifier* that is an entry in her institution's LDAP directory, as shown in Example 10.7.3.

Example 10.7.3 Requester

(Not URL-encoded and with line breaks for readability)

&req_id=ldap://ldap.auni.edu/cn=mary_smith

ServiceType

No information about the *ServiceType* is included in this example. The service Mary wants is the usual service provided by her institution's *Resolver*.

Resolver

The *Resolver* is the base URL to which the query string will be sent. It is not included explicitly in the *ContextObject*.

Referrer

The *Referrer* is the local institution's version of Ulrich's. It may be described by private data as shown in Example 10.7.4.

Example 10.7.4 Referrer

(Not URL-encoded and with line breaks for readability)

&rfr_id=info:sid/auni.edu:ULRICH

ContextObject

The above *Descriptor*s would be combined into a *ContextObject* by including the Keys shown in Example 10.7.5.

Example 10.7.5 ContextObject

(Not URL-encoded and with line breaks for readability)

&ctx_id=10_7
&ctx_tim=2003-04-11T09:35:30TZD

OpenURL Keys

The *OpenURL* Keys for an *Inline OpenURL* are shown in Example 10.7.6.

Example 10.7.6 OpenURL Keys

(Not URL-encoded and with line breaks for readability)

```
url_ver=Z39.88-2004
&url_tim=2003-04-11T09:36:15TZD
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
```

Inline OpenURL

The complete query string for the *OpenURL* is shown in Example 10.7.7.

Example 10.7.7 Inline OpenURL

(URL-encoded)

```
url_ver=Z39.88-2004&url_tim=2003-04-
11T09%3A36%3A15TZD&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx
&ctx_id=10_7&ctx_tim=2003-04-
11T09%3A35%3A30TZD&rft_id=urn%3Aissn%3A1090-
3801&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Ajournal&rft.genre=jour
nal&rft.issn=1090-
3801&rft.jtitle=European+Journal+of+Pain&rft.stitle=Eur+J+Pain&rfe_dat=145870
33&req_id=ldap:%2F%2Fldap.auni.edu%2Fcn%3Dmary_smith&rfr_id=info%3Asi
d%2Fauni.edu%3AULRICH
```

10.8 Book, Referenced from a Book, Inline OpenURL

This book:

> Vergnaud, J.-R. 1985 “Dépendances et niveaux de représentation en syntaxe.” Amsterdam, Philadelphia: Benjamins

is cited in the following book:

> Chomsky, N. 1995. “The Minimalist Program.” Cambridge, Mass: The MIT Press. ISBN 0262531283

Referent

The *Referent* is the first book. It could be described by metadata, as shown in Example 10.8.1.

Example 10.8.1 Referent

(Not URL-encoded and with line breaks for readability)

```
&rft_val_fmt=info:ofi/fmt:kev:mtx:book
&rft.genre=book
&rft.aulast=Vergnaud
&rft.auinit=J.-R.
&rft.btitle=Dépendances et niveaux de représentation en syntaxe
&rft.date=1985
&rft.pub=Benjamins
&rft.place=Amsterdam, Philadelphia
```

The title of this book is in French and includes two occurrences of ‘e’ with an acute accent. These characters must be encoded according to the *Character Encoding* specified for the *ContextObject*, in this case UTF-8 (see Example 10.8.5 below). The UTF-8 encoding for ‘Latin Small Letter E with Acute’ is ‘0xc3 0xa9’. After URL-encoding this becomes ‘%C3%A9’ (as shown in the complete *OpenURL* in Example 10.8.7).

ReferringEntit*y*

The *ReferringEntity* is the second book. It could be described by an *Identifier* and also by metadata, as shown in Example 10.8.2.

Example 10.8.2 ReferringEntity

(Not URL-encoded and with line breaks for readability)

```
&rfe_id=urn:isbn:0262531283
&rfe_val_fmt=info:ofi/fmt:kev:mtx:book
&rfe.genre=book
&rfe.aulast=Chomsky
&rfe.init=N
&rfe.btitle=The Minimalist Program
&rfe.isbn=0262531283
&rfe.date=1995
&rfe.pub=The MIT Press
&rfe.place=Cambridge, Mass
```

Requester

No information about the *Requester* is included in this example.

