

Network Working Group  
Request for Comments: 3368  
Category: Standards Track

M. Mealling  
VeriSign, Inc.  
August 2002

## The 'go' URI Scheme for the Common Name Resolution Protocol

### Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

### Abstract

This document defines a URI scheme, 'go:' to be used with the Common Name Resolution Protocol. Specifically it lays out the syntactic components and how those components are used by URI Resolution to find the available transports for a CNRP service. Care should be taken with several of the URI components because, while they may look like components found in other URI schemes, they often do not act like them. The "go" scheme has more in common with the location independent "news" scheme than any other URI scheme.

## Table of Contents

1.	Goals . . . . .	2
2.	Terminology . . . . .	2
3.	Syntax Rules . . . . .	3
3.1	General Syntax . . . . .	3
3.2	ABNF Grammar . . . . .	3
3.3	Special Cases and Default Values . . . . .	4
3.3.1	If There is Only a Server . . . . .	4
3.3.2	If Server is Empty Then server=localhost . . . . .	4
3.3.3	Default Port . . . . .	4
3.4	Encoding Rules . . . . .	4
4.	Transport Independence . . . . .	4
5.	Examples . . . . .	4
6.	Security Considerations . . . . .	5
7.	IANA Considerations . . . . .	5
	References . . . . .	6
A.	Registration Template . . . . .	7
	Author's Address . . . . .	7
	Full Copyright Statement . . . . .	8

## 1. Goals

The two goals of the CNRP [3] URI [1] are to identify both a specific common-name record at a specific server and to identify a possibly dynamic query or entry point into the query process. Since CNRP requires that the ID be a core query term, these two cases can be generalized down to simply specifying a query that contains only the ID of the item.

On first glance it would seem a simple enough exercise to canonicalize the XML encoded query and then insert it into the query portion of the URL. The problem here is that, due to the encoding rules, any remotely complex query will quickly blow out the URI length limitations. The suggested solution is to provide a simplified query syntax that is a subset of what is available via the XML.

## 2. Terminology

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 RFC 2119 [4].

### 3. Syntax Rules

#### 3.1 General Syntax

The CNRP URI comes in two forms. The first form is for talking to a specific server. The second form is for expressing a query that is meant to be sent to several different CNRP services. The following two examples are for pedagogical purposes only. The complete ABNF grammar in Section 3.2 is the only authoritative syntax definition.

```
go://[<host>]?[<common-name>]*[;<attribute>=[<type>,<value>]
```

and

```
go:<common-name>*[;<attribute>=[<type>,<value>]
```

#### 3.2 ABNF Grammar

The full ABNF [2] (certain values are included by reference from RFC 2396 [1]):

```
cnrp-uri      = "go:" (form1 / form2)
form1         = "://" [server] ["?" ((common-name *avpair) / id-req) ]
form2         = common-name *avpair

id-req        = "id=" value
avpair        = ";" attribute "=" [ type "," ] value

server        = // as specified in RFC2396

common-name   = *(unreserved | escaped)
attribute     = *(unreserved | escaped)
value         = *(unreserved | escaped)
type          = *(unreserved | escaped)

unreserved    = // as specified in RFC2396

escaped       = "%" hex hex
hex           = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
               "8" | "9" | "A" | "B" | "C" | "D" | "E" | "F" |
               "a" | "b" | "c" | "d" | "e" | "f"
```

### 3.3 Special Cases and Default Values

#### 3.3.1 If There is Only a Server

In the case where the CNRP URI contains only the server production then the URI identifies a given CNRP server, not any particular query that is to be done. A client can assume that this server will at least answer the 'servicequery' request.

#### 3.3.2 If Server is Empty Then server=localhost

If the 'server' element has no value then its value MUST be assumed to be "localhost".

#### 3.3.3 Default Port

CNRP's well known HTTP transport port is 1096. If the port value portion of the server production is not specified then port 1096 SHOULD be used if the client has no prior knowledge about other ports or transports that the service may support.

### 3.4 Encoding Rules

The common-name, query parameters, and parameter values must be encoded using the UTF-8 encoding scheme [5], and any octet that is not one of the permitted characters per the above grammar MUST instead be represented by a "%" followed by two characters from the <hex> character set above. The two characters give the hexadecimal representation of that octet.

## 4. Transport Independence

As stated in the CNRP protocol specification [3], CNRP is allowed to be expressed over multiple transport protocols with HTTP being mandatory to implement. In the case where a client attempts to resolve a CNRP URI and it knows nothing about the service being referenced in that URI, then it SHOULD use HTTP on the CNRP default port (1096).

## 5. Examples

go:Mercedes%20Benz

This example shows a general query for the common-name "Mercedes Benz". The intent is that the query should be packaged with any client provided defaults and sent to the one or more services that the client has configured to ask.

`go://?Mercedes%20Benz`

This example shows a general query for the common-name "Mercedes Benz" that is sent to the server running on the 'localhost'.

`go://cnrp.foo.com?Mercedes%20Benz;geography=US-ga`

This example shows a query for the common-name "Mercedes Benz" in the geographic area "US-ga" which should be sent to the server found at `cnrp.foo.com`.

`go://cnrp.foo.org?Martin%20J.%20D%C3%BCrst`

This example includes a UTF-8 character encoded using hex escaping. The value encoded is a u-umlaut (a 'u' with two dots over it). This simple query is sent to a server found at `cnrp.foo.org` with no parameters.

`go://cnrp.foo.com?id=5432345`

Here only an id is given which means that this example points directly at a particular common-name record on a particular server. This example would probably be found in a link on a web page of some type.

## 6. Security Considerations

In addition to the security considerations inherent in CNRP itself (see the Security Considerations section of RFC 3367 [3]), the URI mechanism can also be used to retrieve a URI identifying some other site by including just the ID and not the common-name being linked to. I.e., the user may think he/she is being shown the URI currently mapped to the "BMW" common-name but in the case where only the ID is used the actual common-name is not part of the URI, thus making it possible to use a CNRP URI without knowing which common-name it is referring to.

## 7. IANA Considerations

The IANA is asked to register the URL registration template found in Appendix A in accordance with RFC 2717 [6].

## References

- [1] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.
- [2] Crocker, D., "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.
- [3] Popp, N., Mealling, M. and M. Moseley, "Common Name Resolution Protocol (CNRP)", RFC 3367, August 2002.
- [4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [5] The Unicode Consortium, "The Unicode Standard, Version 2.0: Appendix A.2", ISBN 0-201-48345-9, January 1988.
- [6] Petke, R. and I. King, "Registration Procedures for URL Scheme Names", BCP 35, RFC 2717, November 1999.

## Appendix A. Registration Template

URL scheme name: go

URL scheme syntax: Section 3.2

Character encoding considerations: Section 3.4

Intended usage: Section 1

Applications and/or protocols which use this scheme: [3]

Interoperability considerations: None not specified in [3]

Security considerations: Section 6

Relevant publications: [3]

Contact: CNRP Working Group

Author/Change Controller: IESG

### Author's Address

Michael Mealling  
VeriSign, Inc.  
21345 Ridgetop Circle  
Dulles, VA 20170  
US

Phone: (703) 742-0400  
EMail: michael@verisignlabs.com

## Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.



