

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: June 14, 2008

M. Nottingham  
December 12, 2007

FIQL: The Feed Item Query Language  
draft-nottingham-atompub-fiql-00

## Status of This Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on June 14, 2008.

## Copyright Notice

Copyright (C) The IETF Trust (2007).

## Abstract

The Feed Item Query Language is a simple but flexible, URI-friendly syntax for expressing filters across the entries in a Web feed. It also specifies a mechanism to allow feeds to indicate what types of queries are supported.

Internet-Draft

FIQL

December 2007

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Notational Conventions . . . . .</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">FIQL Queries . . . . .</a>	<a href="#">4</a>
<a href="#">3.1.</a>	<a href="#">FIQL Expressions . . . . .</a>	<a href="#">4</a>
<a href="#">3.2.</a>	<a href="#">FIQL Constraints . . . . .</a>	<a href="#">4</a>
<a href="#">3.2.1.</a>	<a href="#">Selectors . . . . .</a>	<a href="#">5</a>
<a href="#">3.2.2.</a>	<a href="#">FIQL Comparison Types . . . . .</a>	<a href="#">5</a>
<a href="#">4.</a>	<a href="#">Feed Queries and HTTP . . . . .</a>	<a href="#">9</a>
<a href="#">5.</a>	<a href="#">Feed Extensions for Queries . . . . .</a>	<a href="#">10</a>
<a href="#">5.1.</a>	<a href="#">The fq:interface element . . . . .</a>	<a href="#">10</a>
<a href="#">5.2.</a>	<a href="#">The fq:index element . . . . .</a>	<a href="#">11</a>
<a href="#">6.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">12</a>
<a href="#">7.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">12</a>
<a href="#">8.</a>	<a href="#">Normative References . . . . .</a>	<a href="#">12</a>
<a href="#">Appendix A.</a>	<a href="#">Acknowledgements . . . . .</a>	<a href="#">13</a>
<a href="#">Appendix B.</a>	<a href="#">Default Element Comparison Types . . . . .</a>	<a href="#">14</a>
<a href="#">B.1.</a>	<a href="#">Atom 1.0 . . . . .</a>	<a href="#">14</a>
<a href="#">B.2.</a>	<a href="#">RSS 2.0 . . . . .</a>	<a href="#">14</a>
<a href="#">Appendix C.</a>	<a href="#">Collected ABNF . . . . .</a>	<a href="#">15</a>

## 1. Introduction

The Feed Item Query Language (FIQL, pronounced "fickle") is a simple but flexible, URI-friendly syntax for expressing filters across the entries in a syndicated feed. For example,

```
title==foo*;(updated=lt=-P1D,title==*bar)
```

will return all entries in a feed that meet the following criteria;

- o have a title beginning with "foo", AND
- o have been updated in the last day OR have a title ending with "bar".

This specification defines an extensible syntax for FIQL queries (in [Section 3](#)), explains their use in HTTP ([Section 4](#)), and defines feed extensions for discovering and describing query interfaces ([Section 5](#)).

## 2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [[RFC2119](#)], as scoped to those conformance targets.

This specification uses XML Namespaces [[W3C.REC-xml-names-19990114](#)] to uniquely identify XML element names. It uses the following namespace prefix for the indicated namespace URI;

"fq": "http://purl.org/syndication/query"

This specification uses terms from the XML Infoset [[W3C.REC-xml-infoset-20040204](#)]. However, this specification uses a shorthand; the phrase "Information Item" is omitted when naming Element Information Items. Therefore, when this specification uses

the term "element," it is referring to an Element Information Item in Infoset terms.

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [[RFC4234](#)], including the DIGIT rule from the core. Additionally, the unreserved, gen-delims, and pct-encoded rules are included from [[RFC3986](#)], and QName from [[W3C.REC-xml-names-19990114](#)]. The complete syntax is collected in [Appendix C](#).

Although they refer to Atom [[RFC4287](#)] normatively, the mechanisms described herein can be used with similar syndication formats, such as the various flavours of RSS.

### [3.](#) FIQL Queries

A FIQL query's input is a string of Unicode characters in the form of an expression.

The output of an FIQL query is a feed that has the same head section as the input feed, but that only contains entries that match the given expression (that is, those that yield True).

#### [3.1.](#) FIQL Expressions

An FIQL expression is composed of one or more constraints, related to each other with Boolean operators.

