Dublin Core Community Profile (DCCP) for

Simple Dublin Core in KEV Format

DCCP Profile Version: 0.6
Date: 2003-12-16

Introduction

The OpenURL Framework for Context-Sensitive Services Standard 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. 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 Dublin Core Community Profile (DCCP) provides support for a community that uses the Dublin Core metadata element set. This document defines the Registry entries to which that Profile subscribes.

Additional information on the Dublin Core Metadata Initiative can be found at http://dublincore.org.

Related Documents

This Profile must be understood in the context of the OpenURL Framework, as defined in NISO Z39.88-2004. In particular, Part 2 of that standard defines the KEV Context Object Format, which is used by the DCCP.

Purpose and Scope

This Profile will allow the transport of metadata defined with the fifteen Dublin Core elements as Key/Value pairs on an OpenURL. It is intended to provide an OpenURL capability for users of Dublin Core metadata that is a simple transformation from the Dublin Core metadata.

Maintenance of the Dublin Core Community Profile

All OpenURL Community Profiles are registered with the registration agency at http://openurl.info/registry/. This Registry holds the Registry entries that the DCCP subscribes to: Serialization, Character Encoding, Constraint Language, ContextObject Format, Metadata Format and Transport.
The identifier for this Profile in the Registry is:

info:ofi/pro:dccp

The Registry also contains a machine-readable version of this Profile at:

http://www.openurl.info/registry/docs/pro/info:ofi/pro:dccp

The ContextObject and Entities

A ContextObject can contain six different entity types, listed in the table below. Of these entity types, only the Referent is required in the DCCP ContextObject. Detailed

description of the entities is contained in Part 1 of the Z39.88-2004 standard.

Entity	Definition	Prefix	Mandatory/ Optional	Example
Referent	The Entity about which the ContextObject was created – a referenced resource	rft	M	A referenced web document
ReferringEntity	The Entity that references the Referent	rfe	O	A referring web site
Requester	The Entity that requests services pertaining to the Referent	req	O	The user clicking an OpenURL
ServiceType	The Entity that defines the type of service requested	svc	O	Full text
Resolver	The Entity at which a request for services is targeted	res	O	An OpenURL linking server
Referrer	The Entity that generated the ContextObject	rfr	O	Originating resource

Example: Referent (rft)

```
&rft_val_fmt=info:ofi/fmt:kev:mtx:dc
&rft.title=Dr. Jane Smith's Home Page
&rft.identifier=http://example.edu/~jsmith
&rft.language=en-US
&rft.creator=Jane M. Smith
&rft.publisher=mailto:webmaster@university.edu
&rft.description=Bibliography, CV, some fulltext
articles, current courses in Computer Science.
&rft.rights=Copyright 2003 Jane Smith
```

Example: ReferringEntity (rfe)

```
&rfe_val_fmt=info:ofi/fmt:kev:mtx:dc
&rfe.title=Example University Faculty Home Pages
```

Example: Requester (req)

```
&req_id=mailto:jj@example.edu
```

Example: ServiceType (svc)

```
&svc.identifier=http://www.example.edu/service1
```

Example: Resolver (res)

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

```
http://openurl.example.edu/links?
```

Example: Referrer (rfr)

```
&rfr_id=http://example.edu/
```

Registry Entries

The ContextObject Format

The ContextObject of this Profile is represented as a string of ampersand-delimited Key/Encoded-Value pairs, constrained using the Z39.88-2004 MTX matrix as described in Part 2 of the standard.
i.e.

&rft.title=China space craft 'returns safely'
&rft.creator=CNN Online

There are two types of Keys in the KEV ContextObject Format:

- Keys whose names denote a pair indicating both the Entity and a Descriptor method. For example, the key '`rft_id`' denotes an Identifier Descriptor for a Referent. These keys are described in Section 7.1.2 of the Part 2 document.
- Metadata Keys of a KEV Metadata Format used for By-Value Metadata. These have a mandatory prefix that indicates the Entity they describe. For example, the metadata key '`rft.date`' could denote the 'date of publication' for a Referent.