ServiceType

The *Requester* would like to see an abstract of the cited book. This can be described by metadata, as shown in Example 10.8.3.

Example 10.8.3 ServiceType

(Not URL-encoded and with line breaks for readability)

```
&svc_val_fmt=info:ofi/fmt:kev:mtx:sch_svc
&svc.abstract=yes
```

Resolver

The *Resolver* is the base URL to which the query string will be sent. It is not included explicitly in the *ContextObject*.

Referrer

The *Referrer* is the electronic book application in which the user is reading the second book. It may be described by an *Identifier* as shown in Example 10.8.4.

Example 10.8.4 Referrer

(Not URL-encoded and with line breaks for readability)

```
&rfr_id=info:sid/ebookco.com:bookreader
```

ContextObject

The above *Descriptors* would be combined into a *ContextObject* by including the Keys shown in Example 10.8.5.

Example 10.8.5 ContextObject

(Not URL-encoded and with line breaks for readability)

```
&ctx_ver=Z39.88-2004
&ctx_enc=info:ofi/enc:UTF-8
&ctx_id=10_8
&ctx_tim=2003-04-11T10:08:30TZD
```

OpenURL Keys

The *OpenURL* Keys for an *Inline OpenURL* are shown in Example 10.8.6.

Example 10.8.6 OpenURL Keys

```
(Not URL-encoded and with line breaks for readability)

url_ver=Z39.88-2004
&url_tim=2003-04-11T10:09:15TZD
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
```

Inline OpenURL

The complete query string for the *OpenURL* is shown in Example 10.8.7.

Example 10.8.7 Inline OpenURL

```
(URL-encoded)

url_ver=Z39.88-2004&url_tim=2003-04-
11T10%3A09%3A15TZD&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx
&ctx_ver=Z39.88-2004&ctx_enc=info%3Aofi%2Fenc%3AUTF-
8&ctx_id=10_8&ctx_tim=2003-04-
11T10%3A08%3A30TZD&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Aboo
k&rft.genre=book&rft.aulast=Vergnaud&rft.auinit=J.-
R.&rft.btitle=D%C3%A9pendances+et+niveaux+de+repr%C3%A9sentation
en+syntaxe&rft.date=1985&rft.pub=Benjamins&rft.place=Amsterdam%2C+Philad
elphia&rfe_id=urn%3Aisbn%3A0262531283&rfe_val_fmt=info%3Aofi%2Ffmt%3A
kev%3Amtx%3Abook&rfe.genre=book&rfe.aulast=Chomsky&rfe.init=N&rfe.btitle
=The+Minimalist+Program&rfe.isbn=0262531283&rfe.date=1995&rfe.pub=The+MI
T+Press&rfe.place=Cambridge%2C+Mass&svc_val_fmt=info%3Aofi%2Ffmt%3Ak
ev%3Amtx%3Asch_svc&svc.abstract=yes&rfr_id=info%3Asid%2Febookco.com
%3Abookreader
```

Appendix A. Upgrading to OpenURL version 1.0

At first sight it seems that *OpenURL* version 1.0 has become over-complicated and more difficult to implement than the original draft *OpenURL* version 0.1. However, following the San Antonio Level 1 Profile for the scholarly information community, and using an *Inline OpenURL* it is possible to upgrade current version 0.1 *OpenURLs* to version 1.0 *OpenURLs* without too much difficulty. The purpose of much of the extra functionality provided by version 1.0 *OpenURL* is to enable its generalisation and thus its extension to other communities and application domains. Some of the apparent new items within an *OpenURL* were, in reality, already being carried in version 0.1 *OpenURLs* as *Private Data*. Formalising the method of encoding this currently private information will increase the interoperability of *OpenURLs*.

A.1 Difference between version 1.0 Inline OpenURL and version 0.1 OpenURL

Taking the example shown in Section 3, a version 0.1 *OpenURL* could be as in Example A.1.