FIQL expressions yield Boolean values: True or False.

```
expression = [ "(" ]
              ( constraint / expression )
              [ operator ( constraint / expression ) ]
              [ ")" ]
operator    = ";" / ","
```

- o ";" is the Boolean AND operator; it yields True for a particular entry if both operands evaluate to True, otherwise False.
- o "," is the Boolean OR operator; it yields True if either operand evaluates to True, otherwise False.

By default, the AND operator takes precedence (i.e., it is evaluated before any OR operators are). However, a parenthesised expression

can be used to change precedence, yielding whatever the contained expression yields.

### [3.2.](#) FIQL Constraints

A FIQL constraint is composed of a selector, which identifies a portion of an entry's content, and an optional comparison/argument pair, which refines the constraint. When processed, a constraint yields a Boolean value.

```
constraint = selector [ comparison argument ]
selector   = 1*( unreserved / pct-encoded )
comparison = ( ( "=" 1*ALPHA ) / fiql-delim ) "="
argument    = 1*arg-char
arg-char    = unreserved / pct-encoded / fiql-delim / "="
fiql-delim  = "!" / "$" / "'" / "*" / "+"
```

#### [3.2.1.](#) Selectors

A selector identifies the portion of an entry that a constraint applies to. When evaluated against an entry in the feed, it returns one or more node-sets [[W3C.REC-xpath-19991116](#)] from the entry, whether they be elements, attributes or textual content.

By default, a selector is treated as an XML QName [[W3C.REC-xml-names-19990114](#)] which selects any and all child elements of the entry element which share the same syntax (i.e., the same prefix and localname; the namespace URI is not considered), along with their descendant content if any.

For example, given the entry

```
<entry xmlns="http://www.w3.org/2005/Atom"
  xmlns:ex="http://other.example.net/">
  <title>Test</title>
  <ex:foo>
    <bar/>
  </ex:foo>
  <ex:foo>
```

```
<baz/>
</ex:foo>
</entry>
```

- o "title" selects the text "Test".
- o "ex:foo" selects both the "bar" and "baz" elements.

This behaviour can be overridden by explicitly defining a selector in the feed; see [Section 5](#).

### [3.2.2](#). FIQL Comparison Types

Constraints with comparison operators are evaluated according to the comparison type associated with the selector. This can be determined by (in order of precedence):

1. An explicit type associated with it by the feed (see [Section 5](#)), or
2. A default type defined by the specification of the selected entry-level metadata element, or
3. A default type for the entry-level metadata element, as specified in [Appendix B](#), or
4. Falling back to the simple textual type (see [Section 3.2.2.1](#)).

A constraint with no comparison operator will yield True if the selector matches any node.

This specification defines a number of comparison types below, and allows extension types to be identified with URIs.

#### [3.2.2.1](#). Simple Textual Comparisons

Simple text comparisons allow for case insensitive, non-language-specific searching of the textual content of selected nodes.

```
text-arg    = [ "*" ] 1*arg-char [ "*" ]
```

Two comparison operators are applicable to simple text comparisons;

- o "==" yields True if the string value (as per XPath) of any selected node matches the argument; otherwise False.
- o "!=" yields True if the string value of every selected node does

not match the argument; otherwise False.

A match occurs when the text in question is character-by-character equivalent with the argument, after observing the following rules, in order:

1. Both the argument and the selected node's content MUST have their encodings (percent-encoding and entity encoding, respectively) removed, to produce Unicode strings.
2. Leading and trailing white space MUST be stripped from the selected node's string. White space within the selected node's string MUST be collapsed to single space characters.
3. Both strings MUST have locale-independent case folding applied, as specified in Section 5.18 of [\[unicode\]](#).
4. Both strings MUST have Normalization Form C [\[unicode-norm\]](#) applied.
5. If the argument string begins or ends with an asterisk character ("\*"), it acts as a wild card, matching any characters preceding or following (respectively) that position.

For example, given the entry:

```
<entry>
  <title>Hello world</title>
  <description>Just    starting.</description>
  <author>
    <name>Mark Nottingham </name>
  </author>
  <content type="xhtml">
    This is <html:em>just</html:em> the start.
  </content>
  <!-- other stuff -->
</entry>
```

```
"title==Hello%20World" yields True
"title!=Hello" yields True
"title==Hello*" yields True
"title==hello*" yields True
"author==Mark*" yields True
"author==*Nottingham" yields True
"description==*start*" yields True
"description==*Just*" yields True
```

"description==Just%20starting." yields True  
"content==\*just%20the%20start\*" yields True  
"description=="\*just" yields False

This comparison type can be identified with the URI  
"http://purl.org/syndication/query/simple-text".