All Keys in the KEV format follow these rules:

- A Key must be separated from its associated Value by an equals character ('=')
- Key/Encoded-Value pairs must be concatenated using the ampersand ('&') character to form a single string
- All Values of Key/Encoded-Value pairs must be URL-encoded so that the ContextObject is 'transport ready' (hence the term 'Encoded-Value'). URL-encoding is described in Part 2 of the Z39.88 document.
- The encoding of characters in Values must either be UTF-8 encoded Unicode or specified using the '`ctx_enc`' key and an info:ofi/enc Identifier for a character encoding. The `ctx_enc` Key is described in Part 2 of the Z39.88 document.

Transports

The metadata in the DCCP can be transported in one of three ways:

- By-value: the metadata for the referent is contained within the ContextObject
- By-Reference: the ContextObject contains the network location of the metadata
- Inline: the ContextObject is transported as part of a query string

The Registry entries for the transports available to this Profile are:

info:ofi/tsp:http:openurl-by-val
info:ofi/tsp:http:openurl-by-ref
info:ofi/tsp:http:openurl-inline

Metadata Format

The DCCP subscribes to the following registered metadata format:

info:ofi/fmt:kev:mtx.dc

This metadata format consists of the fifteen Dublin Core metadata entities. This metadata format will be used for the referent, but could also be used for any other entities where appropriate.

Character Encodings

DCCP ContextObjects can transport metadata coded in either Unicode UTF-8 or in ISO

8859-1 (Latin-1). The Registry entries for these character encodings are:

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

Namespaces

The ContextObject can contain identifiers that make use of URI and URN namespaces. The namespaces available to DCCP, as registered in the OpenURL Registry, are:

Registry entry	Identifier in Registry
Namespace for “ftp” URI Scheme	info:ofi/nam:ftp:
Namespace for “http” URI Scheme	info:ofi/nam:http:
Namespace for “ldap” URI Scheme	info:ofi/nam:ldap:
Namespace for “mailto” URI Scheme	info:ofi/nam:mailto:
Namespace for "urn" URI Scheme	info:ofi/nam:urn:
Namespace for "info:sid" URI Scheme	info:ofi/nam:info:sid:

OpenURL Transport

OpenURL Keys

The following keys are defined for ContextObjects that are transported as OpenURLs:

`url_ver`: OpenURL signature

- Format: fixed value ‘Z39.88-2004’
- Character set and character encoding: Value is US-ASCII
- Example: `url_ver=Z39.88-2004`

`url_tim`: Datetime of the creation of the OpenURL

- Format: ISO8601-conformant datetime, in the YYYY-MM-DD or YYYY-MM-DDTHH:MM:SSZ representation
- Character set and character encoding: Value is US-ASCII and may need URL-encoding
- Example (not URL-encoded for readability): `url_tim=2002-08-16T17:23:45Z`

`url_ctx_fmt`: Identifier of the ContextObject Format used for the representation of the transported ContextObject(s)

- The KEV ContextObject Format is `info:ofi/fmt:kev:mtx:ctx`
- Character set and character encoding: Value is US-ASCII and may need URL-encoding
- Example (not URL-encoded for readability):
  `url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx`

`url_ctx_val`: The actual representations of ContextObjects by means of a registered ContextObject Format

- Dependency: Requires `url_ctx_fmt`
- Format: Representations of ContextObjects using a registered ContextObject Format. The Value of the `url_ctx_val` OpenURL Key is the actual representation of ContextObjects
- Character set and character encoding: The character set and character encoding of the Value is the Character Encoding applied by the ContextObject Format that is

used to represent the transported ContextObject(s). In the KEV ContextObject Format the default Character Encoding is `info:ofi/enc:UTF-8`, while other Character Encodings may be specified as the Value of the `ctx_enc` Key. However, because the Values of Keys in the KEV ContextObject Format are URL-encoded, the representation of a ContextObject provided as the Value of the `url_ctx_val` OpenURL Key is US-ASCII. When provided on a By-Value OpenURL, the Value of the `url_ctx_val` OpenURL Key may need URL-encoding.

