

Network Working Group	K. Fall	
Internet-Draft	Intel Research Berkeley	
Intended status: Experimental	S. Burleigh, Ed.	
Expires: October 2, 2009	Jet Propulsion Laboratory	
	A. Doria	
	Lulea University of Technology	
	J. Ott	
	Helsinki University of Technology	
	D. Young	
	March 31, 2009	

[TOC](#)

**The DTN URI Scheme**  
**draft-irtf-dtnrg-dtn-uri-scheme-00**

**Status of this Memo**

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and 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/1id-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 October 2, 2009.

**Copyright Notice**

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## Abstract

This document describes the "dtn" Uniform Resource Identifier (URI) scheme. DTN URIs are used as DTN endpoint identifiers (EIDs).

## Requirements Language

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 \(Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.\)](#) [RFC2119].

---

## Table of Contents

- [1.](#) Introduction
  - [2.](#) "dtn" Scheme Syntax and Rules
    - [2.1.](#) Syntax
    - [2.2.](#) Resolution of DTN Endpoint IDs
  - [3.](#) Examples
    - [3.1.](#) dtn:none
    - [3.2.](#) dtn:http://demmermac.dtn/ping
    - [3.3.](#) dtn:ipn:19.6
    - [3.4.](#) dtn:pop:http://www.dtnrg.org/log/
    - [3.5.](#) dtn:pop:mailto:kscott@mitre.org
    - [3.6.](#) dtn:push:ethernet:00:1e:24:82:c3:df
    - [3.7.](#) dtn:push:tcp:119.61.32.4,1816!next:ipn:19.4
    - [3.8.](#) dtn:flood:sql:true
    - [3.9.](#) dtn:flood:sql:batterylevel<.25
    - [3.10.](#) dtn:exec:c:while(foo < 9){bar();foo++;}
  - [4.](#) IANA Considerations
  - [5.](#) Security Considerations
  - [6.](#) Acknowledgements
  - [7.](#) References
    - [7.1.](#) Normative References
    - [7.2.](#) Informative References
  - [§](#) Authors' Addresses
- 

## 1. Introduction

[TOC](#)

A DTN endpoint identifier (EID) is a type of Uniform Resource Identifier (URI) [\*] designed to meet the requirements for endpoint identification as defined in the Bundle Protocol specification, RFC 5050 [\*]:

\*EIDs are used to identify the source and destination endpoints of a bundle, the endpoint to which bundle status reports pertaining

to the bundle are to be sent, and the current custodian of the bundle.

\*EIDs are used, as needed, to identify other endpoints that are involved in the conveyance of a bundle from its source to its destination. For example, the most recent forwarder of a bundle may be identified by EID in some block of the bundle.

\*The length of the scheme-specific part of an EID MUST NOT exceed 1023 bytes.

Any URI whose scheme-specific part is of no more than 1023 bytes MAY be used as a DTN endpoint identifier, regardless of scheme. However, DTN routers may be configured only to forward bundles to destinations whose EIDs conform to more restrictive rules; in particular, bundles destined for endpoints identified by URIs in schemes other than "dtn" are likely to be unforwardable.

EIDs formed in the "dtn" scheme are specifically designed to be parseable by DTN routers, increasing the likelihood that the bundles to which they are destined may be forwarded.

Additional information about the dtn URI scheme - motivation, genesis, and discussion - can be obtained from <http://www.dtnrg.org>.

Note that this draft represents early thinking on this topic and the scheme described here is very likely to change in many respects as we get further input from DTNRG. The authors welcome further discussion and comments.

---

## 2. "dtn" Scheme Syntax and Rules

[TOC](#)

This section specifies the syntax of dtn URIs, then discusses the resolution of DTN endpoint IDs.