#### 3.2.2.2. Date Comparisons

Date comparisons allow both relative and absolute comparison of date-related values in the string-value of selected nodes.

date-arg = dateTime / duration ; as defined in XML Schema Datatypes

Four operators are relevant to date comparisons;

- o "=" yields True if the point in time specified in the argument matches that indicated by the string-value of the selected node; otherwise False.
- o "!=" yields True if the point in time specified in the argument does not match that indicated by the string-value of the selected node; otherwise False.
- o "<=" yields True if the point in time specified in the argument follows that indicated by the string-value of the selected node; otherwise False.
- o "<=" yields True if the point in time specified in the argument follows that indicated by the string-value of the selected node, or is equal to it; otherwise False.
- o ">" yields True if the point in time specified in the argument precedes that indicated by the string-value of the selected node; otherwise False.
- o ">=" yields True if the point in time specified in the argument precedes that indicated by the string-value of the selected node, or is equal to it; otherwise False.

A point in time can be specified in two ways;

- o Absolutely, expressed as an XML Schema dateTime  
[[W3C.REC-xmlschema-2-20010502](http://www.w3.org/TR/2001/REC-xmlschema-2-20010502)].

- o Relatively, expressed as an XML Schema duration



[[W3C.REC-xmlschema-2-20010502](#)]. By default, such arguments are relative to the time that the query is processed.

White space in the string value of the selected node MUST be ignored.

For example, given an entry

```
<entry>
  <title>Hello World</title>
  <updated>2003-12-13T18:30:02Z</updated>
  <!-- other stuff -->
</entry>
```

```
"updated==2003-12-13T18:30:02Z" yields True
"updated=gt=2003-12-13T00:00:00Z" yields True
"updated=lt=2005-01-01T00:00:00Z" yields True
"updated=gt=-P1D12H" yields False
"updated=gt=-P5Y" yields True
```

(assuming processing on July 1st, 2006)

This specification does not define the appropriate behaviour if a selector with a date type returns multiple nodes.

This comparison type can be identified with the URI  
"http://purl.org/syndication/query/date".

### [3.2.2.3](#). Numeric Comparisons

Numeric comparison allows filtering of numeric values.

number-arg = [ "+" / "-" ] 1\*DIGIT [ "." 1\*DIGIT ]

Four operators are relevant to numeric comparisons;

- o "=" yields True if the string-value of the selected node is numerically equal to the argument; otherwise False.
- o "!=" yields True if the string-value of the selected node is not numerically equal to the argument; otherwise False.
- o "=lt=" yields True if the string-value of the selected node evaluates as numerically less than the argument; otherwise, False.
- o "=le=" yields True if the string-value of the selected node evaluates as numerically less than the argument, or as equal to it; otherwise, False.
- o "=gt=" yields True if the string-value of the selected node evaluates as numerically greater than the argument; otherwise, False.

- o `"=ge="` yields True if the string-value of the selected node evaluates as numerically greater than the argument, or as equal to it; otherwise, False.

White space in the string-value of selected node MUST be ignored.

For example, given an entry

```
<entry>
  <title>Hello World</title>
  <x:foo>123</x:foo>
  <x:bar> 456</x:foo>

  "x:foo==123" yields True
  "x:foo==123.00" yields True
  "x:foo!=123.1" yields True
  "x:foo<=200" yields True
  "x:bar==456" yields True
  "x:foo>=500" yields False
```

This specification does not define the appropriate behaviour if a selector with a numeric type returns multiple nodes.

This comparison type can be identified with the URI `"http://purl.org/syndication/query/numeric"`.

#### [4.](#) Feed Queries and HTTP

Although FIQL can be used in many contexts, it is optimised and intended for use in the query component [[RFC3986](#)] of an HTTP [[RFC2616](#)] URI. E.g.,

```
http://example.com/feed.atom?query=title==*new*,author==bob*
http://example.org/feed.rss?title==*great*;ex:rating=gt=4
```

Note that a FIQL query can be used as the entire query component, or as a sub-component. In the latter case, it is important to account for any encoding conventions to be used with the sub-component; for example, if queries are generated from HTML forms, the sub-component will not be a bare FIQL query, but instead an encoded one.

The output of a HTTP query SHOULD be returned in the HTTP response body, with an appropriate media type. HTTP resources SHOULD return a 400 Bad Request status code if an FIQL expression includes an unsupported or unknown selector.

