Open eBook™ File Format 1.0

DRAFT VERSION 001
November 5, 1999

Open eBook™ File Format 1.0

DRAFT VERSION 001

November 5, 1999

This is a draft recommendation. Changes will be made in response to further internal and public review. This document is for review, not implementation.

TABLE OF CONTENTS

1 Overview

1.1 Purpose and Scope

The purpose of the Open eBook File Format is to provide a specification for exchange of electronic books. Specifically:

- The specification provides a format in which an electronic publication can be easily and efficiently transferred between content providers, tool providers, booksellers and reading systems as a single file.
- The specification permits security, authentication and digital rights management, but does not specify the details of such systems.
- The specification specifies a means by which content compression may be accomplished.
- The specification reflects established container standards.

The goal of this specification is to provide a structure within which interoperable electronic book systems can operate today and evolve in the future.

This specification is based on the premise that in order for electronic book technology to achieve widespread success in the marketplace, reading systems must have convenient access to a large number and variety of titles.

1.2 About This Version

This specification was developed by two electronic book companies: NuvoMedia and SoftBook Press. While this draft was written in private, it is explicitly our intent that further development proceed in an open process. To that end, we are submitting this draft for review and further action to the Open eBook Initiative.

Authors:
Garth Conboy, SoftBook Press
Aleksey Novicov, SoftBook Press
David Ornstein, NuvoMedia, Inc.
John Rivlin, SoftBook Press
Marc Tarpenning, NuvoMedia, Inc.

1.3 Definitions

Agent

The term "agent" refers specifically to a computer program or computer system that interprets OEB files.

Body

The term "body", when not further qualified means the body of an entity, that is, the body of either a message or of a body part.

Body Part

The term "body part" refers to an entity inside of a multipart entity.

Entity

The term “entity” refers specifically to the MIME-defined header fields and contents of either a message or one of the parts in the body of a multipart entity. The specification of such entities is the essence of MIME. Since the contents of an entity are often called the “body”, it makes sense to speak about the body of an entity.

OEB File
The MIME entity defined by this specification.

OEB Document
An XML document which conforms to the OEB 1.0 Document DTD.

OEB Manifest
The contents of the XML <manifest> element defined by the OEB 1.0 Package DTD.

OEB Package
An XML document which conforms to the OEB 1.0 Package DTD.

OEB Publication
An OEB Package is a collection of OEB 1.0 documents and other files, typically in a variety of media types, including structured text and graphics that constitutes a cohesive unit for publication.

1.4 Relationship to Other Specifications

This specification combines subsets and applications of existing standards. Together, these facilitate the assembly of an OEB publication into a single, optionally compressed file that may be easily exchanged over any medium:

1. the Open eBook Publication Structure 1.0 (http://www.openebook.org/OEB1.html)

2. the Multipurpose Internet Mail Extensions (MIME) specification Part One, Part Two, Part Three, Part Four and Part Five (http://www.ietf.org/rfc/rfc2045.txt, http://www.ietf.org/rfc/rfc2046.txt, http://www.ietf.org/rfc/rfc2047.txt, http://www.ietf.org/rfc/rfc2048.txt, http://www.ietf.org/rfc/rfc2049.txt)

3. the MIME Multipart/Related Content-Type (http://www.ietf.org/rfc/rfc2387.txt)

4. the GZIP file format specification (http://www.ietf.org/rfc/rfc1952.txt)

5. the Content Disposition specification (http://www.ietf.org/rfc/rfc2183.txt)

6. particular MIME media types (http://www.ietf.org/rfc/rfc1738.txt)

1.4.1 Relationship to OEB Publication Structure

The OEB Publication Structure specifies a content format for the creation of electronic books. An OEB publication typically consists of many documents that are referred to by a single OEB package file. The OEB publication definition does not provide a convenient and efficient format for transporting electronic books across networks.

This specification solves this problem. An OEB file gathers the various documents that are part of an OEB publication into a single file.

1.4.2 Relationship to MIME (RFCs 2045, 2046, 2047, 2048, 2049)

An OEB file is a type of Multipurpose Internet Mail Extensions (MIME) entity. MIME has been used extensively as the primary method of transporting complex multi-part entities among heterogeneous systems over the Internet. Consequently it is ideally suited to the task of combining the multiple OEB documents and the additional files that comprise an OEB publication into a single transportable file.