Example A.1 A version 0.1 OpenURL

```
(Not URL-encoded and with line breaks for readability)

http://example.org/myResolver?
sid=myid:mydb
&id=doi:10.1126/science.275.5304.1320
&id=pmid:9036860
&genre=article
&atitle=Isolation of a common receptor for coxsackie B
&title=Science
&aulast=Bergelson
&auinit=J
&date=1997
&volume=275
&spage=1320
```

```
&epage=1323
```

(URL-encoded)

```
http://example.org/myResolver?sid=myid%3Amydb&id=doi%3A10.1126%2Fscien
ce.275.5304.1320&id=pmid%3A9036860&genre=article&atitle=Isolation+of+a+co
mmon+receptor+for+coxsackie+B&title=Science&aulast=Bergelson&auinit=J&d
ate=1997&volume=275&spage=1320&epage=1323
```

A version 1.0 *Inline OpenURL* for this example could be as in Example A.2.

Example A.2 A version 1.0 Inline OpenURL

(Not URL-encoded and with line breaks for readability)

```
http://example.org/myResolver?
url_ver=Z39.88-2004
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
&rfr_id=info:sid/myid.com:mydb
&rft_id=info:doi/10.1126/science.275.5304.1320
&rft_id=info:pmid/9036860
&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&rft.genre=article
&rft.atitle=Isolation of a common receptor for coxsackie B
&rft.jtitle=Science
&rft.aulast=Bergelson
&rft.auinit=J
&rft.date=1997
&rft.volume=275
&rft.spage=1320
&rft.epage=1323
```

(URL-encoded)

```
http://example.org/myResolver?url_ver=Z39.88-
2004&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx&rfr_id=info%3Asid
%2Fmyid.com%3Amydb&rft_id=info%3Adoi%2F10.1126%2Fscience.275.5304.13
20&rft_id=info%3Apmid%2F9036860&rft_val_fmt=info%3Aofi%2Ffmt%3Akev%3A
mtx%3Ajournal&rft.genre=article&rft.atitle=Isolation+of+a+common+receptor+fo
r+coxsackie+B&rft.jtitle=Science&rft.aulast=Bergelson&rft.auinit=J&rft.date=199
7&rft.volume=275&rft.spage=1320&rft.epage=1323
```

Comparing these two examples it can be seen that the following changes have been made to upgrade the *OpenURL* from version 0.1 to version 1.0:

- The required Key **url_ver** has been added to indicate the version of the *OpenURL*. *OpenURLs* with no version Key are assumed to be version 0.1
- The Key **url_ctx_fmt** has been added to indicate that the *ContextObject Format* is KEV.
- The **sid** Key has become the **rfr_id** Key and the encoding of the *Referrer Identifier* has been altered to follow the required syntax of the 'info:sid/' scheme.
- The **id** Keys have become **rft_id** Keys. The encoding of the *Identifiers* has been altered to use 'info:' *Namespace Identifiers*.
- The **rft_val_fmt** Key has been added to indicate that the *OpenURL* contains *By-Value Metadata* for a journal-type *Referent*.
- The metadata Keys have been prefixed by '**rft.**' to indicate that the metadata applies to the *Referent Entity*.
- The metadata Key for the journal has been changed to the preferred form of '**jtitle**'.

Both of these *OpenURLs* contain:

- *Referent*, described by two *Identifiers* and also *By-Value Metadata*
- *Referrer*. In *OpenURL* version 0.1 this is the 'sid'.

- *Resolver*, as a base URL. In a version 1.0 it is also possible to include the *Resolver* within the *ContextObject*, but this is optional.

The version 1.0 *OpenURL* could also contain: *ReferringEntity*; *Requester*; and *ServiceType*. These appear to be new additions. But, in fact, experience has shown that this information is regularly described in the *Private Data* area of version 0.1 *OpenURLs*.

A.1.1 Non-text Items

Guidelines for upgrading version 0.1 OpenURLs describing non-text items are given in section 5.4.7.1.

A.2 Hybrid OpenURLs

There will be a transition period, of indeterminate length, when applications using *OpenURLs* are upgrading to version 1.0. This transition period poses a particular problem for *Referrers*. In order to continue providing a quality service to their end users they will have to continue supplying version 0.1 *OpenURLs* until they are certain that all their clients' *Resolvers* have been upgraded. For a large information provider it may be difficult to know what version of *OpenURL* is in use by their customers. On the other hand, *Referrers* are encouraged to upgrade to version 1.0 because of the eventual increased functionality and interoperability this will provide.