- Example (not URL-encoded for readability): `url_ctx_val= rft_id=http://www.arxiv.org/pdf/physics/0311121`

The constraints on OpenURL Keys for the OpenURL Transports are:

Transport	Key	Constraint
By-Value	url_ver	1
	url_tim	≤1
	url_ctx_fmt	1
	url_ctx_val	1
Inline	url_ver	1
	url_tim	≤1
	url_ctx_fmt	≤1
By-Reference	url_ver	1
	url_tim	≤1
	url_ctx_fmt	1
	url_ctx_ref	1

Example: Inline OpenURL Using HTTP Get

This example shows an Inline OpenURL that transports a ContextObject consisting of simple Dublin Core metadata.

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

```
http://www.mimas.ac.uk/metadata/dcopenurl?
      url_ver= Z39.88-2004
      &url_tim=2003-10-16T17:23:45Z
      &url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
      &rft_val_fmt=info:ofi/fmt:kev:mtx:dc
      &rft.title=jstor business
      &rft.subject=business
```

Example: By-Value OpenURL Using HTTP Get

This example shows a By-Value OpenURL using HTTP Get that transports a ContextObject that consisting of simple Dublin Core metadata describing a web page.

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

```
http://www.example.net/menu?
      url_ver= Z39.88-2004
      &url_tim=2003-10-16T17:23:45Z
      &url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
      &url_ctx_val=rft_val_fmt=info:ofi/fmt:kev:mtx:dc
      &rft.title=Dr. Jane Smith's Home Page
      &rft.identifier=http://example.edu/~jsmith
      &rft.language=en-US
      &rft.creator=Jane M. Smith
      &rft.publisher=mailto:webmaster@university.edu
      &rft.description=Bibliography, CV, some fulltext articles,
      current courses in Computer Science.
      &rft.rights=Copyright 2003 Jane Smith
      &rfr_id=http://example.edu
```

Note that the By-Value OpenURL gets double URL encoding, first of the KEV ContextObject and then of the URL:

```
http://www.example.net/menu?
      url_ver=Z39.88-2004&url_tim=2003-10-
      16T17%3E23%3E45Z&url_ctx_fmt=info%3Eofi%2Ffmt%3Ekev%3Emtx%3
      Ectx&url_ctx_val=rft_val_fmt%3Dinfo%253Aofi%252Ffmt%253Akev
      %253Amtx%253Adc%26rft.title%3DDr.%2520Jane%2520Smith's%2520
      Home%2520Page%26rft.identifier%3Dhttp%253A%252F%252Fexample
      .edu%252F%257Ejsmith%26rft.language%3Den-
      US%26rft.creator%3DJane%2520M.%2520Smith%26rft.publisher%3D
      mailto%253Awebmaster%2540university.edu%26rft.description%3
      DBibliography,%2520CV,%2520some%2520fulltext%2520articles,%
      2520current%2520courses%2520in%2520Computer%2520Science.%26
      rft.rights%3DCopyright%25202003%2520Jane%2520Smith%26rfr_id
      %3Dhttp%253A%252F%252Fexample.edu
```

Example: By-Value OpenURL Using HTTP Post

This example shows a By-Value OpenURL using HTTP Post that transports a ContextObject consisting of simple Dublin Core metadata describing a document.