All OEB files are valid multipart MIME entities. The converse is not true. This specification describes the proper subset of MIME that is allowable within OEB files. All conforming agents must be capable of processing this subset of MIME.

Because OEB files are a proper subset of MIME it should be possible to use standard MIME software to aid in the construction and deconstruction of OEB files. This is one of the motivations for selecting MIME as a file format.

1.4.3 Relationship to MIME Multipart/Related Content-Type

The multipart/related MIME type is a generalized mechanism for representing objects that are an aggregation of related MIME body parts. OEB files are an aggregation of OEB documents and related files. They are therefore ideally suited to representation as multipart/related entities. This specification therefore defines OEB files as a particular type of multipart/related MIME entity and conforms to the recommendations of RFC 2387, which defines the specifics of multipart/related MIME entities.

1.4.4 Relationship to GZIP

All OEB file agents must be capable of accepting body parts in the GZIP compression format as defined by RFC 1952.

1.4.5 Relationship to Content Disposition

The Content-Disposition header allows an originating system to encode various attributes of a source file. This header must be specified in OEB files, however it is not required that OEB file agents process this field.

1.4.6 MIME Media Types

This specification defines two new MIME media types that all conforming software must support. Other MIME media types may be used in the context of this specification, but no special processing is defined for them by this specification.

The OEB File Format MIME media types are:

Media Type	Reference	Description
Application/x-oeb1	This specification	Used for multipart/related type parameter
Application/x-gzip	RFC 1952	Used to compress MIME bodies

1.5 Conformance

This section defines conformance for OEB files and OEB file agents.

1.5.1 OEB File Conformance

A byte stream is a conformant OEB file if and only if:

(i) it is a conformant MIME entity of type multipart/related (see http://www.ietf.org/rfc/rfc2049.txt);

(ii) it contains one and only one MIME body part, of type text/xml, that contains an OEB package;

(iii) the body part containing the OEB package is either the first body part of the multipart/related entity or is identified by a Content-ID header with a value corresponding to the value of the start parameter specified for the multipart/related entity.

(iv) it contains one and only one MIME body part for each item referred to in the OEB manifest contained in the OEB package;

(v) each body part, except the body part containing the OEB package, is identified by a "Content-OEB-ID" header field with a value corresponding to its "id" field in the OEB manifest;

1.5.2 OEB File Agent Conformance

An agent that processes an OEB file is conformant if and only if:

(i) it is capable of correctly uncompressing body parts compressed using GZIP

(ii) when confronted with a non-conformant OEB file, a conforming agent must halt processing, indicate that it has encountered a non-conforming OEB file and must not process the contents of the file.

(iii) it performs its designate function when presented with a conformant OEB file;

1.5.3 Compatibility of OEB File Format 1.0 with Future Versions of the OEB Publication Structure

It is the intent of the authors of this specification that an OEB File 1.0 be capable of containing future versions of the OEB publication format. It would be expected that these future versions of the publication format would be identified by subsequently defined values of the multipart/related type parameter, but would otherwise be conformant to this specification.

1.5.4 Compatibility of OEB File Format 1.0 with Future Versions of the OEB File Format

It is the intent of the authors of this specification that subsequent generations of this specification continue in the directions established by the 1.0 release of this specification and the 1.0 release of the OEB Publication Structure. Specifically:

- Content format standards will be compatible with W3C (and IETF) standards;
- Any required functionality not present in relevant official standards shall be defined in a manner consistent with its eventual submission to an appropriate standards body as extensions to existing standards;

2 Overview Of The OEB File

Figure One, illustrates the structure of an OEB file:

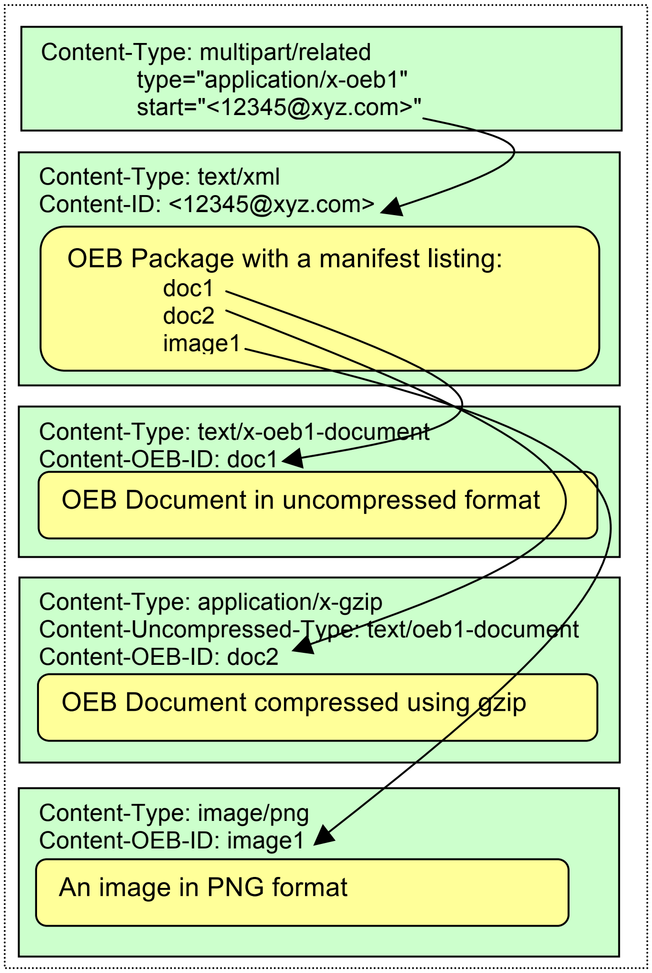
Content-Type: multipart/related
type="application/x-oeb1"
start="<12345@xyz.com>"
Content-Type: text/xml
Content-ID: <12345@xyz.com>
OEB Package with a manifest listing:
doc1
doc2
image1
Content-Type: text/x-oeb1-document
Content-OEB-ID: doc1
OEB Document in uncompressed format
Content-Type: application/x-gzip
Content-Uncompressed-Type: text/oeb1-document
Content-OEB-ID: doc2
OEB Document compressed using gzip
Content-Type: image/png
Content-OEB-ID: image1
An image in PNG format

Figure One - An OEB File

OEB files are a specific type of MIME entity. They are a structured subset of the MIME format. MIME is defined as an extension of RFC 822, which defines the Internet message format. Because OEB files are not envisioned as a type of mail message, OEB file agents may ignore the header fields for mail messages defined in RFC 822. The required header fields for OEB files are those defined for MIME entities within the basic MIME specification (RFC 2045).

If an OEB file is the outermost MIME entity then it must include a “MIME-Version: 1.0” header to identify it as being compliant with the MIME specification (RFC 2045).

A content-type header field of “multipart/related” with a type parameter of “application/x-oeb1” identifies a MIME entity as being an OEB file that conforms to this file specification. The mechanism of extending

MIME through the use of multipart/related is defined by RFC 2387. Note that existing MIME compliant systems that do not recognize multipart/related must treat multipart/related as multipart/mixed, which is non-destructive.

The multipart/related content-type permits a “start” parameter to indicate the root of a compound object. If specified, the start parameter must identify a body part containing the OEB package (root body part). The value of the start parameter must be the “Content-ID” specified in the root body part’s Content-ID field. If omitted, the first body part must contain the OEB package.

The root body part must have a content-type of “text/xml”. This body part must contain an OEB package. Note that this differs from the multipart/related recommendation of RFC 2387 that suggests that the content-type header’s type parameter should match the root body part’s content-type. This deviation from the multipart/related recommendation occurs because using a MIME type of text/xml is not adequately descriptive for a multipart/related type parameter.

In addition to the “text/xml” body part that contains the OEB package definition, there may be zero or more “text/x-oeb1-document” body parts containing OEB documents. Additional body parts identified by other MIME content-type values may also be present within the file.

The mandatory OEB package body part described above will include an OEB manifest. Each “<item>” element present in the OEB manifest will include an “id” attribute. The value of the id attribute provides a unique identifier that allows each MIME body part to be associated with a particular item in the OEB manifest. Specifically, there must be one body part present in the OEB file for each item identified in the OEB manifest. Each body part containing an item from the OEB manifest must contain a “Content-OEB-ID” header field with the id value from the corresponding OEB manifest item.

Additional body parts not referenced in the OEB manifest may be present in an OEB file, however their meaning is not defined by this specification

3 Compression

It is anticipated that electronic book publications may be large. Additionally, document repositories may contain large numbers of publications. As a result, a standard compression method is needed.