## 5. Feed Extensions for Queries

Although FIQL is suitable for use with unmodified Web feeds, it is often desirable to override default comparison types, and to advertise what item elements are available for queries.

This can be accomplished using a top-level feed metadata extension, `fq:interface`. For example,

```
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:ex="http://other.example.net/"
  xmlns:fq="http://purl.org/syndication/query">
  <title>example</title>
  <link href="http://www.example.org/" />
  <id>http://example.org/</id>
  <updated>2006-09-12T12:28:02Z</updated>
  <fq:interface
    xmlns:fiql="http://purl.org/syndication/query"
    template="/feed-search?{fiql-exp}">
    <fq:index name="title"/>
    <fq:index name="ex:foo"
      type="http://purl.org/syndication/query/date"/>
    <fq:index name="ex:bar"
      type="http://purl.org/syndication/query/numeric"/>
  </fq:interface>
  <entry>
    <title>test entry</title>
    <description>something...</description>
    <id>http://example.org/1124</id>
    <ex:foo>2003-12-13T18:30:02Z</ex:foo>
    <ex:bar>15.4</ex:bar>
  </entry>
</feed>
```

Here, the `fq:interface` element indicates that queries can be made at the URI template `"http://www.example.org/feed-search?{fiql-exp}"`, where `{fiql-exp}` is replaced with the desired FIQL expression. Expressions will support the selectors `"title"`, `"ex:foo"` and `"ex:bar"`, and the latter two will be compared as a date and number,

respectively.

### [5.1.](#) The fq:interface element

The fq:interface element describes an interface that processes queries for the feed it occurs in. One or more fq:interface elements MAY occur in a feed's head section.

It MUST have a "template" attribute, whose content is a URI template,

as defined in [[I-D.gregorio-uritemplate](#)]. This template SHOULD have exactly one variable, "fiql-exp", that indicates where a FIQL expression should occur. If it does not have a "fiql-exp" variable, it MUST have some other variable that indicates how to submit queries (e.g., using another query language).

For example, given a feed with the fq:interface declaration:

```
<fq:interface template="http://www.example.com/search?{fiql-exp}"/>
```

someone desiring to search for all entries with a title would use this URI:

```
http://www.example.com/search?title
```

fq:interface MAY have any number of fq:index child elements. Additional extension elements and attributes MAY also occur.

### [5.2.](#) The fq:index element

The fq:index element indicates a selector that is available for use in queries. Optionally, it can override its default type, and refine it with an XPath expression.

It MUST have a "name" attribute, containing a selector (see [Section 3.2](#)) to be used to reference the indexed element.

It MAY have a "path" attribute, containing an XPath [[W3C.REC-xpath-19991116](#)] expression that yields the selected node(s) when evaluated with an entry's element as the context node.

It MAY have a "type" attribute, containing a URI indicating the

comparison type that will be applied to that selector.

For example,

```
<fq:interface template="http://www.example.com/search?{fiql-exp}">
  <fq:index name="ex:foo"
    type="http://purl.org/syndication/query/text"/>
  <fq:index name="foo-case-insensitive" path="ex:foo"
    type="http://example.org/comparison-types/case-insensitive"/>
</fq:interface>
```

Here, the `ex:foo` element can be compared using the simple textual algorithm described in [Section 3.2.2.1](#) with the `"ex:foo"` selector, or using a (here, fictional) case-insensitive extension selector `"foo-case-insensitive"`.

Or,

```
<fq:interface template="http://www.example.com/search;{fiql-exp}">
  <fq:index name="foo-num" path="ex:foo/ex:bar/@num"
    type="http://purl.org/syndication/query/numeric"/>
</fq:interface>
```

Here, queries using the selector `"foo-num"` will be evaluated against the content(s) of the `"num"` attribute of the `"ex:bar"` child element of `"ex:foo"`.

By default, `fq:index` is an empty element; however, comparison types MAY define extension elements and attributes.

## [6.](#) Security Considerations

Servers processing queries should be aware of the potential resource and security issues of allowing arbitrarily long and complex queries. Nothing in this specification requires queries to be successfully processed; e.g., a HTTP response may respond with a 403 Forbidden status code if they believe a request to be a security risk.

## [7.](#) IANA Considerations

This memo has no registration requirements.

