

Internet-Draft  
[draft-ietf-urn-ietf-07.txt](#)  
Expires in six months

Ryan Moats  
AT&T  
October 1998

A URN Namespace for IETF Documents  
Filename: [draft-ietf-urn-ietf-07.txt](#)

## Status of This Memo

This document is an Internet-Draft. 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.''

To learn the current status of any Internet-Draft, please check the ``[1id-abstracts.txt](#)'' listing contained in the Internet-Drafts Shadow Directories on [ftp.is.co.za](#) (Africa), [nic.nordu.net](#) (Europe), [munnari.oz.au](#) (Pacific Rim), [ftp.ietf.org](#) (US East Coast), or [ftp.isi.edu](#) (US West Coast).

## Abstract

A system for Uniform Resource Names (URNs) must be capable of supporting new naming systems. As an example of proposing a new namespace, this document proposes the 'ietf' namespace. This namespace consists of the RFC family of documents (RFCs, STDs, FYIs, and BCPs) developed by the IETF and published by the RFC Editor, the minutes of working groups (WG) and birds of a feather (BOF) meetings that occur during IETF conferences, and the Internet Drafts published by the Internet Drafts Editor. Both the current URN framework and URN syntax support this namespace.

### [0.1](#) Changes from -02

This document has been restructured to use the template proposed in [draft-ietf-urn-nid-reg-03.txt](#). Example Perl scripts for resolving this namespace have been supplied in an Appendix.

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

---

## [0.2](#) Changes from -03

Text was added to the template clarifying persistence and uniqueness. Support for the BCP document series added to the examples and the ABNF.

## [0.3](#) Changes from -04

More clarifications for the persistence section. In addition, cleaned up some typos and ensured that the Appendix examples are functional

## [0.4](#) Changes from -05

Based on a number of requests, i-ds have been added to the structure and resolvers. In addition, some examples have been added.

## [0.4](#) Changes from -06

Some changes of the template to conform with [draft-ietf-urn-nid-req-06.txt](#). Minor editing of some of the sections for more clarity.

## [1](#). Introduction

This document proposes the "ietf" namespace, which consists of the RFC family of documents (RFCs, STDs, FYIs, and BCPs) developed by the IETF and published by the RFC editor and the minutes of working groups (WG) and birds of a feather (BOF) meetings that occur during IETF conferences.

The namespace specification is for a formal namespace.

## [2](#). Specification Template

Namespace ID:

"ietf" requested.

Registration Information:

Registration version number: 1

Registration date: 1998-10-09

Declared registrant of the namespace:

Ryan Moats  
jayhawk@att.com

Expires 4/30/99

[Page 2]

---

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

#### Declaration of structure:

The identifier has the following ABNF [\[2\]](#) specification:

```
NSS = (family ":" number) / ("mtg-" number "-" wgbofname)
      ( "id:" name)
family = "rfc" / "std" / "fyi" / "bcp"
number = 1*DIGIT
wgbofname = 1*LETDIGIT
name = id draft name beginning with "draft-" and trailing
      type removed (e.g. ietf-urn-ietf-06)
LETDIGIT = DIGIT / %x41-%x5a / %x61-%x7a
DIGIT = %x30-%x39
```

If the IESG (or its successor) adds a new document series, this ABNF specification will need to be updated. Further, if a working group or BOF is created that used characters outside the range of this ABNF specification, this specification will need to be updated. Any system intended to resolve names for this namespace should be written with the awareness that this could occur at any time.

#### Identifier uniqueness considerations:

Because the rfc-editor assigns the RFC number uniquely these URNs are unique. Since the mapping between RFCs and other rfc-editor document series (STDs, FYIs or BCPs) is not necessarily one-to-one, uniqueness of STDs, FYIs and BCPs are defined based on the document mappings maintained by the RFC Editor (the index files "rfc-index.txt", "fyi-index.txt", "bcp-index.txt", "std-index.txt") are defined to be the definitive statement of the assignment of RFC Family URNs in this namespace. The meeting minutes portion of the namespace is guaranteed unique because the URN

includes the sequence number of the IETF conference. The document mapping maintained by the Internet Drafts editor ("lid-abstracts.txt") is defined as the definitive statement of the assignment of URNs for the internet draft portion of this namespace.

#### Identifier persistence considerations:

Persistence of the URNs of this namespace is independent of the mutability of the underlying documents. A URN once assigned will never be reassigned to a different resource; the assignment is

Expires 4/30/99

[Page 3]

---

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

persistent and immutable. Distinct from this, the resources identified as RFCs are immutable, whereas the other sorts of documents, STDs, FYIs, and BCPs are not. They may be composites of one or more RFCs and the set of RFCs that includes them may change with time. It is important to note that this mutability of some resources is independent of the immutability of URN assignment to a resource.

#### Process of identifier assignment:

Assignment of URNs from this namespace occurs in three ways. The first is through publication of a new RFC, FYI, STD or BCP is by the RFC Editor. This new document will have a new series number and will therefore define a new URN. The document mappings maintained by the RFC Editor (the index files "rfc-index.txt", "fyi-index.txt", "bcp-index.txt" and "std-index.txt") are defined to be the definitive statement of the assignment of RFC Family URNs in this namespace.

The second way a URN is assigned is through the filing of meeting minutes by a working group or birds of a feather as part of an IETF conference. The list of minutes maintained by the IETF for each working group and conference in the subtree pointed at by the URL <ftp://ietf.org/ietf/> is considered the definitive assignment of URNs for working group or birds of a feather minutes.

The third way a URN is assigned is through the publication of a new internet-draft by the Internet Draft Editor. This draft will have a distinct name (and version number) and therefore defined a new URN. The document mapping maintained by the Internet Drafts editor ("lid-abstracts.txt") is defined as the definitive statement of the assignment of URNs for this portion of the namespace.

Process of identifier resolution:

A mirrored copy of the underlying documentation is required to resolve these URNs. Resolution via HTTP is done by a set of simple Perl cgi-bin scripts presented in [Appendix A](#).

