The "active" URI scheme

Status of this Memo

Abstract

A new URI scheme, "active", is defined. It allows processing results
to be referenced uniquely and invariantly by compounding their
dependencies into a URI.

Conventions used in this document

   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 [1].

[1](#). **Description**

In a declarative resource processing pipeline it is helpful for all
resources, whether external or internal intermediate results, to be
identified by URI. An _active_ URI may uniquely specify the
resulting resource of a process, such a URI can embody a processing
specification and may be used as a request for a generative process.

This document defines a new URI scheme that allows specification of
a generative resource process. The URIs are of the form:

              active:<processURI>*(+<inputName>@<inputURI>)

The <processURI> is a URI that uniquely identifies a processing
step. Any reserved characters must be escaped.
The <inputName> is an identifier for a named input to the processing
step. It is not necessarily unique within the scope of the URI.
The <inputURI> is a URI that uniquely identifies an input resource
to the processing step. Any reserved characters must be escaped. The
URI may be another _active_ URI, making recursion possible.

Zero or more inputName-inputURI pairs are allowed.

The "active" URI scheme has no relative URI forms.

[2](#). **Syntax**

```
    activeURI   := _active:_ processURI *( input )
    processURI  := absoluteURI | relativeURI
    input       := _+_ inputName _@_ inputURI
    inputName   := *( alphanum )
    inputURI    := absoluteURI | relativeURI
```

where "alphanum" is imported from [[RFC2396](#)] and "absoluteURI" and
relativeURI are imported from [[RFC2396](#)] represented using URL
escaped encoding of [[RFC2396](#)] as necessary.

Character encoding of the active URI MUST force nested _inputURI_s
to escape invalid characters if they are not within the character
set.

[3](#). **Interoperability**

   There are no known interoperability issues.

[4]. Comparisions

   Syntactic similarities can be drawn to the generic URIs query
   component syntax. However the key differences are:
   1) Symantically the inputs are not queries to a resource.
   2) All inputURIs must be URIs, not alphanumeric.
   3) Recursive nesting of URIs must be defined.

[5]. Examples

   An _active_ URI might define an XSLT transform on an XML document:

              active:xslt+stylesheet@foo.xsl+operand@bar.xml

   or conversion to upper-case of a data[RFC2397] URI:

               active:toUpper+operand@data:text/plain,foobar

[6]. History

   This idea was originally prototyped in June 2002 within a working
   implementation of a research project within Hewlett Packard Labs.
   The idea was first used as a way of identifying intermediate results
   in XML pipelines.
   1060 Research acquired the intellectual property from this research
   project in July 2003. The idea has since been refined for use in a
   generalized resource processing pipeline where unique and invariant
   references to derived results is key to an efficient caching
   strategy. _active_ URIs are used as the internal addressing model
   of 1060 NetKernel which is open source and available from:
   http://www.1060.org

[7]. Security
   The active: URI scheme is currently intended for local processing
   only so there are no exceptional security considerations that are
   particular to this URI. Active URI specified processes are no
   different to any other local mechanism for specifying a process. By
   formalizing a request for processing into an active URI it is
   possible to programmatically analyse the request in a structured way
   for possible security issues.
   Security related to any nested URIs should be given the

consideration they would normally have.

8. References

1   Bradner, S., "Key words for use in RFCs to Indicate Requirement
     Levels", BCP 14, RFC 2119, March 1997

[RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
     Resource Identifiers (URI): Generic Syntax", RFC2396, August
     1998.

[RFC2397] Masinter, L., "The data URL scheme", RFC2397, August 1998

Author's Addresses

   Tony Butterfield
   1060 Research Limited
   23 Horse Street
   Chipping Sodbury
   Bristol BS37 6DA
   United Kingdom

   Email: tab@1060.org