SMTP Transferred-By-Reference (TBR) Extension
draft-otis-smtp-tbr-ext-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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 March 4, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document describes an extension to SMTP that allows email to be exchanged as a storage system immutable XAM (eXtensible Access Method) reference. When MTAs employ the TBR mode, message origination can not be spoofed, and message acceptance is not asserted until retrieval of the referenced message. This strategy ensures a minimal SMTP overhead, increasing the responsibility of senders in order to limit the load of unwanted messages upon
receivers.

In addition, the TBR extension requires an [RFC2821] MAIL FROM address in the same domain as the server from which the XAM will be fetched, so that a dependable status-reporting path is assured.

Table of Contents

1. Introduction .................................................. 3
2. Conventions Used in this Document ............................ 4
3. TBR SMTP Extension ............................................ 4
   3.1. SMTP TBR Extension Registration ............................ 4
   3.2. TBR Transaction ........................................... 5
   3.3. TBR Requirements .......................................... 6
       3.3.1. Examples ............................................... 7
   3.4. Formal Syntax ............................................. 8
4. 8-Bit and Binary ............................................... 9
5. Handoff of responsibility to deliver or report errors ...... 10
6. Additional Enhanced Status Codes (RFC3463) .................. 10
7. Response Codes ................................................ 11
8. Reputation Checking ............................................ 12
9. Handoff of responsibility to deliver or report errors ...... 12
10. IANA Considerations .......................................... 13
11. Security Considerations ...................................... 13
12. Acknowledgements ............................................. 15
13. References ................................................... 15
    13.1. Normative References .................................... 15
    13.2. Informative References .................................. 16
Appendix A. Change Log ........................................... 17
Appendix B. XUID overview ........................................ 17
Appendix C. Meeting regulatory requirements ..................... 18
Authors’ Addresses ................................................ 18
Intellectual Property and Copyright Statements .................. 19
1. Introduction

This document defines an extension to Simple Mail Transfer Protocol (SMTP) that permits messages to be Transferred-By-Reference (TBR) using HTTP and eXtensible Access Method (XAM) infrastructure.

The TBR command changes the responsibility for storing an email message in transit such that a Message Transmission Agent (normally the Message Submission Agent) retains responsibility for storing the message instead of handing off responsibility at each SMTP step.

A Message Submission Agent (MSA) or subsequent MTA, upon initial receipt, stores messages using XAM infrastructure and provides XAM access via HTTP or HTTPS. Conversion from TBR mode is expected to be performed by Message Delivery Agents. This extension MAY be used in conjunction with a command PIPELINING mechanism [RFC2920].

Once published, the message can be fetched from a publicly accessible HTTP or HTTPS server using the eXAM-URI reference. Within the eXAM-URI reference, two parameters indirectly identify the originator and intended recipient. HTTP server logs permit identifying which messages have been retrieved by their recipient. By using the intended recipient reference parameters, publishing domains are able to identify messages which have not been delivered without reliance on SMTP feedback.

Upon receiving a TBR eXAM-URI email reference, the receiving SMTP server MAY perform any domain-reputation and/or IP-address-reputation checks called for by their policies. The TBR mode allows reputation checks to be done on the actual originator of an email (rather than merely the last-hop) before formal handoff, thus greatly simplifying and reducing the computational/network load on the receiver.

The TBR mode eliminates any need to send "unknown recipient" notifications to dubious sources, thwarting the "harvesting" of known-good email addresses.

Failure to receive a request to fetch the eXAM-URI should be logged by the TBR originator after a suitable delay, and a notice of failure SHOULD be forwarded to the original sender. Likewise, failure to complete fetching should be logged by the Message Delivery Agent, and a notice of failure MAY be sent to the recipient if such an option is requested. The number and timing of attempts to fetch the TBR eXAM-URI is a local option, but SHOULD follow an exponential backoff algorithm if more than one attempt is made.