A possible solution to this dilemma is for *Referrers* to supply hybrid *OpenURLs* that contain both version 1.0 and version 0.1 Keys where these differ. It is expected that *OpenURL Resolvers* will deal gracefully with foreign Keys that they do not understand by ignoring them. Example A.3 shows a hybrid *OpenURL* built from the above examples.

Example A.3 A Hybrid OpenURL

(Not URL-encoded and with line breaks for readability)

```
http://example.org/myResolver?
url_ver=Z39.88-2004
&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
&rfr_id=info:sid/myid.com:mydb
&sid=myid:mydb
&rft_id=info:doi/10.1126/science.275.5304.1320
&rft_id=info:ofi/pmid:9036860
&id=doi:10.1126/science.275.5304.1320
&id=pmid:9036860
&rft_val_fmt=info:ofi/fmt:kev:mtx:journal
&rft.genre=article
&rft.atitle=Isolation of a common receptor for coxsackie B
&rft.jtitle=Science
&rft.aulast=Bergelson
&rft.auinit=J
&rft.date=1997
&rft.volume=275
&rft.spage=1320
&rft.epage=1323&genre=article
&atitle=Isolation of a common receptor for coxsackie B
&title=Science
&aulast=Bergelson
&auinit=J
&date=1997
&volume=275
&spage=1320
&epage=1323
```

(URL-encoded)

```
http://example.org/myResolver?url_ver=Z39.88-
2004&url_ctx_fmt=info%3Aofi%2Ffmt%3Akev%3Amtx%3Actx&rfr_id=info%3Asid
```

```
%2Fmyid.com%3Amydb&sid=myid%3Amydb&rft_id=info%3Adoi%2F10.1126%2F
science.275.5304.1320&rft_id=info%3Apmid%2F9036860&id=doi%3A10.1126%2F
science.275.5304.1320&id=pmid%3A9036860&rft_val_fmt=info%3Aofi%2Ffmt%3
Akev%3Amtx%3Ajournal&rft.genre=article&rft.atitle=Isolation+of+a+common+re
ceptor+for+coxsackie+B&rft.jtitle=Science&rft.aulast=Bergelson&rft.auinit=J&rft.
date=1997&rft.volume=275&rft.spage=1320&rft.epage=1323&genre=article&atitle
=Isolation+of+a+common+receptor+for+coxsackie+B&title=Science&aulast=Ber
gelson&auinit=J&date=1997&volume=275&spage=1320&epage=1323
```

Appendix B. URL Considerations

This information is reproduced here from Appendix E of the Standard, for quick reference.

B.1 Length of HTTP GET URLs

Transport techniques based on HTTP GET are subject to length limitations on the GET URI. The *OpenURL* Standard does not place any a priori limit on the length of an *OpenURL*. However, *OpenURL Resolvers* must be able to accept *OpenURLs* as long as 255 bytes after encoding and should be able to accept *OpenURLs* as long as 2048 bytes. Implementers planning to store OpenURLs in databases for E-Res or similar systems should anticipate very long URL strings and should set the variable length to an appropriate value.) HTTP GET OpenURLs longer than 2048 bytes do not work in Microsoft Internet Explorer, so long OpenURLs are better sent over HTTP POST.

B.2 Foreign Keys

Foreign keys are any keys that are not *OpenURL* keys, *KEV ContextObject* Keys, *or KEV Metadata* keys. Foreign keys have no meaning assigned by the *OpenURL* Framework.

B.3 URL Encoding

URL Encoding and Decoding of HTTP GET and POST querystring values prevents the misinterpretation of special characters occurring in these values.

To form an Encoded Value from a Value, a procedure called URL encoding is used:

1. The alphanumeric characters “a” through “z”, “A” through “Z”, and “0” through “9” remain unchanged.
2. The special characters “.”, “-“, “*”, and “_” remain unchanged.
3. The space character “ ” is replaced with a plus sign “+” or the character string “%20” (quotes not included).
4. All other characters are unsafe. They are first converted into one or more bytes using the UTF-8 encoding method (or another encoding if specified by the *ContextObject Format*). Then, each byte is represented by the 3-byte string “%xy”, where xy is the two-digit hexadecimal representation of the byte.