Rules for Lexical Equivalence:

Expires 4/30/99

[Page 4]

---

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

The entire URN is case-insensitive.

Conformance with URN Syntax:

There are no additional characters reserved.

Validation mechanism:

None specified.

Scope:

Global.

### [3](#). Examples

The following are examples of URNs that a resolver for this namespace can resolve:

```
urn:ietf:rfc:2141
urn:ietf:std:50
urn:ietf:id:ietf-urn-ietf-06.txt
```

#### [4. Security Considerations](#)

Because this namespace defines no additional reserved characters, it does not add any security considerations beyond those inherent from the existence of the reserved characters from [\[1\]](#). Further, none of the reserved characters from [\[1\]](#) are used in the definition of the NSS. This means that resolvers for this namespace may be considered "secure" in the sense that any escaping of characters in the NSS MUST result in the resolver indicating that the URN has incorrect syntax.

#### [5. Acknowledgments](#)

Thanks to various members of the URN working group for comments on earlier drafts of this document. The work described in this document is partially supported by the National Science Foundation, Cooperative Agreement NCR-9218179.

#### [6. References](#)

Request For Comments (RFC) and Internet Draft documents are available from <URL:ftp://ftp.internic.net> and numerous mirror sites.

[1] R. Moats, "URN Syntax," [RFC 2141](#), May 5, 1997.

Expires 4/30/99

[Page 5]

---

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

[2] D. Crocker, P. Overell, "Augmented BNF for Syntax Specifications: ABNF," [RFC 2234](#), November 1997.

#### [7. Author's Address](#)

Ryan Moats  
AT&T  
15621 Drexel Circle  
Omaha, NE 68135-2358  
USA

Phone: +1 402 894-9456  
EMail: jayhawk@att.com

#### [Appendix A. Example Resolution Scripts](#)

The following scripts are examples that can be used for resolving URNs in this namespace.

#### [A.1](#) I2C

```
#!/usr/local/bin/perl

use strict;

#
# this is a URN 2 URC resolver for the ietf namespace
#

my(%cite) = (
    bcp => "/ftp/rfc/bcp-index.txt",
    fyi => "/ftp/fyi/fyi-index.txt",
    id => "/ftp/internet-drafts/lid-abstracts.txt",
    rfc => "/ftp/rfc/rfc-index.txt",
    std => "/ftp/std/std-index.txt"
);

my(%number2date) = (
    41 => "98apr",
    40 => "97dec", 39 => "97aug", 38 => "97apr",
    37 => "96dec", 36 => "96jun", 35 => "96mar",
    34 => "95dec", 33 => "95jul", 32 => "95apr",
    31 => "94dec", 30 => "94jul", 29 => "94mar",
    28 => "93nov", 27 => "93jul", 26 => "93mar",
    25 => "92nov", 24 => "92jul", 23 => "92mar",
    22 => "91nov", 21 => "91jul", 20 => "91mar",
    19 => "90dec" );

my($wgpath) = "/ftp/ietf";
```

Expires 4/30/99

[Page 6]

---

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```
my($urn) = $ENV{'QUERY_STRING'};
my($host) = $ENV{'SERVER_NAME'}; #get my host name for ftp: URLs
my($accept) = $ENV{'HTTP_ACCEPT'}; #this is the "Accept:" HTTP header

(&resolveid($1), exit) if ($urn =~ /urn:ietf:id:(i);
(&resolverfc($1, $2), exit) if ($urn =~ /urn:ietf:(72)/i);
(&resolvemt看($1, $2), exit) if ($urn =~ /urn:ietf:mtg-(9*)-(
120&urn_error("400 Bad Request");
```

```

sub resolvemt {
    my($ietfnum, $sesnam) = @_ ;
    &urn_error("404 Not Found") if (!defined $number2date{$ietfnum});
    my($date)=$number2date{$ietfnum};
    my($link)="$wgpath/$sesnam/$sesnam-minutes-$date.txt";
    if (-f $link) {
        print "Status: 200 OK";
        print "Content-type: text/html";
        print "<HTML><TITLE>Citation for $urn</TITLE>";
        print "<BODY>";
        print "<H1><A HREF="
        print "Minutes of the $sesnam working group from the " . &end($ietfnum)
        . " IETF";
        print "</BODY></HTML>";
        return;
    }
    my($link)="$wgpath/$date/$sesnam-minutes-$date.txt";
    if (-f $link) {
        print "Status: 200 OK";
        print "Content-type: text/html";
        print "<HTML><TITLE>Citation for $urn</TITLE>";
        print "<BODY>";
        print "<H1><A HREF="
        print "Minutes of the $sesnam working group from the " . &end($ietfnum)
        . " IETF";
        print "</BODY></HTML>";
        return;
    }
    &urn_error("404 Not Found");
}

sub end {
    my($inarg)=@_ ;
    return $inarg . "st" if ($inarg =~ /1$/);
    return $inarg . "nd" if ($inarg =~ /2$/);
    return $inarg . "rd" if ($inarg =~ /3$/);
    return $inarg . "th";
}

sub resolverfc {
    my($flag,@bib,$i,$k,$j,$done,@ref);

```

```

my($l,$link);
my($scheme, $value) = @_ ;
$scheme =~ tr/A-Z/a-z/;
if (!defined $cite{$scheme}) {
    &urn_error("404 Not Found");
}