---

### 2.1. Syntax

[TOC](#)

The general syntax of a dtn URI, in ABNF [\*], is:

```
|dtnURI = "dtn:" ("none" / nontrivialSSP)
```

where:

```
|nontrivialSSP = dtnURIelt *("!" dtnURIelt)
```

```
|dtnURIelt = [opname] ":" URI ; URI as defined in RFC 3986 [*]
```

```
|opname = "push" / "pop" / "next" / "flood" / "exec"
```

---

## 2.2. Resolution of DTN Endpoint IDs

[TOC](#)

Whenever the scheme-specific part (SSP) of a "dtn" URI is not "none" - that is, it is non-trivial - this SSP comprises one or more dtn URI elements, each of which comprises an optional operation name followed by a URI. (That is, each non-trivial "dtn" URI's SSP contains at least one embedded URI.) The scheme component of the embedded URI in each dtn URI element identifies the name space from which the name-specific string (NSS) of that URI - the characters following the colon that terminates the scheme component - is drawn and in so doing it indicates the syntax rules governing the NSS.

The scheme component of the embedded URI in a dtn URI element MAY be some name that is not the name of a registered URI scheme. When the scheme component of the embedded URI in a DTN URI element is the name of a registered URI scheme, the syntax of the NSS of that embedded URI MUST conform to the syntax rules defined for the registered URI scheme. Otherwise, determining the applicable syntax rules for the NSS of the embedded URI is an implementation issue.

Each URI element in a "dtn" URI's SSP constitutes a single bundle handling request, identifying some handling that the bundle agent SHOULD perform when it is required to dispatch a bundle destined for the endpoint identified by the "dtn" URI. When the URI comprises multiple URI elements, the handling performed in response to each handling request MUST either (a) be performed after any handling performed in response to all preceding requests and be performed before any handling performed in response to all subsequent requests or (b) not be performed. Note that in some cases the handling requested by a given URI element may be precluded by the performance of handling requested by a preceding URI element, leaving the bundle agent with no alternative but to fail to perform the requested handling.

**Note that the opnames identified in the "dtn" scheme syntax defined above are merely those that are currently proposed, and the handling defined below for each operation is purely notional at this point.**

The handling requested by each URI element of a destination EID is defined as follows:

**push:** When the operation name is "push", the bundle agent MUST regard the URI in the URI element as the convergence layer protocol endpoint to which the bundle is to be sent and SHOULD invoke the services of the applicable convergence layer adapter to send the bundle to that endpoint. This handling is intended to supersede any decision on binding proximate node to convergence layer that the bundle agent might make in the absence of this handling request.

**pop:** When the operation name is "pop", the bundle agent MUST regard the URI in the URI element as the Internet application endpoint to which the payload of the bundle is to be sent and SHOULD invoke Internet services to send the payload of the bundle to that application endpoint. This handling is intended to provide a generalized mechanism for directing bundle payloads to proxies for Internet applications.

**next:**

When the operation name is "next", the bundle agent MUST regard the concatenation of the string "dtn:/" and the URI in the URI element as the DTN endpoint to which the bundle agent is to forward the bundle and SHOULD forward the bundle to that endpoint. This handling is intended to supersede any dispatching decision that the bundle agent might make in the absence of this handling request.

**flood:** When the operation name is "flood", the bundle agent MUST regard the URI in the URI element as a node selection expression and SHOULD (a) deliver the bundle to the administrative endpoint of the node if the state of the node satisfies the selection expression and (b) forward the bundle to all of the node's neighbors other than the one from which the bundle was received. This handling is intended to provide a generalized mechanism for content-based addressing.

**exec:** When the operation name is "exec", the bundle agent MUST regard the URI in the URI element as the definition of a procedure expressed in a programming language and SHOULD execute that procedure in the context of the state of the node. This handling is intended to provide a generalized mechanism for programmable forwarding.

When no operation name is present, the bundle agent MUST regard the concatenation of the string "dtn:/" and the URI in the URI element as the name of the destination of the bundle and SHOULD dispatch the bundle as prescribed in RFC5050.