## 8. Normative References

- [I-D.gregorio-uritemplate] Gregorio, J., "URI Template", [draft-gregorio-uritemplate-00](#) (work in progress), October 2006.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](#), January 2005.
- [RFC4234] Crocker, D. and P. Overell,

Nottingham

Expires June 14, 2008

[Page 12]

---

Internet-Draft

FIQL

December 2007

- "Augmented BNF for Syntax Specifications: ABNF", [RFC 4234](#), October 2005.
- [RFC4287] Nottingham, M. and R. Sayre, "The Atom Syndication Format", [RFC 4287](#), December 2005.
- [W3C.REC-xml-infoset-20040204] Cowan, J. and R. Tobin, "XML Information Set (Second Edition)", W3C REC REC-xml-infoset-20040204, February 2004.
- [W3C.REC-xml-names-19990114] Bray, T., Hollander, D., and A. Layman, "Namespaces in XML", W3C REC REC-xml-names-19990114, January 1999.

[W3C.REC-xmlschema-2-20010502]	Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes", W3C REC REC-xmlschema-2-20010502, May 2001.
[W3C.REC-xpath-19991116]	Clark, J. and S. DeRose, "XML Path Language (XPath) Version 1.0", W3C REC REC-xpath-19991116, November 1999.
[unicode]	The Unicode Consortium, "The Unicode Standard, Version 4.1.0", 2003, < <a href="http://www.unicode.org/versions/Unicode4.1.0">http://www.unicode.org/versions/Unicode4.1.0</a> >.
[unicode-norm]	Davis, M. and M. Duerst, "Unicode Normalization Forms", 10 2006, < <a href="http://www.unicode.org/reports/tr15/">http://www.unicode.org/reports/tr15/</a> >.

## [Appendix A](#). Acknowledgements

Thanks to Wendell Craig Baker, Dave Beckett, Jason Douglas, Tom Gordon, Hugo Haas, John Nienart, Addison Phillips, Pasha Sadri, Jayavel Shanmugasundaram, Tex Texin, Evan Torrie, and Chris Westin for their suggestions. The author takes all responsibility for errors and omissions.

## [Appendix B](#). Default Element Comparison Types

Below are a selection of default comparison types for existing elements.

### [B.1](#). Atom 1.0

- o atom:author - simple text ([Section 3.2.2.1](#))
- o atom:category - simple text ([Section 3.2.2.1](#))
- o atom:content - simple text ([Section 3.2.2.1](#))
- o atom:contributor - simple text ([Section 3.2.2.1](#))

- o atom:id - simple text ([Section 3.2.2.1](#))
- o atom:link - simple text ([Section 3.2.2.1](#))
- o atom:published - date ([Section 3.2.2.2](#))
- o atom:rights - simple text ([Section 3.2.2.1](#))
- o atom:source - simple text ([Section 3.2.2.1](#))
- o atom:summary - simple text ([Section 3.2.2.1](#))
- o atom:title - simple text ([Section 3.2.2.1](#))
- o atom:updated - date ([Section 3.2.2.2](#))

## [B.2.](#) RSS 2.0

- o title - simple text ([Section 3.2.2.1](#))
- o link - simple text ([Section 3.2.2.1](#))
- o description - simple text ([Section 3.2.2.1](#))
- o author - simple text ([Section 3.2.2.1](#))
- o category - simple text ([Section 3.2.2.1](#))
- o comments - simple text ([Section 3.2.2.1](#))
- o enclosure - simple text ([Section 3.2.2.1](#))
- o guid - simple text ([Section 3.2.2.1](#))
- o pubDate - date ([Section 3.2.2.2](#))
- o source - simple text ([Section 3.2.2.1](#))

## [Appendix C.](#) Collected ABNF

```
expression = [ "(" ]
              ( constraint / expression )
```



```

        [ operator ( constraint / expression ) ]
        [ ")" ]
operator    = ";" / ","
constraint  = selector [ comparison argument ]
selector    = 1*( unreserved / pct-encoded )
comparison  = ( ( "=" 1*ALPHA ) / fiql-delim ) "="
argument    = 1*arg-char
arg-char    = unreserved / pct-encoded / fiql-delim / "="
fiql-delim  = "!" / "$" / "'" / "*" / "+"
text-arg    = [ "*" ] 1*arg-char [ "*" ]
date-arg    = dateTime / duration ; as defined in XML Schema Datatypes
number-arg  = [ "+" / "-" ] 1*DIGIT [ "." 1*DIGIT ]

```

#### Author's Address

Mark Nottingham

EMail: [mnot@mnot.net](mailto:mnot@mnot.net)

URI: <http://www.mnot.net/>

## Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

Nottingham

Expires June 14, 2008

[Page 16]