$flag = 0;
open(INPUT, "$cite{$scheme}");
while (<INPUT>) {
    $flag = 1 if (/^0*$value /);
    if ($flag == 1) {
        last if (/^$/);
        chop;
        push @bib,$_;
    }
}

if ($scheme ne "rfc") {
    print "Status: 200 OK";
    print "Content-type: text/html";
    $bib[0] =~ s/^[0-9]*<B>/;
    for ($i=0; $i<=$#bib; $i+=1) {
        last if ($bib[$i] =~ s/./.</B>/);
    }
    for ($i=0;$i<=$#bib;$i+=1) {
        $k=$bib[$i];
        while ($k =~ /(fyi|std|rfc|bcp)([0-9]+)(.*)/i) {
            push @ref,"$1$2";
            $k=$3;
        }
        $done="";
        foreach $j (@ref) {
            next if ($done =~ $j);
            $done .= "$j ";
            $l = $j;
            $l =~ tr/A-Z/a-z/;
            $link=&make_link("$l");
            $bib[$i] =~ s/$j/<A HREF="$link">$j</A>/g;
        }
    }
    print "<HTML><TITLE>Citation for $urn</TITLE>";
    print "<BODY>";
    $link=&make_link("$scheme$value");
    print "<H1><A HREF="
    foreach $i (@bib) {
        print "$i0;
    }
}

```

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

    print "</BODY>0/HTML>0;
  } else {
    print "Status: 200 OK70;
    print "Content-type: text/html70;
    $bib[0] =~ s/^[0-9]*//;
    $j=0;
    for ($i=0; $i<=$#bib; $i+=1) {
      $j += ($bib[$i] =~ s/, "/", <B>"/);
      $j += ($bib[$i] =~ s/", /"</B>, /);
    }
    for ($i=0; $i<=$#bib; $i+=1) {
      $k=$bib[$i];
      while ($k =~ /(fyistdrfc|bcp)([0-9]+)(.*)/i) {
        push @ref, "$1$2";
        $k=$3;
      }
      $done="";
      foreach $j (@ref) {
        next if ($done =~ $j);
        $done .= "$j ";
        $l = $j;
        $l =~ s//g;
        $l =~ tr/A-Z/a-z/;
        $link=&make_link("$l");
        $bib[$i] =~ s/$j/<A HREF="$link">$j</A>/g;
      }
    }
    print "<HTML>0TITLE>Citation for $urn</TITLE>0;
    print "<BODY>0;
    $link=&make_link("$scheme$value");
    print "<H1><A HREF=
    foreach $i (@bib) {
      print "$i0;
    }
    print "</BODY>0/HTML>0;
  }
}

sub make_link {
  my($sc);
  my($inarg)=@_;
  ($sc=$1) if ($inarg =~ /([a-z]*)/);
  return "$sc/$inarg.ps" if (-e "/ftp/$sc/$inarg.ps");
}

```

```

    return "$sc/$inarg.html" if (-e "/ftp/$sc/$inarg.html");
    return "$sc/$inarg.txt";
}

sub urn_error {

```

Expires 4/30/99

[Page 9]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

    my($code) = @_; #store failure code here...

    print "Status:  $code";
    print "Content-type: text/html0HTML>0;
    print "<head><title>URN Resolution: I2C $code</title></head>0;
    print "<BODY>0;
    print "<h1>URN to URC resolution failed for the URN:</h1>0;
    print "<hr><h3>$urn</h3>0;
    print "</body>0;
    print "</html>0;
    exit;
};

sub resolveid {
    my($flag,@bib,$i,$k,$j,$count,@ref);
    my($l,$link, $hdr, $done);
    my($value) = @_;
    my($scheme) = "id";

    open(INPUT, "$cite{$scheme}");
    while (<INPUT>) {
#
# capture record
#
        if ($flag == 1 || /^
            push @bib,$_;
            ($hdr = -1, $count = 0, $flag = 1) if (/^
            $count++ if (/^/);
        }
        if ($count == 1) {
            $hdr = $#bib if ($hdr == -1);
        }
        if ($count == 2) {
            for ($i=0; $i<=$hdr; $i+=1) {
                if ($bib[$i] =~ /<(.*?)>/) {
                    $l = $1;

```

```

if ($l eq "draft-$value.txt" || $l eq "draft-$value.ps") {
    print "Status: 200 OK";
    print "Content-type: text/html";
    print "<HTML>0TITLE>Citation for $urn</TITLE>0;
    print "<BODY>0;
    print "<a href="
    print "<pre>0;
    foreach $i (@bib) {
        print "$i";
    }
    print "</pre>0;
    print "</BODY>0/HTML>0;

```

Expires 4/30/99

[Page 10]

---

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

        exit;
    }
}
}
$flag = 0;
@bib = ();
}
}
&urn_error("404 Not Found");
}

```

## [A.2](#) I2L

```
#!/usr/local/bin/perl
```

```
use strict;
```

```

#
# this is a URN 2 URL resolver for the ietf namespace
#

```

```

my(%pathbase) = (
    rfc => "rfc/rfc",
    fyi => "fyi/fyi",
    std => "std/std",
    bcp => "bcp/bcp",
    id => "internet-drafts/draft-",
);

```

```

my(%number2date) = (
  41 => "98apr",
  40 => "97dec", 39 => "97aug", 38 => "97apr",
  37 => "96dec", 36 => "96jun", 35 => "96mar",
  34 => "95dec", 33 => "95jul", 32 => "95apr",
  31 => "94dec", 30 => "94jul", 29 => "94mar",
  28 => "93nov", 27 => "93jul", 26 => "93mar",
  25 => "92nov", 24 => "92jul", 23 => "92mar",
  22 => "91nov", 21 => "91jul", 20 => "91mar",
  19 => "90dec" );

my($wpath) = "/ftp/ietf";
my($urn) = $ENV{'QUERY_STRING'};
my($host) = $ENV{'SERVER_NAME'}; #get my host name for ftp: URLs
my($accept) = $ENV{'HTTP_ACCEPT'}; #this is the "Accept:" HTTP header

(&resolveid($1), exit) if ($urn =~ /urn:ietf:id:(i);
(&resolverfc($1, $2), exit) if ($urn =~ /urn:ietf:(72)/i);
(&resolvemt看($1, $2), exit) if ($urn =~ /urn:ietf:mtg-(9*)-(

```

Expires 4/30/99

[Page 11]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

120&urn_error("400 Bad Request");

```

```

sub resolvemt看 {
  my($ietfnum, $sesnam) = @_;
  &urn_error("404 Not Found") if (!defined $number2date{$ietfnum});
  my($date)=$number2date{$ietfnum};
  my($link)="$wpath/$sesnam/$sesnam-minutes-$date.txt";
  if (-f $link) {
    print "Status: 302 Moved temporarily";
    print "Location: $link";
    return;
  }
  my($link)="$wpath/$date/$sesnam-minutes-$date.txt";
  if (-f $link) {
    print "Status: 302 Moved temporarily";
    print "Location: $link";
    return;
  }
  &urn_error("404 Not Found");
}

```

