A MIME Library for Boost

Marshall Clow
mclow@qualcomm.com
marshall@idio.com

What is MIME?

- Multipart Mail Extension
- Defined in RFCs 2045-7, 2049, 4288-9
- Used in SMTP, HTTP, SIP, and other internet protocols
- Self-describing data

```
Resent-To: mail.app@flagg2.qualcomm.com
Resent-From: win-eudora-bugs@flagg2.qualcomm.com
Resent-Message-Id: <B0001796251@flagg2.na.qualcomm.com>
Resent-Date: Mon, 1 Nov 2004 15:38:13 -0800
Received: from sanders.rain.org (maxmp-189.rain.org [198.68.144.189])
   by coyote.rain.org (Postfix) with ESMTP id 648227775
   for <win-eudora6-bugs@eudora.com>; Mon,  1 Nov 2004 15:38:06 -0800 (PST)
Message-Id: <5.0.2.1.2.20041101153017.00a26ec0@rain.org>
X-Sender: maia3@rain.org
X-Mailer: QUALCOMM Windows Eudora Version 5.0.2
Date: Mon, 01 Nov 2004 15:31:46 -0800
To: win-eudora6-bugs@qualcomm.com
From: Maia <maia3@rain.org>
Subject: New Bugs
Mime-Version: 1.0
Content-Type: text/plain; charset="us-ascii"; format=flowed
X-PMX-Version: 4.6.0.97784, Antispam-Core: 4.6.0.97340, Antispam-Data:
2004.11.1.3

Just now, without any reason, an email of mine was turned into a mixture of
text and directions and numbers and letters and is now unusable. Do you
know why this is happening?
      Thanks, Maia
```

- A set of headers, followed by a body
- The content and format of the body is described in the headers
- Some headers are defined by the IETF, others are protocol-specific, others are application-specific

How do people use MIME?

- Create a new MIME structure
- Parse a stream into a MIME structure
- Convert a MIME structure into a stream
  - You should be able to round-trip

MIME Parts

- Simple parts
  - Just header + body
- Multi-parts
  - The body is a sequence of MIME parts
- Message-parts

The MIME Library

- Uses Boost.Spirit for (most) parsing
- Header-only
- Configurable storage
- Parses from a stream or a pair of iterators
- Serializes out to a stream
- Getters and setters for headers, body, and sub-parts

Interface (partial)

```
template <class traits = detail::default_types>
class basic_mime {

   basic_mime ( const char *type, const char *subtype );
   basic_mime ( const headerList &theHeaders, const string_type
&default_content_type );

// What kind of part is this (simple, multi, message)
   part_kind get_part_kind () const;

   partIter     subpart_begin ();
   partIter     subpart_end   ();

   void append_part ( boost::shared_ptr<basic_mime> newPart );
   void remove_part ( partIter thePart );

// points to std::pair<std::string, string_type>
  headerIter    header_begin ();
  headerIter    header_end   ();

   void set_header_value  ( const char *key, const string_type &value, bool
replace = false );

   void remove_header ( headerIter theHeader );
```

What does the library do today?

- Parses MIME structures
- Outputs MIME structures as a stream.
- Can interrogate/edit structures

```
typedef boost::mime::basic_mime<>  mime_part;

//  (1) a really simple part
  mime_part mp ( "text", "plain" );
  mp.set_body ( "Hello World\n", 12 );
  std::cout << mp;
```

```
Content-Type:text/plain
Mime-Version:1.0 (Proposed.Boost.Mime 0.1)

Hello World
```

```
typedef boost::mime::basic_mime<>   mime_part;

   std::string str ( "<HTML><HEAD></HEAD><BODY>Hi Mom!</BODY></HTML>\n" );
   boost::shared_ptr<mime_part> mp0 (
      new mime_part ( mime_part::make_simple_part ( "text", "html",
str.begin (), str.end ())));

   boost::shared_ptr<mime_part> mp1 ( new mime_part ( "text", "plain" ));
   mp1->set_body ( "This is a test.....\n", 20 );

// Build a multipart
   mime_part mp2 ( "multipart", "multiple" );
   mp2.set_body ( "This is the body of a multipart\n", 32 );
   mp2.append_part ( mp0 );
   mp2.append_part ( mp1 );
   std::cout << mp2;
```

```
Content-Type:multipart/multiple; boundary="------=_NextPart-
Proposed.Boost.Mime.00008844"
Mime-Version:1.0 (Proposed.Boost.Mime 0.1)

This is the body of a multipart

--------=_NextPart-Proposed.Boost.Mime.00008844
Content-Type:text/html
Mime-Version:1.0 (Proposed.Boost.Mime 0.1)

<HTML><HEAD></HEAD><BODY>Hi Mom!</BODY></HTML>

--------=_NextPart-Proposed.Boost.Mime.00008844
Content-Type:text/plain; charset="usascii"
Mime-Version:1.0 (Proposed.Boost.Mime 0.1)

This is a test.....

--------=_NextPart-Proposed.Boost.Mime.00008844--
```

```
std::ostringstream os1, os2;
os1 << mp2;
std::istringstream is ( os1.str()); is >> std::noskipws;
boost::shared_ptr<mime_part> strmp = mime_part::parse_mime ( is );
os2 << strmp;
if ( os1.str () == os2.str ())
   std::cout << "Strings match!!" << std::endl;
```

What's still to do?

- Integration with Dean's cpp-netlib
- Bugs
- A couple more features to add
- Documentation
- Examples
- More tests

Future Work

- Additional parsers to make the library more useful
  - RFC2822 email addresses
  - RFC2047 encoded words
  - Requests that come in
- Some kind of “proxy” mechanisms for large MIME bodies; don’t require entire structure to be memory-resident
- A restartable parsing mechanism.

Questions?

The source for the library is available at:
http://github.com/mikhailberis/cpp-netlib/tree/0.6-devel
(this is both the cpp-netlib and MIME libraries)

marshall@idio.com