Since a high percentage of current email is unwanted, it is expected that only a minority of TBR eXAM-URIs will actually be fetched. This
is good in that it reduces network traffic. Intentional failures to
fetch behaves very much like graylisting, in that it may benefit the
sender to keep the message queued, but the sender gets no indication
of why the fetching is delayed.

TBR provides an alternative strategy that reduces the overhead in
handling high levels of undesired messages. The TBR option offers a
safe and immediate means to exchange messages. TBR messages can be
processed for patterns of abuse prior to formally accepting the
obligation to deliver by fetching the message. A message not being
retrieved might be due to the source itself being considered abusive,
or the recipient being invalid. When the content of a message is
found to be undesired after fetching, the address for a DSN will have
been assured. TBR protects the identify of valid recipients and
eliminates any need for inbound email services to maintain a list of
valid recipients. This strategy enables TBR to restore the integrity
of email delivery.

2. 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 [RFC2119].

The formal syntax uses the Augmented Backus-Naur Form (ABNF)
[RFC4234] notation including the core rules defined in Appendix B of
RFC 4234.

3. TBR SMTP Extension

This section defines the TBR SMTP Extension.

3.1. SMTP TBR Extension Registration

1. The name of this submission extension is "TBR". This extends
   regular SMTP [RFC2821] server on port 25, and SHOULD NOT be
   advertised for Message Submission protocol on port 587 per
   [RFC4409].

2. The EHLO keyword value associated with the extension is "TBR".

3. An MTA which supports TBR will respond to EHLO with a TBR
   keyword. Clients MUST ignore arguments after the TBR EHLO
   keyword, unless defined by a subsequent IETF specification.

4. This extension adds the TBR SMTP verb. This verb is used as a
   replacement for the DATA command and is only permitted during a
   mail transaction after at least one RCPT TO which was not
   rejected.
3.2. TBR Transaction

The TBR command supports transfer of http or https eXAM-URIs, instead of message content using the DATA command.

A simple TBR transaction will consist of MAIL FROM, one or more RCPT TO commands, and a TBR command terminating with the End-Mark tag. The TBR command takes the place of the DATA command, and includes three, or four arguments:

- Forwarding Count (mandatory)
  Starting at zero, this count is expressed as a decimal integer representing the number of times the TBR eXAM-URI has been transferred by SMTP.

- eXAM URI (mandatory)
  This is a URI pointing to a fully formed message ready for injection into SMTP infrastructure.

- Additional Trace Headers (optional)
  Each MTA after the one which first encodes the eXAM-URI MAY include trace headers to be prepended when the eXAM URI is retrieved; however there is no assurance that they will be passed on through the chain of MTAs. Each MTA receiving these optional headers MAY log them instead of passing them on. If present, trace header(s) will be preceded and followed by CRLF, and a line containing only the End-Mark tag will follow the last trace header. Trace headers MUST follow the format defined in [RFC2822].

- End-Mark tag (mandatory)
  The End-Mark tag is a line containing only the character "." (period or full stop) followed by CRLF.

If PIPELINING [RFC2920] is advertised, the client MAY send the entire transaction in one round trip. If no valid RCPT TO address is supplied, the TBR command will simply fail. If at least one valid RCPT TO address is supplied, then the TBR eXAM-URI argument will be accepted.

Trace Headers can call for large amounts of storage which serves no useful purpose in the case where eXAM-URI retrieval will not be done. Thus, storage of Trace Headers included within the TBR command is entirely optional (even on a message-by-message basis).

When Trace Headers are not saved for passing on to succeeding MTAs, storage requirements per TBR command SHOULD be limited to the 512 bytes as specified by [RFC2821] for SMTP commands. The TBR command includes the TBR verb and line containing the eXAM-URI, in addition to the terminating line containing the End-Mark tag.

When validation of message acceptance by each recipient is desired,
this necessitates separate messages containing one recipient and an
eXAM-URI containing a corresponding ‘rcpt-ref’ field.