```

sub end {

```

```

my($inarg)=@_;
return $inarg . "st" if ($inarg =~ /1$/);
return $inarg . "nd" if ($inarg =~ /2$/);
return $inarg . "rd" if ($inarg =~ /3$/);
return $inarg . "th";
}

sub resolverfc {
    my($flag,@bib,$i,$k,$j,$done,@ref);
    my($l,$link);
    my($scheme, $value) = @_;
    $scheme =~ tr/A-Z/a-z/;
    &urn_error("404 Not Found") if (!defined $pathbase{$scheme});
    my($txttry)="/ftp/$pathbase{$scheme}$value.txt";
    my($pstry)="/ftp/$pathbase{$scheme}$value.ps";
    my($htmltry)="/ftp/$pathbase{$scheme}$value.html";
    MIME_SWITCH: {
        if ($accept =~ /application/postscript/ && -f $pstry) {
            print "Status: 302 Moved temporarily";
            print "Location: http://$host/$pathbase{$scheme}$value.ps";
            last MIME_SWITCH;
        }
        if ($accept =~ /text/html/ && -f $htmltry) {
            print "Status: 302 Moved temporarily";
            print "Location: http://$host/$pathbase{$scheme}$value.html";
            last MIME_SWITCH;
        }
    }
}

```

Expires 4/30/99

[Page 12]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

    }
    if ($accept =~ /text/plain/ && -f $txttry) {
        print "Status: 302 Moved temporarily";
        print "Location: http://$host/$pathbase{$scheme}$value.txt";
        last MIME_SWITCH;
    }
    &urn_error("404 Not Found");
}
}

sub urn_error {
    my($code) = @_; #store failure code here...

    print "Status: $code";
    print "Content-type: text/html<HTML>";
}

```

```

print "<head><title>URN Resolution: I2L $code</title></head>0;
print "<BODY>0;
print "<h1>URN to URL resolution failed for the URN:</h1>0;
print "<hr><h3>$urn</h3>0;
print "</body>0;
print "</html>0;
exit;
}

sub resolveid {
    my($flag,@bib,$i,$k,$j,$done,@ref);
    my($l,$link);
    my($scheme) = "id";
    my($value) = @_;
    $scheme =~ tr/A-Z/a-z/;
    &urn_error("404 Not Found0)if (!defined $pathbase{$scheme});
    my($txttry)="/ftp/$pathbase{$scheme}$value.txt";
    my($pstry)="/ftp/$pathbase{$scheme}$value.ps";
    my($htmltry)="/ftp/$pathbase{$scheme}$value.html";
MIME_SWITCH: {
    if ($accept =~ /application/postscript/ && -f $pstry) {
        print "Status: 302 Moved temporarily0;
        print "Location: http://$host/$pathbase{$scheme}$value.ps0;
        last MIME_SWITCH;
    }
    if ($accept =~ /text/html/ && -f $htmltry) {
        print "Status: 302 Moved temporarily0;
        print "Location: http://$host/$pathbase{$scheme}$value.html0;
        last MIME_SWITCH;
    }
    if ($accept =~ /text/plain/ && -f $txttry) {
        print "Status: 302 Moved temporarily0;
        print "Location: http://$host/$pathbase{$scheme}$value.txt0;

```

```

        last MIME_SWITCH;
    }
    &urn_error("404 Not Found0);
}
}

```

### [A.3](#) I2Ls

```
#!/usr/local/bin/perl

use strict;

#
# this is a URN 2 URLs resolver for the ietf namespace
#

my(@urls);

my(%pathbase) = (
    rfc => "rfc/rfc",
    fyi => "fyi/fyi",
    std => "std/std",
    bcp => "bcp/bcp",
    id => "internet-drafts/draft-"
);

my(%number2date) = (
    41 => "98apr",
    40 => "97dec", 39 => "97aug", 38 => "97apr",
    37 => "96dec", 36 => "96jun", 35 => "96mar",
    34 => "95dec", 33 => "95jul", 32 => "95apr",
    31 => "94dec", 30 => "94jul", 29 => "94mar",
    28 => "93nov", 27 => "93jul", 26 => "93mar",
    25 => "92nov", 24 => "92jul", 23 => "92mar",
    22 => "91nov", 21 => "91jul", 20 => "91mar",
    19 => "90dec" );

my($wgpath) = "/ftp/ietf";
my($urn) = $ENV{'QUERY_STRING'};
my($host) = $ENV{'SERVER_NAME'}; #get my host name for ftp: URLs
my($accept) = $ENV{'HTTP_ACCEPT'}; #this is the "Accept:" HTTP header

(&resolveid($1), exit) if ($urn =~ /urn:ietf:id:(i);
(&resolve1($1, $2), exit) if ($urn =~ /urn:ietf:(72)/i);
(&resolve2($1, $2), exit) if ($urn =~ /urn:ietf:mtg-(9*)-(
120&urn_error("400 Bad Request0);

sub resolve2 {
```

```

&urn_error("404 Not Found") if (!defined $number2date{$ietfnum});
my($date)=$number2date{$ietfnum};
my($link)="$wgpath/$sesnam/$sesnam-minutes-$date.txt";
if (-f $link) {
    $link=~s/^/ftp///;
    my($ftplink)="ftp://$host/$link";
    my($httplink)="http://$host/$link";
    my($glink)="gopher://$host:70/0/$link";
    if ($accept =~ /text/uri-list/) { #look for text/uri-list, otherwise
text/html
        print "Status: 200 OK";
        print "Content-type: text/uri-list";
        print "#$urn";
        print "$ftplink";
        print "$httplink";
        print "$glink";
    }
    if ($accept =~ /text/html/) {
        print "Status: 200 OK";
        print "Content-type: text/html0HTML>";
        print "<head><title>URN Resolution: I2Ls</title></head>";
        print "<BODY>";
        print "<h1>URN $urn resolves to the following URLs:</h1>";
        print "<hr><ul>";
        print "<a href="
        print "<a href="
        print "<a href="
        print "</UL>0/body>0/HTML>";
    }
    return;
}
my($link)="$wgpath/$date/$sesnam-minutes-$date.txt";
if (-f $link) {
    $link=~s/^/ftp///;
    my($ftplink)="ftp://$host/$link";
    my($httplink)="http://$host/$link";
    my($glink)="gopher://$host:70/0/$link";
    if ($accept =~ /text/uri-list/) { #look for text/uri-list, otherwise
text/html
        print "Status: 200 OK";
        print "Content-type: text/uri-list";
        print "#$urn";
        print "$ftplink";
        print "$httplink";
        print "$glink";
    }
    if ($accept =~ /text/html/) {
        print "Status: 200 OK";
        print "Content-type: text/html0HTML>";
    }
}