To achieve the goal of publication portability, a minimally conformant implementation agent must support the GZIP compression method. GZIP is described in detail in RFC 1952. Body parts using GZIP compression must be identified by a content-type of “application/x-gzip”.

The selection of GZIP is motivated by a number of factors including:

- broad availability across a wide variety of systems
- lack of intellectual property issues
- acceptable compression of text data

A compressed body part that is part of an OEB publication must specify a Content-Uncompressed-Type header field. This will allow an agent to process the body part properly once the decompression operation has occurred.

Compression must be applied individually to documents or other items identified in the OEB manifest. A compressed body must contain a single OEB document or other single item referred to in the OEB manifest. Once a body has been uncompressed, the resultant data can then be processed as if it were of the type specified in the Content-Uncompressed-Type field.

Compression must not be applied to the body part containing the OEB package. This requirement exists to insure processing agents that they can rely on the ability to interpret the OEB package prior to performing any decompression.

Additionally, agents implementing this specification must ignore the file names and other file information contained within the GZIP compression directory. All document naming must be accomplished through the use of the OEB document naming mechanisms specified by the Content-Disposition header.

4 Content-Types

This specification defines the "multipart/related" content-type with a type parameter of "application/x-oeb1". It also specifies support for text/xml, text/x-oeb1-document and application/x-gzip document types. This section defines the specific usage of these content-types.

Encoding of some of the content-types defined by this specification require the existence of specific header fields. In the absence of specification to the contrary, rules for header fields should be taken from the basic MIME specification (RFC 2045) and the multipart/related specification (RFC 2387).

4.1 multipart/related Content-Type

An OEB file is a MIME entity with a content-type of "multipart/related" and a type parameter of "application/x-oeb1". The multipart/related entity may use the start parameter to identify the "root" body part. The root body part is the text/xml body part containing the OEB package describing the OEB publication. Zero or more additional body parts may be included in addition to the OEB package.

A multipart/related entity must contain all body parts necessary to represent the entire OEB publication it contains. Only one OEB publication may be present within a multipart/related entity.

Any of the Content-Transfer-Encoding values permitted by the multipart/mixed content-type may be used in conjunction with the multipart/x-oeb1 content-type.

The start of an OEB file might appear as follows:

```
Content-Type: multipart/related;
      type="application/x-oeb1";
      start= "<12345@xyz.com>";
      boundary="unique-boundary"

--unique-boundary
Content-Type: text/xml
Content-ID: <12345@xyz.com>

<?xml version="1.0"?>
<!DOCTYPE package
  PUBLIC "+//ISBN 0-0673008-1-9//DTD OEB 1.0 Package/EN"
  http://openebook.org/dtds/oeb-1.0/package.dtd>
<package>
…
</package>
```

4.1.1 application/x-oeb1 type parameter

The type parameter of the multipart/related entity specifies the type of object contained within the multipart/related entity. The parameter must be present with a value of “application/x-oeb1”.

4.1.2 Start parameter

The multipart/related content-type permits a “start” parameter to indicate the root of the compound object it contains. If specified, the start parameter must identify the body part containing the OEB package describing the OEB publication. The value of the start parameter must be the “Content-ID” specified in the root body part’s Content-ID field. If omitted, the first body part must contain the OEB package describing the OEB publication.

4.2 text/xml Content-Type containing the OEB Package

A multipart/related entity contains a root body part. The root is either the first body part in the multipart/related entity or is identified by the start parameter.

The root body part of an OEB file must have a content-type of “text/xml”. The XML document must be an OEB package. Only one body part containing an OEB package may be present in each OEB file. All content-transfer-encoding values defined by the basic MIME specification (RFC 2045) are permitted. A body part containing and OEB package might appear as follows:

```
Content-Type: text/xml

<?xml version=”1.0”?>
<!DOCTYPE package
  PUBLIC “+//ISBN 0-9673008-1-9//DTD OEB 1.0 Package//EN”
  http://openebook.org/dtds/oeb-1.0/package.dtd>
<package>
      metadata
      manifest
      spine
      guide
</package>
```

The OEB package body may not be compressed. The restriction allows receiving agents to be guaranteed that they may easily process the OEB package if necessary.