Following the completion of all performance of handling requested by the URI elements of a destination EID, if the bundle has not yet been either forwarded or delivered then the bundle agent MUST regard the entire destination EID as the name of the destination of the bundle and MUST dispatch the bundle as prescribed in RFC 5050.

The results of any prior handling performed by the bundle agent MAY affect the dispatching of a bundle in implementation-specific ways. Note that one possible use of a succession of URI elements with operation names "next" and "push" in an EID is to provide a loose source route, prescribing the forwarding of a bundle along a specified sequence of network hops.

---

### 3. Examples

[TOC](#)

---

[TOC](#)

### 3.1. `dtm:none`

Citing this endpoint ID, the "null endpoint", as the destination of a bundle causes the bundle to be discarded.

---

### 3.2. `dtm:http://demmermac.dtm/ping`

[TOC](#)

Citing this EID as the destination of a bundle causes the bundle simply to be dispatched: it is delivered to the indicated endpoint ID if the local node is registered at that EID, and otherwise it is forwarded to that endpoint.

---

### 3.3. `dtm::ipn:19.6`

[TOC](#)

Citing this EID as the destination of a bundle again causes the bundle simply to be dispatched, as in example 2. Note that in this case the NSS of the URI embedded in the EID's first URI element is simply the string "19.6".

---

### 3.4. `dtm:pop:http://www.dtnrg.org/log/`

[TOC](#)

Citing this EID as the destination of a bundle causes the payload of the bundle to be sent as an HTTP message to the indicated Web site. Note that this differs from example 2 in that (a) only the payload, not the entire bundle, is transmitted and (b) the protocol used for transmission is HTTP rather than Bundle Protocol. In example 2, the presence of the scheme name "http" identifies the syntax rules governing the name text but does **not** cause the HTTP protocol to be used.

---

### 3.5. `dtm:pop:mailto:kscott@mitre.org`

[TOC](#)

Citing this EID as the destination of a bundle causes the payload of the bundle to be sent as an email message to the indicated email address.

---

[TOC](#)

### **3.6. `dtn:push:ethernet:00:1e:24:82:c3:df`**

Citing this EID as the destination of a bundle causes the bundle to be passed to the Ethernet convergence layer adapter for transmission to the indicated Ethernet address. This implies an expectation that an Ethernet convergence layer adapter of some other DTN node is listening for bundle traffic at this address and will, upon reception of the bundle, pass it up to that node's bundle agent for dispatching.

---

### **3.7. `dtn:push:tcp:119.61.32.4,1816!next:ipn:19.4`**

[TOC](#)

Citing this EID as the destination of a bundle causes the bundle to be passed to the TCP/IP convergence layer adapter for transmission to the indicated port on the indicated host. The implied expectation is that a TCP convergence layer adapter of some other DTN node is listening for bundle traffic at this port on this host and will, upon reception of the bundle, pass it up to that node's bundle agent for dispatching. That bundle agent, recognizing (in an implementation-dependent manner) that the first URI element in the destination EID is inapplicable, will ignore that first URI element and will respond only to the second URI element. The second URI element requests that the bundle agent immediately forward the bundle to the endpoint identified by "dtn::ipn:19.4" rather than perform standard dispatch procedures. That is, this EID prescribes a source route including selection of a specific initial convergence-layer channel.

---

### **3.8. `dtn:flood:sql:true`**

[TOC](#)

Citing this EID as the destination of a bundle causes the bundle to be delivered to the local node's administrative endpoint (because the node's state is satisfied by the selection expression "true") and then forwarded to all of the node's neighbors other than the one from which the bundle was received.

---

### **3.9. `dtn:flood:sql:batterylevel<.25`**

[TOC](#)

Citing this EID as the destination of a bundle causes the bundle to be delivered to the local node's administrative endpoint, provided the "batterylevel" variable in the state of the node has a value less than .25, and then forwarded to all of the node's neighbors other than the one from which the bundle was received.