If PIPELINING [RFC2920] is also advertised, then the client may
pipeline the entire transaction in one round-trip. However, the
client MUST wait for the results of the End-Mark tag in the TBR
command prior to initiating a new transaction. The TBR command does
not direct the server to fetch the message to which the eXAM-URI
refers, nor to replace the eXAM-URI.

The Forwarding count (fwd-cnt) is initially set to zero. Every time
the eXAM-URI is forwarded, the count is incremented. When the
forwarding count exceeds 100, as recommended by [RFC2821] in Section
6.2 Loop Detection, a routing loop has been detected, and should be
handled accordingly.

Prior to being fetched, TBR messages are not required to generate
Delivery Status Notifications when being dropped. Instead, fetching
the referenced message offers the indication of message acceptance.

When an MTA has accepted a TBR message and must forward it to an MTA
which does not support TBR, it MAY (after all appropriate checking)
retrieve the replacement message content from the eXAM-URI, prepend
any additional trace headers, and forward it as a non-TBR message.
As an alternative, instead of retrieving the replacement message
content, it MAY prepend any additional trace headers to a
notification sent to the recipient containing the eXAM-URI itself,
along with any other appropriate information.

3.3. TBR Requirements

The domain part of the [RFC2821] MAIL FROM address MUST exactly match
the domain part of the URI, in order to assure that recipients do not
generate "backscatter" to forged return addresses at domains which do
not support TBR. A domain which supports TBR SHOULD employ methods
of coding the localpart so as to recognize (and discard) backscatter
DSNs. The MX server for the TBR domain SHOULD coordinate with the
encoding MTA to properly decode and check the localpart of the return
address.

The assurance that a domain supports TBR is proven by the DNS records
for a domain starting with _tbr.* returning both MX and address
records (adequate for fetching the URI). Ideally, address record(s)
will be present for each address family (such as IPv4 and IPv6)
currently in use; if it turns out there’s a mismatch, the MTA
attempting retrieval should generate a DSN (after performing these
validity checks).
Fetching the message SHOULD occur after the Mail Delivery Agent has finished all processing prior to placing the email into a mailbox or forwarding it to a processing program.

Removal of published messages should be delayed for a short period to accommodate a retry resulting from possible message storage related failures.

3.3.1. Examples

In examples, "C:" and "S:" indicate lines sent by the client and server, respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange. To simplify the example, an abbreviated XUID is used. Two successful submissions (without and with pipelining) follow:

(non-pipelined transaction)
C: EHLO client.example.com
S: 250-server.example.com
S: 250-TBR
S: 250-ENHANCEDSTATUSCODES
S: 250-PIPELINING
S: 250-8BITMIME
S: 250-SIZE 30000000
S: 250-DSN
S: 250-ETRN
S: 250-AUTH PLAIN LOGIN
S: 250-STARTTLS
S: 250-DELIVERBY
S: 250 HELP
C: MAIL FROM:<prvs=02460E6DB6=tom@_tbr.example.com>
S: 250 2.5.0 Address Ok.
C: RCPT TO:<dick@users.example.com>
S: 250 2.1.5 dick@users.example.com OK.
C: TBR 0 https://_tbr.example.com/~Q012?XUID=A42L0M726P&RCPT=R012
C: .
S: 250 2.5.0 Ok.

(pipelined transaction)
C: EHLO client.example.com
S: 250-server.example.com
S: 250-TBR
S: 250-ENHANCEDSTATUSCODES
S: 250-PIPELINING
S: 250-8BITMIME
S: 250-SIZE 30000000
S: 250-DSN
S: 250-ETRN
S: 250-AUTH PLAIN LOGIN
S: 250-STARTTLS
S: 250-DELIVERBY
S: 250 HELP
C: MAIL FROM:<prvs=02460E6DB6=tom@_tbr.example.com>
C: RCPT TO:<dick@users.example.com>
C: TBR 0 https://_tbr.example.com/~Q012?XUID=A42L0M726P&RCPT=R012
C: .
S: 235 2.7.0 PLAIN authentication successful.
S: 250 2.5.0 Address Ok.
S: 250 2.1.5 dick@users.example.com OK.
S: 250 2.5.0 Ok.