4.3 text/x-oeb1-document Content-Type

The “text/x-oeb1-document” content-type identifies a body part as containing an OEB document. A text/x-oeb1-document must contain a Content-OEB-ID header field if it is to be considered to be part of an OEB publication. All content-transfer-encoding values defined by the basic MIME specification (RFC 2045) are permitted. A text/x-oeb1-document body part might appear as follows:

```
Content-Type: text/x-oeb1-document
Content-OEB-ID: chap1

<?xml version=”1.0”?>
<!DOCTYPE html PUBLIC
  PUBLIC “+//ISBN 0-0673008-1-9//DTD OEB 1.0 Document/EN”
  http://openebook.org/dtds/oeb-1.0/document.dtd>
```

```
<html>
  <body>
    <p>The text of chapter 1</p>
  </body>
</html >
```

4.4 application/x-gzip Content-Type

The “application/x-gzip” content-type identifies a body part as containing compressed data in the gzip format specified by RFC 1952. All OEB file agents must be capable of handling application/x-gzip body parts.

Because gzip is a binary format, the binary or base64 content-transfer-encoding values must be used.

A Content-OEB-ID header must be specified as one of the headers that describes the application/x-gzip body part.

An application/x-gzip body part might appear as follows:

```
Content-Type: application/x-gzip
Content-Uncompressed-Type: text/x-oeb1-document
Content-OEB-ID: chap1
Content-Transfer-Encoding: base64

33429dfkldsjk3949495495469043695kdsjflksdoeoewremrjeioerow
3834893798423skdkk39329323ikekjtlioew94393kejkjwedsi9ds3kk
 …
3i239039dskljdfcv908dskljdsfkjldsfkjldsfkjl3w90e89ewjhgjhg
sd90u39ridskljdsfio8ew9ewiuoeiueriohdihosdgiogiogiogsdiodg
```

4.5 Compressing Other Content Types

In addition to compression of text/x-oeb1-document bodies, it is also possible to compress bodies identified by other content-types. Compressing a JPEG might look as follows:

```
Content-Type: application/x-gzip
Content-Uncompressed-Type: image/jpeg
Content-OEB-ID: pict1
Content-Transfer-Encoding: base64

9823jkfglkjfgkjfgfdgklfdj93497kld89KJDFIdkljfsj9wedfjkljdf
kjsdfio34weiomDLKFJKLIOUEieioewioewlkKLJ938rioiuotr9oudweo
 …
sdf9393KJdmncxijdsfkludsiouewiuioueklfklnfbiri0e89ewjhgjhg
sdfdslfkopeioiewt4weew9ewiuoeiueriohdihosdgiogiogiogsdiodg
```

5 Content-Uncompressed-Type Header Field

The Content-Uncompressed-Type header field allows a compression agent to preserve the original MIME type of a body part. This field must be an exact copy of the content-type field that would have been used if the data had been transferred in an uncompressed binary format.

The Content-Uncompressed-Type header must be specified for all body parts with a content-type of application/x-gzip.

6 Content-OEB-ID Header Field

The purpose of the Content-OEB-ID header field is to identify body parts that have been enumerated in the OEB manifest. Each “<item>” element in the OEB manifest must have an “id” attribute. To identify a body part as representing an item in an OEB manifest, the Content-OEB-ID header must specify the same identifier used in the item element’s id attribute for the corresponding item.

A body part identified by a Content-OEB-ID header field must be present for each item enumerated in the OEB manifest.

The only uniqueness requirement for Content-OEB-ID is that it be unique within the OEB publication they define.

If an OEB manifest were to contain the following item:

```
<item id=”chap1” href=”chapter1.html”
      media-type=”text/x-oeb1-document” />
```

the corresponding body part within its OEB file must contain the following header:

```
Content-OEB-ID: chap1
```

7 Content-Disposition Header Field

The Content-Disposition header field as defined by RFC 2183 specifies a method of suggesting a filename and file attribute to be used on the destination system. RFC 2183 does not define a method for providing a pathname on a destination system. The reason for this apparent omission is the difficulty of specifying names that are guaranteed to be legal in all possible destination environments.

To preserve the document structure of an OEB publication as it migrates from system to system, the href parameter must be used.

Additional parameters defined by RFC 2183 may be specified for OEB files however none are required and a conforming agent is not required to interpret them.

7.1 href Parameter

The OEB package specification requires each item to contain an “href” attribute. The value specified for href may be either an absolute or relative URI pointing to an OEB document. Note that when an OEB file is unpacked, it may not be possible to place documents in the locations specified by absolute URIs.