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

```
base URL : http://www.example.net/menu
POST http://www.example.net/menu HTTP/1.0
Content-Length: 284
Content-Type: application/x-www-form-urlencoded

      url_ver= Z39.88-2004
      &url_tim=2003-10-16T17:23:45Z
      &url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
      &url_ctx_val=&rft_val_fmt=info:ofi/fmt:kev:mtx:dc
      &rft.title= The combinatorics of resource sharing
      &rft.creator=V. C. Barbosa
      &rft.publisher=Kluwer
      &rft.identifier=cs.OS/0309044
      &svc_val_fmt=info:ofi/fmt:kev:mtx:sch_svc
      &svc.fulltext=yes
```

Example: By-Reference OpenURL Using HTTP Get

This example shows a By-Reference OpenURL that contains a reference to a representation of a ContextObject consisting of simple Dublin Core metadata describing a bibliography entry.

.

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

```
http://www.example.net/menu?
      url_ver= Z39.88-2004
      &url_tim=2003-10-16T17:23:45Z
      &url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx
      &url_ctx_ref= http://example.edu/~jsmith/bib/metadata-1
```

Appendix A – Profile Summary

OpenURL Framework Component	Registry Entry	Identifier in Registry
ContextObject format		
	KEV ContextObject Format	info:ofi/fmt:kev:mtx:ctx
Serialization		
	Key/Encoded-Value (KEV)	info:ofi/fmt:kev
Character encodings		
	ISO Latin 1	info:ofi/enc:ISO-8859-1
	UTF-8 Unicode	info:ofi/enc:UTF-8
Constraint language		
	Z39.88-2004 Matrix	info:ofi/fmt:kev:mtx
Metadata format		
	KEV Format for Simple Dublin Core	info:ofi/fmt:kev:mtx:dc
Transports	By-Value OpenURL over HTTP	info:ofi/tsp:http:openurl-by-val
	By-Reference OpenURL over HTTP	info:ofi/tsp:http:openurl-by-ref
	Inline OpenURL over HTTP	info:ofi/tsp:http:openurl-inline
Identifiers		
	Namespace for “ftp” URI Scheme	info:ofi/nam:ftp:
	Namespace for “http” URI Scheme	info:ofi/nam:http:
	Namespace for “ldap” URI Scheme	info:ofi/nam:ldap:
	Namespace for “mailto” URI Scheme	info:ofi/nam:mailto:
	Namespace for "urn" URI Scheme	info:ofi/nam:urn:
	Namespace for "sid" URI Scheme	info:ofi/nam:info:sid:

Appendix B – Machine-Readable Profile

```
 <?xml version="1.0" encoding="UTF-8"?>
<profile xmlns="info:ofi/pro" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="info:ofi/pro
http://www.openurl.info/registry/docs/info:ofi/fmt:xml:xsd:pro">
        <registry-identifier>info:ofi/pro:dc</registry-identifier>
        <name>Dublin Core Community Profile</name>
        <context-object-format>
                <context-object minOccurs="1" maxOccurs="1">
                        <referent minOccurs="1" maxOccurs="1">
                                <identifier minOccurs="0" maxOccurs="unbounded"/>
                                <by-value-metadata minOccurs="0" maxOccurs="1"/>
                                <by-reference-metadata minOccurs="0" maxOccurs="1"/>
                                <private-data minOccurs="0" maxOccurs="1"/>
                        </referent>
                        <referring-entity minOccurs="0" maxOccurs="1">
                                <identifier minOccurs="0" maxOccurs="unbounded"/>
                                <by-value-metadata minOccurs="0" maxOccurs="1"/>
                                <by-reference-metadata minOccurs="0" maxOccurs="1"/>
                                <private-data minOccurs="0" maxOccurs="1"/>
                        </referring-entity>
                        <requester minOccurs="0" maxOccurs="1">
                                <identifier minOccurs="0" maxOccurs="unbounded"/>
                                <by-value-metadata minOccurs="0" maxOccurs="1"/>
                                <by-reference-metadata minOccurs="0" maxOccurs="1"/>
                                <private-data minOccurs="0" maxOccurs="1"/>
                        </requester>
                        <service-type minOccurs="0" maxOccurs="1">
                                <identifier minOccurs="0" maxOccurs="unbounded"/>
                                <by-value-metadata minOccurs="0" maxOccurs="1"/>
                                <by-reference-metadata minOccurs="0" maxOccurs="1"/>
                                <private-data minOccurs="0" maxOccurs="1"/>
                        </service-type>
                        <resolver minOccurs="0" maxOccurs="1">
                                <identifier minOccurs="0" maxOccurs="unbounded"/>
                                <by-value-metadata minOccurs="0" maxOccurs="1"/>
                                <by-reference-metadata minOccurs="0" maxOccurs="1"/>
                                <private-data minOccurs="0" maxOccurs="1"/>
                        </resolver>
                        <referrer minOccurs="0" maxOccurs="1">
                                <identifier minOccurs="0" maxOccurs="unbounded"/>
                                <by-value-metadata minOccurs="0" maxOccurs="1"/>
                                <by-reference-metadata minOccurs="0" maxOccurs="1"/>
                                <private-data minOccurs="0" maxOccurs="1"/>
                        </referrer>
                </context-object>
        </context-object-format>
        <format>
```

```
        <registry-identifier>info:ofi/fmt:kev:mtx:ctx</registry-identifier>
        <name>Key/Encoded-Value ContextObject Format</name>
</format>
<serialization>
        <registry-identifier>info:ofi/fmt:kev</registry-identifier>
        <name>Key/Encoded-Value Physical Representation</name>
</serialization>
<constraint-language>
        <registry-identifier>info:ofi/fmt:kev:mtx</registry-identifier>
        <name>NISO Z39.88-2004 Matrix Constraint Language</name>
</constraint-language>
<character-encodings>
        <character-encoding type="default">
                <registry-identifier>info:ofi/enc:UTF-8</registry-identifier>
                <name>UTF-8 encoded Unicode</name>
        </character-encoding>
        <character-encoding>
                <registry-identifier>info:ofi/enc:ISO-8859-1</registry-identifier>
                <name>ISO Latin 1</name>
        </character-encoding>
</character-encodings>
<metadata-formats>
        <metadata-format>
                <registry-identifier>info:ofi/fmt:kev:mtx:dc</registry-identifier>
                <name>KEV Metadata Format for Simple Dublin Core</name>
        </metadata-format>
</metadata-formats>
<namespaces>
        <namespace>
                <registry-identifier>info:ofi/nam:http:</registry-identifier>
                <name>Namespace for http URI Scheme</name>
        </namespace>
        <namespace>
                <registry-identifier>info:ofi/nam:mailto:</registry-identifier>
                <name>Namespace for mailto URI Scheme</name>
        </namespace>
        <namespace>
                <registry-identifier>info:ofi/nam:ldap</registry-identifier>
                <name>Namespace for ldap URI Scheme</name>
        </namespace>
        <namespace>
                <registry-identifier>info:ofi/nam:ftp</registry-identifier>
                <name>Namespace for ftp URI Scheme</name>
        </namespace>
        <namespace>
                <registry-identifier> info:ofi/nam:info:sid:</registry-identifier>
                <name>Namespace for sid URI Scheme</name>
        </namespace>
</namespaces>
<transports>
        <transport>
```

```
                <registry-identifier>info:ofi/tsp:http:openurl-by-val</registry-
identifier>
                <name>By-Value OpenURL over HTTP</name>
            </transport>
            <transport>
                <registry-identifier>info:ofi/tsp:http:openurl-by-ref</registry-
identifier>
                <name>By-Reference OpenURL over HTTP</name>
            </transport>
            <transport>
                <registry-identifier>info:ofi/tsp:openurl-http:inline</registry-
identifier>
                <name>Inline OpenURL over HTTP</name>
            </transport>
        </transports>
</profile>
```