NDIIPP Content Transfer Project

California Digital Library
March 27, 2008

The GrabIt Package Exchange Protocol (V0.1)

Abstract

This document specifies GrabIt, an HTTP-based protocol for systematically transferring lists of digital packages. A package is an arbitrary sequence of octets (e.g., a single TAR archive) that can come with an associated content type indicating that extra steps (not central to GrabIt) may need to be taken to complete and validate a package before successful receipt can be reported. The sender issues an HTTP "post" with a list URLs corresponding to packages to "grab". In response, the receiver returns a job URL in an HTTP "Location" header that can be probed periodically to retrieve transfer status reports.

1. Packages, Lists, and Reports

GrabIt is an HTTP-based protocol for a sender to systematically post lists of digital packages for a receiver to "grab", (to pull). A digital *package* is an octet sequence, often the serialization of an aggregated single-file representation of a hierarchy, such as a filesystem tree in TAR or ZIP format. A package list is a sequence of text lines containing triples of the form,

```
ALG:CHECKSUM  LENGTH  PKGID
```

where ALG:CHECKSUM is a digest algorithm name followed by a checksum calculated over the package, LENGTH is its size (number of octets), and PKGID is a URL from which the package is to be fetched. For ALG, implementors are encouraged to support at least two widely used checksum algorithms: "md5" **[RFC1321]** and "sha1" **[RFC3174]**. Examples include: XXX

A package report list is a sequence of text lines containing package list entries preceded by a transfer status code value and an indented line may follow a package triple if received package reference is supported. A package report list entry has the form,

```
STATUSCODE  ALG:CHECKSUM  LENGTH  PKGID
    Package-Claim-URL:  PKGREF
```

where PKGREF is a URL (or "-") if received package reference is supported and STATUSCODE is one of

done
: Transfer operation completed successfully, including all verification implied by checksums and validation type.

failed
: Transfer operation failed or was aborted, and a reason should be given in the transfer details.

running

Transfer operation is in progress.

pending

Transfer operation is waiting to start.

Values in GrabIt package lists and report lists are separated by one or more spaces and any value except PKGID may be left unspecified using "-". The HTTP Content-Type for lists and report lists are "text/x-grabit-list" and "text/x-grabit-reportlist", respectively.

XXX package report may start with fewer lines The GrabIt protocol is essentially about sending lists of package triples and getting back lists of package report entries. A GrabIt *job* is the set of package transfers implied by a single GrabIt request. A job is initiated when a package list to "grab" is accepted by a receiver. The receiver returns a report list URL that can be probed periodically to retrieve updated reports on transfer status. Each report contains an overall job and per-package status information.

Authentication is not addressed by GrabIt. It is assumed that senders and receivers will make their own arrangements and take measures (certificates, SSL, passwords, IP address filtering, etc.) appropriate to their transfer security requirements.

2. Initiating a GrabIt Job

A GrabIt job is initiated by a "post" request to a URL that a receiver has made known to a sender by pre-arrangement outside the scope of GrabIt. learns the presence of "?grabit" in the query string of a URL. The package sender learns of the base URL by arrangement (outside of GrabIt), the sender will have learned that the receiver at that URL is prepared for It is assumed that the receiver at rcvr.example.com maintains a drop box at that location and that the sender has the right to "post" to it. Here is a sample HTTP session.

```
C: POST http://rcvr.example.com/?grabit HTTP/1.1
C: Content-Type: text/x-grabit
C: Content-Length: 71
C:
C: - - http://s.example.org/bar.tar.gz
C: - - http://s.example.org/zaf.tar.gz
S: HTTP/1.1 201 Created
S: Location: http://rcvr.example.com/grabitjob/15893
S: Date: Wed, 02 Apr 2008 04:00:47 GMT
```

S: Content-Type: text/x-grabit-report which asks rcvr.example.com to grab two packages without either checksum or length specified.