It is desirable to manage the body parts of an OEB file without dissecting the OEB package body. As a result the href attribute specified in the OEB manifest must be copied to the Content-Disposition header as an href parameter value:

```
<item id=”chap1” href=”arabian_nights/chapter1.html”
      media-type=”text/x-oeb1-document” />
```

the resultant Content-Disposition header would be:

```
      Content-Disposition: inline; href=”arabian_nights/chapter1.html”
```

The href parameter of the Content-Disposition field must specify a value that is identical to the value specified for the item element’s href parameter in the OEB manifest. The href parameter is intended to provide a suggestion for file location. It is not intended to mandate any naming methodology.

8 .oeb File Type

To allow operating software to easily identify files that conform to this specification, OEB files conforming to this specification should use the file extension of .oeb whenever possible.

9 Example

Following is an example of an OEB publication that consists of two OEB documents and two pictures. Only the second OEB document (chap2) is compressed using the gzip format.

```
MIME-Version: 1.0
Content-Type: multipart/related;
      type=”application/x-oeb1”;
      start=”uniqueid.1”;
      boundary=”unique-boundary”

This section could be used to provide a description of what
follows. Processing software ignores it.

--unique-boundary
Content-Type: text/xml
Content-ID: uniqueid.1

<?xml version=”1.0”?>
<!DOCTYPE package
  PUBLIC “+//ISBN 0-0673008-1-9//DTD OEB 1.0 Package/EN”
  http://openebook.org/dtds/oeb-1.0/package.dtd>
<package>
  <metadata>
  …
  </metadata>
  <manifest>
    <item id=”chap1” href=”text/chapter1.html”
      media-type=”text/x-oeb1-document” />
    <item id=”chap2” href=”text/chapter2.html”
      media-type=”text/x-oeb1-document” />
    <item id=”pict1” href=”pictures/pict1.png”
      media-type=”image/png” />
    <item id=”pict2” href=”pictures/pict2.jpeg”
      media-type=”image/jpeg” />
  </manifest>
  <spine>
  …
  </spine>
  <guide>
  …
  </guide>
</package>

--unique-boundary
Content-Type: text/x-oeb1-document
Content-Disposition: href=”text/chapter1.html”
Content-OEB-ID: chap1

<?xml version=”1.0”?>
<!DOCTYPE html PUBLIC
  PUBLIC “+//ISBN 0-0673008-1-9//DTD OEB 1.0 Document/EN”
  http://openebook.org/dtds/oeb-1.0/document.dtd>
<html>
  <body>
    <p>The text of chapter 1</p>
  </body>
</html >

--unique-boundary
Content-Type: application/x-gzip
Content-Uncompressed-Type: text/x-oeb1-document
Content-Disposition: href=”text/chapter2.html”
Content-OEB-ID: chap1
Content-Transfer-Encoding: base64

33429dfkldsjk3949495495469043695kdsjflksdoeoewremrjeioerow
3834893798423skdkk39329323ikekjtlioew94393kejkjwedsi9ds3kk
 …
3i239039dskljdfcv908dskljdsfkjldsfkjldsfkjl3w90e89ewjhgjhg
sd90u39ridskljdsfio8ew9ewiuoeiueriohdihosdgiogiogiogsdiodg

--unique-boundary
Content-Type: image/png
Content-Disposition: href=”pictures/pict1.png”
Content-OEB-ID: pict1
Content-Transfer-Encoding: base64

GhyHhHUujhJhijH77n8weriuerwieor332j4lkejdskfjdsfi3idfkdsfd
SdfiwerewionmmnbuhdsjkW89wekdifeoejkKKLJkdfklsdijdsdsklJks
…
O3iwedsdofoeodfe9ejrekewiureoiwdcmkdsfiemdfeioeKJjdskjsdfs
9efemdse9efeoeie989etdskldsf

--unique-boundary
Content-Type: image/jpeg
Content-Transfer-Encoding: base64
Content-Disposition: href=”pictures/pict2.jpeg”
Content-OEB-ID: pict2

389402lkdfslkjsd8973wrwwerjksdfjsdiouewioKLjdklsjfsdkdfjdk
w9ekksjdsfsdfdfsdfdsfdsweorpXewmYd99nm4etweeerrrewrersdbgS
…
sdooiewsefDJKLSDFPieofsuoiDFKJKDKJDksdfjkldsfkdssdfSdfjkkd
```

```
9efemdse9efeoeie989etdskldsf

--unique-boundary-
```