```

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

        print "<head><title>URN Resolution: I2Ls</title></head>0;
        print "<BODY>0;
        print "<h1>URN $urn resolves to the following URLs:</h1>0;
        print "<hr><ul>0;
        print "<a href=
        print "<a href=
        print "<a href=
        print "</UL>0/body>0/HTML>0;
    }
    return;
}
&urn_error("404 Not Found0);
}

sub resolve1 {
    my($flag,@bib,$i,$k,$j,$done,@ref);
    my($l,$link);
    my($scheme, $value) = @_;
    $scheme =~ tr/A-Z/a-z/;
    &urn_error("404 Not Found0)if (!defined $pathbase{$scheme});
    my($try)="/ftp/$pathbase{$scheme}$value.txt";
    if (-f $try) {
        push(@urls, "http://$host/$pathbase{$scheme}$value.txt");
        push(@urls, "ftp://$host/$pathbase{$scheme}$value.txt");
        push(@urls, "gopher://$host:70/0/$pathbase{$scheme}$value.txt");
    }
    $try="/ftp/$pathbase{$scheme}$value.ps";
    if (-f $try) {
        push(@urls, "http://$host/$pathbase{$scheme}$value.ps");
        push(@urls, "ftp://$host/$pathbase{$scheme}$value.ps");
        push(@urls, "gopher://$host:70/0/$pathbase{$scheme}$value.ps");
    }
    $try="/ftp/$pathbase{$scheme}$value.html";
    if (-f $try) {
        push(@urls, "http://$host/$pathbase{$scheme}$value.html");
        push(@urls, "ftp://$host/$pathbase{$scheme}$value.html");
    }
}

&urn_error("404 Not Found0) if ($#urls == -1);

MIME_SWITCH: {
    if ($accept =~ /text/uri-list/) { #look for text/uri-list, otherwise
text/html

```

```

print "Status: 200 OK";
print "Content-type: text/uri-list";
print "#$urn";
foreach $i (@urls) {
    print "$i";
}

```

Expires 4/30/99

[Page 16]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

    last MIME_SWITCH;
}
if ($accept =~ /text/html/) {
    print "Status: 200 OK";
    print "Content-type: text/htmlHTML";
    print "<head><title>URN Resolution: I2Ls</title></head>";
    print "<BODY>";
    print "<h1>URN $urn resolves to the following URLs:</h1>";
    print "<hr><ul>";
    foreach $i (@urls) {
        print "<LI><A HREF="
    }
    print "</UL>/body>/HTML";
    last MIME_SWITCH;
}
}
}

sub urn_error {
    my($code) = @_; #store failure code here...

    print "Status: $code";
    print "Content-type: text/htmlHTML";
    print "<head><title>URN Resolution: I2L $code</title></head>";
    print "<BODY>";
    print "<h1>URN to URL resolution failed for the URN:</h1>";
    print "<hr><h3>$urn</h3>";
    print "</body>";
    print "</html>";
    exit;
}

sub resolveid {
    my($flag,@bib,$i,$k,$j,$done,@ref);
    my($l,$link);

```

```

my($value) = @_;
my($scheme) = "id";
$scheme =~ tr/A-Z/a-z/;
&urn_error("404 Not Found") if (!defined $pathbase{$scheme});
my($try)="/ftp/$pathbase{$scheme}$value.txt";
if (-f $try) {
    push(@urls, "http://$host/$pathbase{$scheme}$value.txt");
    push(@urls, "ftp://$host/$pathbase{$scheme}$value.txt");
    push(@urls, "gopher://$host:70/0/$pathbase{$scheme}$value.txt");
}
$try="/ftp/$pathbase{$scheme}$value.ps";
if (-f $try) {
    push(@urls, "http://$host/$pathbase{$scheme}$value.ps");

```

Expires 4/30/99

[Page 17]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

    push(@urls, "ftp://$host/$pathbase{$scheme}$value.ps");
    push(@urls, "gopher://$host:70/0/$pathbase{$scheme}$value.ps");
}
$try="/ftp/$pathbase{$scheme}$value.html";
if (-f $try) {
    push(@urls, "http://$host/$pathbase{$scheme}$value.html");
    push(@urls, "ftp://$host/$pathbase{$scheme}$value.html");
}

```

```

&urn_error("404 Not Found") if ($#urls == -1);

```

```

MIME_SWITCH: {
    if ($accept =~ /text/uri-list/) { #look for text/uri-list, otherwise
text/html
        print "Status: 200 OK";
        print "Content-type: text/uri-list";
        print "#$urn";
        foreach $i (@urls) {
            print "$i";
        }
        last MIME_SWITCH;
    }
    if ($accept =~ /text/html/) {
        print "Status: 200 OK";
        print "Content-type: text/html;HTML>";
        print "<head><title>URN Resolution: I2Ls</title></head>";
        print "<BODY>";
        print "<h1>URN $urn resolves to the following URLs:</h1>";
    }
}

```

```

        print "<hr><ul>0;
        foreach $i (@urls) {
            print "<LI><A HREF=
        }
        print "</UL>0/body>0/HTML>0;
        last MIME_SWITCH;
    }
}
}

```

#### [A.4](#) I2Ns

