URL Working Group Charlie Kindel
INTERNET-DRAFT Microsoft Corporation
<draft-kindel-uuid-uri-00.txt> Expires 1 Year from November, 24 1997

The uuid: URI scheme

1. 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 at <ftp://ftp.is.co.za/internet-drafts> (Africa), <ftp://nic.nordu.net/internet-drafts> (Europe), <ftp://munnari.oz.au/internet-drafts> (Pacific Rim), <ftp://ds.internic.net/internet-drafts> (US East Coast), or <ftp://ftp.isi.edu/internet-drafts> (US West Coast).

Distribution of this document is unlimited. Please send comments to the author at <mailto:ckindel@microsoft.com>.

2. ABSTRACT

This memo describes a Universal Resource Identifier (URI) scheme that allows resources to be uniquely named over time and space using Universally Unique Identifiers (UUIDs). A UUID URI is a pure URI, in that it contains no information about location. However due to the uniqueness of UUIDs, a UUID URI is persistent over time, much like URNs. UUID URIs are useful in situations where a unique identifier is required that cannot or should not be tied to a particular physical root namespace (such as a DNS name).

3. TABLE OF CONTENTS

4. INTRODUCTION

A UUID (also known as a GUID, for Globally Unique Identifier), is 128 bits long, and if generated correctly [UUID], is either guaranteed to be different from all other UUIDs/GUIDs generated until 3400 A.D. or extremely likely to be different (depending on the mechanism chosen). UUIDs were originally used in the Network Computing System (NCS) [NCS] and later in the Open Group's Distributed Computing Environment [CAE RPC].

One of the main reasons for using UUIDs is that no centralized authority is required to administer them. As a result, generation on demand can be completely automated, and they can be used for a wide variety of purposes. UUID generation algorithms support very high allocation rates: 10 million per second per machine if you need it, so that they could even be used as transaction IDs. UUIDs have a fixed-size (128-bits) which is reasonably small relative to other alternatives. This fixed, relatively small size lends itself well to sorting, ordering, and hashing of all sorts, storing in databases, simple allocation, and ease of programming in general. The primary disadvantage of UUIDs is that they are difficult for humans to parse.

Universal Resource Identifiers [URI1] based on UUIDs are useful in situations where a unique identifier is required that cannot or should not be tied to a particular physical namespace (such as a DNS name). Many platforms have built-in UUID generators (any system with DCE or COM support, for example). However, for those that do not the source code to the reference implementation UUID generator has been made available as part of the Leach and Salz Internet-Draft [UUID].

5. UUID URI

As mentioned above, a UUID is a 128 bit value. UUIDs can be expressed in either binary or string form. Because URIs are expressed as string, this memo is only concerned with the string representation. A typical UUID in string form looks like this:

```
f81d4fae-7dec-11d0-a765-00a0c91e6bf6
```

To enable the use of UUIDs as URI's within Internet documents this memo defines a simple UUID URI scheme. Using the scheme described in this memo, the above UUID would be represented as:

```
uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
```

The basic structure of a UUID URI is:

```
uuid:<uuid>;<qualifiers>
```

where <uuid> is a string representation of a UUID, which is documented in [UUID]. The optional set of qualifiers are specified as name/value pairs separated by semi-colons. There are currently no qualifiers defined. However, UUID URI parsers should be written to gracefully ignore any characters that appear after the last character of the UUID (UUID strings are always 36 characters long).
UUIDs must be generated using well known algorithms.

6. BNF FOR THE UUID URI

A uuid URI begins with the scheme prefix "uuid" and is defined by the following grammar:

```
uuid-uri      = "uuid:" uuid [qualifiers]
uuid          = 8hex "-" 4hex "-" 4hex "-" 4hex "-" 12hex
qualifiers    = [";" name = value]*
hex           = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" |
                "8" | "9" | "a" | "b" | "c" | "d" | "e" | "f" |
                "A" | "B" | "C" | "D" | "E" | "F"
```

The set of qualifiers and their BNF syntax are currently not defined.

4. SECURITY CONSIDERATIONS

It should not be assumed that UUIDs are hard to guess; they should not

be used as capabilities.

7. REFERENCES

[UUID] P. Leach, R. Salz, 'UUIDs and GUIDs', Internet-Draft, <ftp://ftp.isi.edu/internet-drafts/draft-leach-uuids-guids-00.txt>, February 24, 1997.

[CAE RPC] CAE Specification, 'X/Open DCE: Remote Procedure Call', X/Open Company Limited, Reading, Berkshire, UK (xopen.co.uk), 1994. X/Open Document Number C309. ISBN 1-85912-041-5. <http://www.camb.opengroup.org/tech/dce/download>

[NCA] Lisa Zahn, et. al., 'Network Computing Architecture', Prentice Hall, Englewood Cliffs, NJ, 1990

[MD5] R. Rivest, RFC 1321, "The MD5 Message-Digest Algorithm", 04/16/1992.

[URI1] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax and Semantics", 05 Nov 1997. <http://www.ics.uci.edu/pub/ietf/uri/draft-fielding-uri-syntax-00.txt>

[URI2] T. Berners-Lee. "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web." RFC 1630, CERN, June 1994. <http://ds.internic.net/rfc/rfc1630.txt>

[URL] T. Berners-Lee, L. Masinter, and M. McCahill. "Uniform Resource Locators (URL)." RFC 1738, CERN, Xerox PARC, University of Minnesota, December 1994. <http://ds.internic.net/rfc/rfc1738.txt>

8. AUTHOR'S ADDRESSES

Charlie Kindel One Microsoft Way Redmond, WA 98052-6399, U.S.A.
Fax: +1 (206) 936 7329 Email:
<mailto:ckindel@microsoft.com>

9. APPENDIX A - UUID URNS

Universal Resource Names (URNs) have great promise. However they are not widely implemented at the time of this writing, and it is not clear when implementation will become widespread. However, it is likely that UUIDs will also be useful in some situations where URNs are used. For this reason, this appendix proposes a UUID URN scheme. The syntax is exactly the same as the UUID URI scheme described above

with the exception that the uuid: scheme qualifier be preceded with the urn: qualifier. For example, a UUID URN would look like this:

```
urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6
```


10. APPENDIX B - GENERATING UUIDS

Any system that includes a DCE RPC runtime will provide built-in UUID generation facilities. At the lowest level these facilities are exposed through the RPC library API "uuid_create()".

On Windows systems, there are two equivalent APIs for generating UUIDs. One is exposed by the MS RPC library (UuidCreate()), and the other via the Component Object Model (COM) library (CoCreateGuid()). CoCreateGuid() is simply a wrapper around UuidCreate().

On systems with an RPC runtime, a command line utility named "uuidgen" is provided. On the Windows platform, this tool is available as part of the Win32 Software Development Kit and supports the following command line syntax:

```
usage: uuidgen [-isonvh?]
i - Output UUID in an IDL interface template
s - Output UUID as an initialized C struct
o<filename> - redirect output to a file, specified immediately
after o
n<number> - Number of UUIDs to generate, specified immediately
after n
v - display version information about uuidgen
h,? - Display command option summary
```

The command line syntax may vary from system to system.