Some examples of failure cases:

C: MAIL FROM:<prvs=02460E6DB6=tom@_tbr.example.com>
C: RCPT TO:<harry@users.example.com>
C: TBR 0 https://_tbr.example.com/~Q012?XUID=A42L0M726P&RCPT=R012
C: .
S: 250 2.5.0 Address Ok.
S: 550 5.7.1 Relaying not allowed: harry@users.example.com
S: 554 5.5.0 No recipients have been specified.

3.4. Formal Syntax

This specification inherits ABNF [RFC4234], Uniform Resource Identifiers [RFC3986] and Internet Message Format [RFC2822].

\[
\begin{align*}
\text{dash} & = \%d45 ; "-" \\
\text{dot} & = \%d46 ; "." \\
\text{slash} & = \%d47 ; "/" \\
\text{underscore} & = \%d95 ; "_" \\
\text{tilde} & = \%d126 ; "~" \\
\text{tbr-ref} & = *(\text{ALPHA} / \text{DIGIT} / \text{dash} / \text{dot} / \text{underscore} / \text{tilde} / \text{slash} / \text{pct-encoded}) \\
\text{pct-encoded} & = "\%" \text{HEXDIG HEXDIG} \\
\text{dns-char} & = \text{ALPHA} / \text{DIGIT} / \text{dash} \\
\text{id-prefix} & = \text{ALPHA} / \text{DIGIT}
\end{align*}
\]
label = id-prefix [*61dns-char id-prefix]
sldn = label dot label

hostname = *(label dot) sldn
; not to exceed 249 characters
; excluding "_tbr."

tbr-prot = "http" / "https"
tbr-host = "_tbr." hostname

port = 1*5DIGIT

base-char = (dns-char / "_")

rcpt-ref = *43(base-char) *2("=")

orig-ref = *43(base-char) *2("=")

con-id = 1*107(base-char) *2("=") ; url safe base64

tbr-cmd = "TBR" SP fwd-cnt SP eXAM-URI FWS tbr-end

tbr-end = [trace] end-mark CRLF

fwd-cnt = 1*3DIGIT

dot

end-mark = dot

tbr-pub = tbr-prot"://"tbr-host[": port]

tbr-ref = orig-ref"?XUID="con-id"&RCPT="rcpt-ref

eXAM-URI = tbr-pub "/" tbr-ref

eXAM-ref = "TBR" SP fwd-cnt SP eXAM-URI

4. 8-Bit and Binary

A submit server that advertises TBR MUST also advertise 8BITMIME [RFC1652] and MUST perform the down conversion described in that specification on the resulting complete message if 8-bit data is obtained after replacing the eXAM-URI with the referenced message and passed to a 7-bit server. If the eXAM-URI argument to TBR refers to binary data, then the submit server MUST down convert as described in Binary SMTP [RFC3030]. The correct character repertoire MUST be asserted when offering 8-bit data. See [RFC4646] and [RFC4647].
5. Handoff of responsibility to deliver or report errors

When a TBR command is given, the formal handoff of responsibility does not occur during the SMTP session, but is delayed until the eXAM-URI is fetched. The formal handoff of responsibility to deliver or report problems comes when the command to fetch the URI is sent. There MAY be no failure report when TBR messages are undesired and thus never fetched. When an attempt to fetch a message is made, the fetching agent thereby accepts responsibility for either delivering the message or properly reporting a failure to do so.