```
#!/usr/local/bin/perl
```

```
use strict;
```

```

#
# this is a URN 2 URNs resolver for the ietf namespace
#

```

```
my(%cite) = (
```

Expires 4/30/99

[Page 18]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

    rfc => "/ftp/rfc/rfc-index.txt",
    fyi => "/ftp/fyi/fyi-index.txt",
    std => "/ftp/std/std-index.txt",
    bcp => "/ftp/rfc/bcp-index.txt"
);

```

```

my(%number2date) = (
    41 => "98apr",
    40 => "97dec", 39 => "97aug", 38 => "97apr",
    37 => "96dec", 36 => "96jun", 35 => "96mar",
    34 => "95dec", 33 => "95jul", 32 => "95apr",
    31 => "94dec", 30 => "94jul", 29 => "94mar",
    28 => "93nov", 27 => "93jul", 26 => "93mar",
    25 => "92nov", 24 => "92jul", 23 => "92mar",
    22 => "91nov", 21 => "91jul", 20 => "91mar",
    19 => "90dec" );

```

```

my($wgpath) = "/ftp/ietf";
my($urn) = $ENV{'QUERY_STRING'};
my($host) = $ENV{'SERVER_NAME'}; #get my host name for ftp: URLs

```

```

my($port) = $ENV{'SERVER_PORT'};
my($accept) = $ENV{'HTTP_ACCEPT'}; #this is the "Accept:" HTTP header

(&resolve1("id", $1), exit) if ($urn =~ /urn:ietf:id:(/i);
(&resolve1($1, $2), exit) if ($urn =~ /urn:ietf:(72)/i);
(&resolve2($1, $2), exit) if ($urn =~ /urn:ietf:mtg-(9*)-(
120&urn_error("400 Bad Request");

sub resolve2 {
    my($ietfnum, $sesnam) = @_;
    &urn_error("404 Not Found") if (!defined $number2date{$ietfnum});
    my($date)=$number2date{$ietfnum};
    my($link)="$wgpath/$sesnam/$sesnam-minutes-$date.txt";
    if (-f $link) {
        if ($accept =~ /text/uri-list/) {
            print "Status: 200 OK";
            print "Content-type: text/uri-list";
            print "#$urn";
            return;
        }
        if ($accept =~ /text/html/) {
            print "Status: 200 OK";
            print "Content-type: text/htmlHTML>";
            print "<head><title>URN Resolution: I2Ns</title></head>";
            print "<BODY>";
            print "<h1>URN $urn resolves to the following URNs:</h1>";
            print "<hr><ul>";
            print "</UL>/body>/HTML>";

```

Expires 4/30/99

[Page 19]

```

        return;
    }
}
my($link)="$wgpath/$date/$sesnam-minutes-$date.txt";
if (-f $link) {
    if ($accept =~ /text/uri-list/) {
        print "Status: 200 OK";
        print "Content-type: text/uri-list";
        print "#$urn";
        return;
    }
    if ($accept =~ /text/html/) {
        print "Status: 200 OK";

```

```

        print "Content-type: text/html0HTML>0;
        print "<head><title>URN Resolution: I2Ns</title></head>0;
        print "<BODY>0;
        print "<h1>URN $urn resolves to the following URNs:</h1>0;
        print "<hr><ul>0;
        print "</UL>0/body>0/HTML>0;
        return;
    }
}
&urn_error("404 Not Found0);
}

sub end {
    my($inarg)=@_;
    return $inarg . "st" if ($inarg =~ /1$/);
    return $inarg . "nd" if ($inarg =~ /2$/);
    return $inarg . "rd" if ($inarg =~ /3$/);
    return $inarg . "th";
}

sub resolve1 {
    my($flag,@bib,$i,$k,$j,$done,@ref);
    my($l,$link);
    my($scheme, $value) = @_;
    $scheme =~ tr/A-Z/a-z/;
    if (!defined $cite{$scheme}) {
        &urn_error("404 Not Found0);
    }

    $flag = 0;
    open(INPUT, "$cite{$scheme}");
    while (<INPUT>) {
        $flag = 1 if (/^0*$value /);
        if ($flag == 1) {
            last if (/^$/);
        }
    }
}

```

```

        chop;
        push @bib,$_;
    }
}

$k=join " ",@bib;

```

```

while ($k =~ /(fyi|std|rfc|bcp)([0-9]+)(.*)/i) {
    $k=$4;
    $a=$2; $b=$3;
    if (($a ne $scheme || $b ne $value) && ($1 !~ /obso/i)){
        $a =~ tr/A-Z/a-z/;
        $b =~ s/^0*//;
        push @ref,"urn:ietf:$a:$b";
    }
}

}

MIME_SWITCH: {
    if ($accept =~ /text/uri-list/) {
        print "Status: 200 OK0;
        print "Content-type: text/uri-list0;
        print "#$urn0;
        foreach $i (@ref) {
            print "$i0;
        }
        last MIME_SWITCH;
    }
    if ($accept =~ /text/html/) {
        print "Status: 200 OK0;
        print "Content-type: text/html0HTML>0;
        print "<head><title>URN Resolution: I2Ns</title></head>0;
        print "<BODY>0;
        print "<h1>URN $urn resolves to the following URNs:</h1>0;
        print "<hr><ul>0;
        foreach $i (@ref) {
            print "<li>$i: Click to resolve using0;
            print "<a href="
            print "<a href="
            print "<a href="
            print "<a href="
            print "<a href="
        }
        print "</UL>0/body>0/HTML>0;
    }
}
}

sub make_link {
    my($sc);

```

```

my($inarg)=@_;
($sc=$1) if ($inarg =~ /([a-z]*)/);
return "/$sc/$inarg.ps" if (-e "/ftp/$sc/$inarg.ps");
return "/$sc/$inarg.html" if (-e "/ftp/$sc/$inarg.html");
return "/$sc/$inarg.txt";
}

sub urn_error {
    my($code) = @_; #store failure code here...

    print "Status: $code";
    print "Content-type: text/html0HTML>0;
    print "<head><title>URN Resolution: I2Ns $code</title></head>0;
    print "<BODY>0;
    print "<h1>URN to URN resolution failed for the URN:</h1>0;
    print "<hr><h3>$urn</h3>0;
    print "</body>0;
    print "</html>0;
    exit;
};