---

[TOC](#)

### 3.10. `dtm:exec:c:while(foo < 9){bar();foo++;}`

Citing this EID as the destination of a bundle causes function "bar" to be executed so long as the value of node state variable "foo" is less than 9. As soon as "foo" is 9 or greater, all requested handling for this destination endpoint ID is complete; if execution of function "bar" has not yet caused the bundle to be dispatched, the bundle is dispatched at this time. Since, for this purpose, the entire destination EID must be regarded as the name of the destination, the node's routing mechanism might require an a default dispatching decision for all bundles whose destination names conform to the wild-card expression "dtm:exec:\*". This is an implementation matter.

---

## 4. IANA Considerations

[TOC](#)

The IANA has registered the dtm URI scheme as specified in this document and summarised in the following template.

URI scheme name: dtm

Status: permanent

URI scheme syntax: see Section 2

Character encoding considerations: none

Intended usage: see Sections 1 through 3

Applications and/or protocols that use this URI scheme name: Any applications and protocols that use the DTN Bundle Protocol.

Interoperability considerations: none

Security considerations: see Section 5

Relevant publications: RFC 5050

Contact: Kevin Fall(kfall@intel.com) and Stephen Farrell (stephen.farrell@cs.tcd.ie)

Author/Change controller: Kevin Fall and Stephen Farrell

---

## 5. Security Considerations

[TOC](#)

dtm URIs whose URI elements lack operation names or are confined to the operation names "push" and "next" raise no security considerations beyond those addressed in RFC 5050.

The "pop" operation could be used to circumvent firewall rules that accept Bundle Protocol traffic but reject traffic destined for endpoints of the popped Internet application protocol.

Attacks built on the "flood" operation could exploit the possible side effects of evaluating selection expressions.

Attacks built on the "exec" operation could modify bundle node state in a limitless variety of ways.

Bundle protocol implementations that support these more dangerous operations will need to exercise extreme care.

---



## 6. Acknowledgements

[TOC](#)

[We can only have up to 5 authors, so Stephen, Joseph, Elwyn, and others need to be acknowledged here.)

---

## 7. References

[TOC](#)

### 7.1. Normative References

[TOC](#)

[RFC2119]	<a href="#">Bradner, S.</a> , " <a href="#">Key words for use in RFCs to Indicate Requirement Levels</a> ," BCP 14, RFC 2119, March 1997 ( <a href="#">TXT</a> , <a href="#">HTML</a> , <a href="#">XML</a> ).
-----------	--

---

### 7.2. Informative References

[TOC](#)

[InfRef]	"," 2004.
----------	-----------

---

## Authors' Addresses

[TOC](#)

	Kevin Fall
	Intel Research Berkeley
	2150 Shattuck Avenue, #1300
	Berkeley, California 94704
	USA
Phone:	+1-510-495-3014
Fax:	
Email:	<a href="mailto:kfall@intel.com">kfall@intel.com</a>
	Scott Burleigh (editor)
	Jet Propulsion Laboratory
	4800 Oak Grove Drive, m/s 301-490
	Pasadena, California 91109
	USA
Phone:	+1-818-393-3353
Fax:	
Email:	<a href="mailto:scott.c.burleigh@jpl.nasa.gov">scott.c.burleigh@jpl.nasa.gov</a>
URI:	
	Avri Doria
	Lulea University of Technology
	Lulea 971 87

	Sweden
Email:	<a href="mailto:avri@ltu.se">avri@ltu.se</a>
	Joerg Ott
	Helsinki University of Technology
	Department of Communications and Networking
	PO Box 3000
	TKK 02015
	Finland
Email:	<a href="mailto:jo@netlab.tkk.fi">jo@netlab.tkk.fi</a>
	David Young
Phone:	
Fax:	
Email:	
URI:	