B.4 URL Decoding

To form a Value from an Encoded Value, a procedure called URL decoding is used. It reverses the URL encoding procedure:

1. The plus sign “+” is replaced by space “ ”.
2. Each instance of a 3-byte string “%xy”, where xy is a hexadecimal number, is replaced with the corresponding byte.
3. The bytes are converted to Unicode characters using UTF-8, unless otherwise specified by a *ContextObject Format*.

B.5 Parsing of HTTP Querystrings

Upon receiving an *OpenURL* request, the *Resolver* may parse and URL-decode the querystring into a set of Key/Encoded-Value pairs. Depending on the type of *OpenURL*

Transport, an *OpenURL Resolver* may encounter three types of keys: *OpenURL* keys, *KEV ContextObject* keys, and foreign keys:

1. All *OpenURL Transports* use *OpenURL* keys (see Section 6). They have a prefix 'url_'.
2. In addition to the *OpenURL* keys, the *Inline OpenURL Transport* also uses keys from the *KEV ContextObject Format* (see Sections 4 and 5.1). These keys are:
   - Administrative keys (prefixed by 'ctx_')
   - Entity keys (prefixed by 'rft_', 'rfe_', 'req_', 'rfr_', 'res_', or 'svc_').
   - Keys from the *KEV Metadata Formats* (prefixed by 'rft.', 'rfe.', 'req.', 'rfr.', 'res.', or 'svc.').
3. Foreign keys (see Section C.2).

Once keys and values have been extracted from the querystring, the acquisition of the *ContextObject* from the querystring may proceed.

Appendix C. Security Considerations and Precautions

C.1 General Security Considerations

By-Reference *Transport*, either of Context object or entities within *ContextObjects*, introduces certain security risks. In particular, the ability for a request to specify arbitrary network-locations from which the *Resolver* is expected to fetch documents raises particular concerns.

OpenURL 1.0 implementers must be aware of several possible attacks that could compromise security.

1. When an *OpenURL Resolver* has privileged access to resources, such as IP-authenticated licensed content, it is possible that an attacker could hijack this access by sending *OpenURL* with by-reference URLs. *Resolvers* should take care not to expose licensed metadata by returning the resulting metadata to unprivileged users.

2. In a cross-site scripting attack, crafted data values are used to insert code into a webpage seen by a user. This code can be used to insert foreign content or steal personal data, such as authentication data or passwords from a user's "cookie" file.

3. In a vortex or maelstrom attack, By-Reference URLs and the data they return might be constructed in such a way that an endless series of requests is generated. Where possible, *Resolvers* may need to recognize protocols and services which that might generate such endless loops and prevent their propagation.

C.2 Precautions for OpenURL Resolvers

By design, there are no requirements on the behavior of *OpenURL* resolving systems. However, in recognition that a network of *OpenURL Resolvers* must perform robustly and securely even in the presence of potentially malicious attack or inadvertent misuse, this Standard recommends that *Resolvers* take precautions to avoid behaviors that might be exploited to destabilize a digital information environment.

An example of such a precaution is the following:

1. The *OpenURL Resolver* does not fetch metadata, whether for *By-Reference Metadata* or for *By-Reference Transport* of *ContextObjects*, from URLs that can be identified as *OpenURLs*. In addition, if a network-location URL is redirected to a URL that can be identified as an *OpenURL*, the *Resolver* does not follow the redirect.

2. The *OpenURL Resolver* does not fetch metadata from a network location specified in an *OpenURL* with an invalid version string.

References

[1] ANSI/NISO Z39.88-2004. The OpenURL Framework for Context-Sensitive Services

[2] Registry for the OpenURL Framework. http://www.openurl.info/registry/

[3] OpenURL Syntax Description, Draft version 0.1.
http://www.openurl.info/registry/docs/pdf/openurl-01.pdf

[4] Dublin Core Community Profile (DCCP) for Simple Dublin Core in KEV.
http://www.openurl.info/registry/docs/pdf/DublinCoreProfile.pdf