The SMTP server receiving a TBR reference MAY reject invalid recipients during the SMTP session, or it may quietly ignore recipients which turn out to be invalid. It MUST NOT generate Delivery Status Notification messages before it verifies that the MAIL FROM domain exactly matches the domain part of the URI, and that the same domain also publishes an MX record. Having verified that, it MAY report any conditions via DSNs, but is not required to report any errors until it finishes reputation checks and fetches the URI.

Postponing the formal handoff of responsibility requires the originating MSA to obtain notification when an eXAM-URI has not been fetched after some period. This period is determined by local option, but should be set to suppress most failure notifications which might occur while delivery of the eXAM-URI is still pending.

6. Additional Enhanced Status Codes (RFC3463)

SMTP or Submit servers that advertise ENHANCEDSTATUSCODES [RFC2034] use enhanced status codes defined in [RFC3463]. The TBR extension introduces new error cases that RFC 3463 did not consider. The following additional enhanced status codes are defined by this specification:

451 4.7.8 TBR HTTP/HTTPS mode requested for immediate acceptance
451 4.7.9 TBR HTTP mode requested for immediate acceptance
451 4.7.10 TBR HTTPS mode requested for immediate acceptance

Where a receiving SMTP server implements greylisting due to low trust in the sending SMTP server or the originating domain, this temporary error may be given, committing the receiver to accepting a TBR email without further temporary errors for greylisting.

550 5.1.9 MAIL FROM not within eXAM-URI domain
The domain publishing the message MUST receive all Delivery Status Notifications and MAY redirect them to the original RFC2821 MAIL FROM address, or otherwise process them in accordance with directives agreed to by the originator and MSA.

504 5.5.6 eXAM-URI protocol not supported

The domain fetching the message does not support the protocol indicated within the eXAM-URI. This status code can also extend a response code of 504 (504 Command parameter not implemented).

504 5.5.7 eXAM-URI IP address not supported

The domain fetching the message does not support the IP address returned for the eXAM-URI host.

7. Response Codes

This section includes example response codes to the TBR command. Other text may be used with the same response codes. This list is not exhaustive, and TBR clients MUST tolerate any valid SMTP response code. Most of these examples include the appropriate enhanced status code [RFC3463].

554 5.5.0 No recipients have been specified

This response code occurs when TBR is used (for example, with PIPELINING) and all RCPT TOs failed.

503 5.5.0 Valid RCPT TO required before TBR

This response code is an alternative to the previous one when TBR is used (for example, with PIPELINING) and all RCPT TOs failed.

503 5.5.0 TBR command not terminated

A TBR command without the "." End-Mark tag was sent. The eXAM-URI may have been received, but will not be forwarded.

554 5.3.4 Message too big for system

The message (subsequent to eXAM-URI message replacement) is larger than the per-message size limit for this server.

552 5.2.2 Mailbox full

The recipient is local, the submit server supports direct delivery,
and the recipient has exceeded his quota and any grace period for delivery attempts.

554 5.6.7 eXAM-URI resolution failed

Obtaining the message from the HTTP server returned an error or no data.

250 2.5.0 Ok.

The eXAM-URI was accepted.

8. Reputation Checking

The TBR mode of operation allows greater emphasis to be placed upon the reputation of the originator. Messages containing malware or undesired content can be substantially reduced without risking ever-growing burdens upon the receiving server. The TBR mode of operation is able to ensure that application of message security remains scalable, and does so by imposing a greater burden upon the sender.

Upon receiving a TBR email reference, the receiving SMTP server MAY perform any domain-reputation and/or IP-address-reputation checks called for by their policies. The TBR mode allows reputation checks to be done on the actual originator of an email (rather than merely the last-hop) before formal handoff, thus greatly simplifying the computational/network load on the receiver.

Since a high percentage of current email is unwanted, it is expected that only a minority of TBR URIs will actually be fetched. This is good in that it reduces network traffic. Intentional failure to fetch behaves very much like graylisting, in that it may benefit the sender to keep the message queued, but the sender gets no indication of why the fetching is delayed.