```

## [A.5](#) I2R

```
#!/usr/local/bin/perl
```

```
use strict;
```

```

#
# this is a URN 2 resource resolver for the ietf namespace
#

```

```

my(%pathbase) = (
    rfc => "rfc/rfc",
    fyi => "fyi/fyi",
    std => "std/std",
    bcp => "bcp/bcp",
    id => "internet-drafts/draft-"
);

```

```

my(%number2date) = (
    41 => "98apr",
    40 => "97dec", 39 => "97aug", 38 => "97apr",
    37 => "96dec", 36 => "96jun", 35 => "96mar",
    34 => "95dec", 33 => "95jul", 32 => "95apr",
    31 => "94dec", 30 => "94jul", 29 => "94mar",
    28 => "93nov", 27 => "93jul", 26 => "93mar",
    25 => "92nov", 24 => "92jul", 23 => "92mar",

```

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```
22 => "91nov", 21 => "91jul", 20 => "91mar",
19 => "90dec" );

my($wgpath) = "/ftp/ietf";
my($urn) = $ENV{'QUERY_STRING'};
my($host) = $ENV{'SERVER_NAME'}; #get my host name for ftp: URLs
my($accept) = $ENV{'HTTP_ACCEPT'}; #this is the "Accept:" HTTP header

print "$urn0;
(&resolveid($1), exit) if ($urn =~ /urn:ietf:id:(i);
(&resolve1($1, $2), exit) if ($urn =~ /urn:ietf:(72)/i);
(&resolve2($1, $2), exit) if ($urn =~ /urn:ietf:mtg-(9*)-(
120&urn_error("400 Bad Request0);

sub resolve2 {
    my($ietfnum, $sesnam) = @_;
    &urn_error("404 Not Found0) if (!defined $number2date{$ietfnum});
    my($date)=$number2date{$ietfnum};
    my($link)="$wgpath/$sesnam/$sesnam-minutes-$date.txt";
    if (-f $link) {
        print "Status: 200 OK0;
        print "Content-type: text/plain0;
        open(FILE, "$link");
        while (<FILE>) {
            print $_;
        }
        close FILE;
        return;
    }
    my($link)="$wgpath/$date/$sesnam-minutes-$date.txt";
    if (-f $link) {
        print "Status: 200 OK0;
        print "Content-type: text/plain0;
        open(FILE, "$link");
        while (<FILE>) {
            print $_;
        }
        close FILE;
        return;
    }
    &urn_error("404 Not Found0);
}
```

```

sub end {
    my($inarg)=@_;
    return $inarg . "st" if ($inarg =~ /1$/);
    return $inarg . "nd" if ($inarg =~ /2$/);
    return $inarg . "rd" if ($inarg =~ /3$/);

```

Expires 4/30/99

[Page 23]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

    return $inarg . "th";
}

```

```

sub resolve1 {
    my($flag,@bib,$i,$k,$j,$done,@ref);
    my($l,$link);
    my($scheme, $value) = @_;
    $scheme =~ tr/A-Z/a-z/;
    &urn_error("404 Not Found")if (!defined $pathbase{$scheme});
    my($txttry)="/ftp/$pathbase{$scheme}$value.txt";
    my($pstry)="/ftp/$pathbase{$scheme}$value.ps";
    my($htmltry)="/ftp/$pathbase{$scheme}$value.html";

```

```

MIME_SWITCH: {
    if ($accept =~ /application/postscript/ && -f $pstry) {
        print "Status: 200 OK0;
        print "Content-type: application/postscript0;
        open(FILE, "$pstry");
        while (<FILE>) {
            print $_;
        }
        close FILE;
        last MIME_SWITCH;
    }
    if ($accept =~ /text/html/ && -f $htmltry) {
        print "Status: 200 OK0;
        print "Content-type: text/html0;
        open(FILE, "$htmltry");
        while (<FILE>) {
            print $_;
        }
        close FILE;
        last MIME_SWITCH;
    }
    if ($accept =~ /text/plain/ && -f $txttry) {
        print "Status: 200 OK0;
        print "Content-type: text/plain0;

```

```

        open(FILE, "$txttry");
        while (<FILE>) {
            print $_;
        }
        close FILE;
        last MIME_SWITCH;
    }
    &urn_error("404 Not Found");
}
}

```

```
sub resolveid {
```

Expires 4/30/99

[Page 24]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

my($flag,@bib,$i,$k,$j,$done,@ref);
my($l,$link);
my($scheme) = "id";
my($value) = @_;
$scheme =~ tr/A-Z/a-z/;
&urn_error("404 Not Found")if (!defined $pathbase{$scheme});
my($txttry)="/ftp/$pathbase{$scheme}$value.txt";
my($pstry)="/ftp/$pathbase{$scheme}$value.ps";
my($htmltry)="/ftp/$pathbase{$scheme}$value.html";
MIME_SWITCH: {
    if ($accept =~ /application/postscript/ && -f $pstry) {
        print "Status: 200 OK";
        print "Content-type: application/postscript";
        open(FILE, "$pstry");
        while (<FILE>) {
            print $_;
        }
        close FILE;
        last MIME_SWITCH;
    }
    if ($accept =~ /text/html/ && -f $htmltry) {
        print "Status: 200 OK";
        print "Content-type: text/html";
        open(FILE, "$htmltry");
        while (<FILE>) {
            print $_;
        }
        close FILE;
        last MIME_SWITCH;
    }
}

```

```

    }
    if ($accept =~ /text/plain/ && -f $txttry) {
        print "Status: 200 OK";
        print "Content-type: text/plain";
        open(FILE, "$txttry");
        while (<FILE>) {
            print $_;
        }
        close FILE;
        last MIME_SWITCH;
    }
    &urn_error("404 Not Found");
}
}

sub urn_error {
    my($code) = @_; #store failure code here...