XXX do "curl" example. To verify a single package that is included in the "post" payload, specify the checksum as a parameter, as in

3. Package Lists and Package Report Lists

Package lists and package report lists are text files. In HTTP they are accompanied by a Content-Type of "text/x-grabit". A package list has the form

```
ALG:CHECKSUM  LENGTH  URL
```

```
ALG:CHECKSUM  LENGTH  URL
...
```

where URL identifies the package to be fetched, LENGTH is the number of octets in the file (or "-" to leave it unspecified), and ALG:CHECKSUM is a cryptographic checksum calculated over the package serialization (or "-" to leave it unspecified). If a CHECKSUM is specified in a package list, it is an error if the algorithm is not named via an initial "sumtype" parameter. One or more linear whitespace characters (spaces or tabs) separate the three values, and any such characters in the URL must be hex-encoded.

curl --data-binary @file -o outfile --trace fulldumpalldata XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXx

A package report list is created by the receiver for each received package list. It has the form

```
STATUSCODE  CHECKSUM  LENGTH  URL
   [ details ]
STATUSCODE  CHECKSUM  LENGTH  URL
   [ details ]
...
```

DATE is given according to [W3CDTF xxx TEMPER?]. [XXX: Optional headers include: List-Started-Date, List-Finished-Date, and List-Transfer-Rate. ?]

The package report list is essentially an annotated package list; a status code is prepended to each line, optional package-specific details are added on indented lines below the relevant package line, and a few per-list header lines are prepended. For example,

```
List-Report: http://rcvr.example.com/pkgjobs/x2468
GrabIt-Status: running
Transfer-Report-Date: 20080425145543

done - 1234567 http://s.example.org/foo.tar.gz
    Package-Claim-Identifier: http://r.example.com/pkglib/p2459
running - 987654 http://s.example.org/bar.tar.gz
pending - 9753188 http://s.example.org/zaf.tar.gz
```

For the purpose of creating a package report list, a single package included as a "post" is considered to have a URL of "-". So a successful transfer for a minimal GrabIt specification of "?grab=-" would generate a package report containing the line, "done - - -".

The STATUSCODE field is used to summarize the per-list status (line 2) and the per-package status and that starts each package line. It may apply to any job, package list, or package transfer. STATUSCODE values include:

done
: Transfer operation completed successfully, including all verification implied by checksums and validation type.

failed
: Transfer operation failed or was aborted, and a reason should be given in the transfer details.

running

Transfer operation is in progress.

pending

Transfer operation is waiting to start.

Indented lines below a package line convey package transfer details and any comments. After successful transfer of a package, among the details the receiver may place an indented line of the form "Package-Claim-Identifier: PID" to indicate a persistent package reference identifier (e.g., a URL that might support package return).

4. GrabIt Response

The receiver responds to a GrabIt request by promptly returning a text file that is the initial transfer status report. There is no requirement that the transfer be finished and immediate reporting is encouraged. Transfer operations implied by GrabIt jobs often take minutes to hours to complete and reporting early and often is normal. In fact there are no GrabIt responses that are not transfer status reports. For example,

```
XXXX full example with lists of package lists goes here
List-Report: JOBURL
GrabIt-Status: STATUSCODE [ details ]
Transfer-Report-Date: DATE
STATUSCODE  CHECKSUM  LENGTH  URL
   [ details ]
STATUSCODE  CHECKSUM  LENGTH  URL
   [ details ]
...
```

The List-Report header contains a receiver-supported URL that can be accessed by the sender as long as the job is relevant (e.g., active) in order to retrieve the latest transfer status report. "Reloading" the page returned by this URL is the only form of communication from the receiver defined by GrabIt.

5. Optimizing Package Transfer

Optimal network transfer on high-speed wide-area TCP networks tends to require both parallel streams and the ability to size TCP buffers appropriately. Depending on requirements, however, sufficient throughput may be achieved with standard tools run in parallel processes and without either exotic protocols or TCP buffer sizes. In particular, quite adequate results may be obtained using standard tools such as WGET or RSYNC managed by simple wrappers running in multiple backgrounded processes.

6. References

[BAGIT] LC/CDL, “**The Bagit File Package Format**,” 2008 (**HTML**).

[METS] LC, “**Metadata Encoding and Transmission Standard**,” 2007 (**HTML**).

[RFC1321] **Rivest, R.**, “**The MD5 Message-Digest Algorithm**,” RFC 1321, April 1992.

[RFC2822] Resnick, P., “**Internet Message Format**,” RFC 2822, April 2001.

[RFC3174] Eastlake, D. and P. Jones, “**US Secure Hash Algorithm 1 (SHA1)**,” RFC 3174, September 2001.

[RFC3629] Yergeau, F., “**UTF-8, a transformation format of ISO 10646**,” STD 63, RFC 3629, November 2003.

Author's Address

CDL
California Digital Library
415 20th St, 4th Floor
Oakland, CA 94612
US
Fax: +1 510-893-5212