    print "Status: $code";

```

Expires 4/30/99

[Page 25]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

print "Content-type: text/html";
print "<head><title>URN Resolution: I2R $code</title></head>";
print "<BODY>";
print "<h1>URN to URL resolution failed for the URN:</h1>";
print "<hr><h3>$urn</h3>";
print "</body>";
print "</html>";
exit;
}

```

## [A.6](#) I2Rs

```
#!/usr/local/bin/perl
```

```
use strict;
```

```

#
# this is a URN 2 resources resolver for the ietf namespace
#

```

```
my(@urls);
```

```

my(%pathbase) = (
    rfc => "rfc/rfc",
    fyi => "fyi/fyi",
    std => "std/std",
    bcp => "bcp/bcp",
    id => "internet-drafts/draft-"
);

my(%number2date) = (
    41 => "98apr",
    40 => "97dec", 39 => "97aug", 38 => "97apr",
    37 => "96dec", 36 => "96jun", 35 => "96mar",
    34 => "95dec", 33 => "95jul", 32 => "95apr",
    31 => "94dec", 30 => "94jul", 29 => "94mar",
    28 => "93nov", 27 => "93jul", 26 => "93mar",
    25 => "92nov", 24 => "92jul", 23 => "92mar",
    22 => "91nov", 21 => "91jul", 20 => "91mar",
    19 => "90dec" );

my($wgpath) = "/ftp/ietf";
my($urn) = $ENV{'QUERY_STRING'};
my($host) = $ENV{'SERVER_NAME'}; #get my host name for ftp: URLs
my($accept) = $ENV{'HTTP_ACCEPT'}; #this is the "Accept:" HTTP header

(&resolveid($1), exit) if ($urn =~ /urn:ietf:id:()/i);

```

Expires 4/30/99

[Page 26]

INTERNET DRAFT

A URN Namespace for IETF Documents

October 1998

```

(&resolve1($1, $2), exit) if ($urn =~ /urn:ietf:(72)/i);
(&resolve2($1, $2), exit) if ($urn =~ /urn:ietf:mtg-(9*)-(
120&urn_error("400 Bad Request0);

```

```

sub resolve2 {
    my($ietfnum, $sesnam) = @_;
    my(@vers,$i);
    &urn_error("404 Not Found0) if (!defined $number2date{$ietfnum});
    my($date)=$number2date{$ietfnum};
    my($link)="$wgpath/$sesnam/$sesnam-minutes-$date.txt";
    if (-f $link) {
        push(@vers,$link);
    }
    $link="$wgpath/$date/$sesnam-minutes-$date.txt";
    if (-f $link) {

```

```

        push(@vers,$link);
    }
    &urn_error("404 Not Found") if ($#vers== -1);

    print "Status: 200 OK";
    print "Content-type: multipart/alternative; boundary=endpart";
    foreach $i (@vers) {
        print "--endpart";
        if ($i =~ /html$/) {
            print "Content-Type: text/html";
        }
        if ($i =~ /txt$/) {
            print "Content-Type: text/plain";
        }
        if ($i =~ /ps$/) {
            print "Content-Type: application/postscript";
        }
        open(FILE, "$i");
        while (<FILE) {
            print "$_";
        }
        close FILE;
    }
    print "--endpart";
}

sub resolve1 {
    my($flag,@bib,$i,$k,$j,$done,@ref);
    my($l,$link,@vers);
    my($scheme, $value) = @_;
    $scheme =~ tr/A-Z/a-z/;
    &urn_error("404 Not Found") if (!defined $pathbase{$scheme});
    my($try)="/ftp/$pathbase{$scheme}$value.txt";

```

```

    if (-f $try) {
        push(@vers, $try);
    }
    $try="/ftp/$pathbase{$scheme}$value.ps";
    if (-f $try) {
        push(@vers, $try);
    }
    $try="/ftp/$pathbase{$scheme}$value.html";

```

```

if (-f $try) {
    push(@vers, $try);
}
print "Status: 200 OK0;
print "Content-type: multipart/alternative; boundary=endpart0;
foreach $i (@vers) {
    print "--endpart0;
    if ($i =~ /html$/) {
        print "Content-Type: text/html0;
    }
    if ($i =~ /txt$/) {
        print "Content-Type: text/plain0;
    }
    if ($i =~ /ps$/) {
        print "Content-Type: application/postscript0;
    }
    open(FILE, "$i");
    while (<FILE>) {
        print "$_";
    }
    close FILE;
}
print "--endpart0;
}

```

```

sub resolveid {
    my($flag,@bib,$i,$k,$j,$done,@ref);
    my($l,$link,@vers);
    my($scheme) = "id";
    my($value) = @_;
    $scheme =~ tr/A-Z/a-z/;
    &urn_error("404 Not Found0)if (!defined $pathbase{$scheme});
    my($try)="/ftp/$pathbase{$scheme}$value.txt";
    if (-f $try) {
        push(@vers, $try);
    }
    $try="/ftp/$pathbase{$scheme}$value.ps";
    if (-f $try) {
        push(@vers, $try);
    }
}

```

```

$try="/ftp/$pathbase{$scheme}$value.html";

```

```

if (-f $try) {
    push(@vers, $try);
}
print "Status: 200 OK";
print "Content-type: multipart/alternative; boundary=endpart";
foreach $i (@vers) {
    print "--endpart";
    if ($i =~ /html$/) {
        print "Content-Type: text/html";
    }
    if ($i =~ /txt$/) {
        print "Content-Type: text/plain";
    }
    if ($i =~ /ps$/) {
        print "Content-Type: application/postscript";
    }
    open(FILE, "$i");
    while (<FILE) {
        print "$_";
    }
    close FILE;
}
print "--endpart";
}
sub urn_error {
    my($code) = @_; #store failure code here...

    print "Status: $code";
    print "Content-type: text/html";
    print "<head><title>URN Resolution: I2Rs $code</title></head>";
    print "<BODY>";
    print "<h1>URN to URL resolution failed for the URN:</h1>";
    print "<hr><h3>$urn</h3>";
    print "</body>";
    print "</html>";
    exit;
}

```