The TBR mode eliminates any need to send "unknown recipient" notifications to dubious sources, thwarting the "harvesting" of known-good email addresses.

9. Handoff of responsibility to deliver or report errors

The SMTP server receiving a TBR reference does not immediately accept responsibility for delivery or reporting problems. It MAY reject invalid recipients during the SMTP session, or it may quietly ignore recipients which turn out to be invalid. It MUST NOT generate Delivery Status Notification messages before it verifies that the
MAIL FROM domain exactly matches the domain part of the URI, and that the same domain also publishes an MX record. Having verified that, it MAY report any conditions via DSNs, but is not required to report any errors until it finishes reputation checks and fetches the URI.

There may be no failure report if TBR messages are undesired and thus not fetched. If the message is fetched, the fetching agent accepts responsibility for either delivering the message or properly reporting a failure to do so. The formal handoff of responsibility to deliver or report problems comes when the command to fetch the URI is sent.

Defining formal handoff this way replaces a single race-condition (whether the 250 response to DATA is received) with a more complicated set of race-conditions. We define the sending of the command to fetch the URI as formal handoff, without regard to whether the command is received, or even whether a server exists to receive it. While the RFC2821 race-condition could theoretically generate duplicate messages, the TBR race-conditions cannot. It is critical, of course, that the validity tests for the DSN path be performed before the command to fetch is issued, so that a DSN may be issued if fetching fails.

Postponing the formal handoff of responsibility requires the originating MSA to obtain notification when an eXAM-URI has not been fetched after some period. This period is determined by local option, but should be set to suppress most failure notifications which might occur while delivery of the eXAM-URI is still pending.

10. IANA Considerations

The "TBR" SMTP extension as described in Section 3 needs to be registered. This registration should be marked in the registry as not intended for message submission [RFC4409].

11. Security Considerations

Messages published on HTTP servers could in principle be subject to URI-guessing attacks. Protective schemes SHOULD be employed to discourage testing large numbers of generated URIs in hopes of obtaining a private email; this issue has been addressed in systems that depend upon confidential responses prior to granting access. Often protective schemes log an IP address and related CIDR with invalid reference attempts where delays are introduced to rate-limit guessing.
In addition to the XUID obscuring valid message locations, the eXAM-URI should cryptographically encode a valid recipient and a path component that represents the originator. This added protection further ensures message access is not obtained through guessing.

The TBR mode of operation requires greater emphasis be placed upon the reputation of the originator. Detection of messages containing malware or undesired content can be defeated with polymorphic techniques which place ever growing burdens upon the receiving server. Only the TBR mode of operation is able to ensure that application of message security remains scalable, and does so by imposing a greater burden upon the sender.
12. Acknowledgements

This document follows the general structure of Message Submission BURL Extension [RFC4468].

13. References

13.1. Normative References


Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006.


13.2. Informative References


Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.


Gellens, R., "The Text/Plain Format and DelSp Parameters", RFC 3676.


Appendix A. Change Log

Appendix B. XUID overview

An XSet (data and metadata storage entity) is identified by a XUID. The XUID (XSet Unique Identifier, pronounced zoo’id) is created by the XAM storage system. The XUID conforms to a defined format. See http://www.snia.org/xam. The native format of a XUID is a variable-length byte string, with a maximum length of 80 bytes. XAM applications are expected to treat XUIDs as opaque byte strings. However, the XUID format is defined such that the XUID integrity can be validated, and that vendors are able to assign XUID values independently.

An OID field is contained within the XUID. The OID field represents the SNMP enterprise number of the XAM Storage System vendor that created the XUID, in network byte order. See [RFC2578] and http://www.iana.org/assignments/enterprisenumbers. An OID of 0 is reserved.

The native format for a XUID is binary. The XUID textual representation will be "URL and Filename safe" base64-encoded, as described in [RFC4648], which uses US-ASCII. The XUID is able to contain a hash as large as 576 bits allowing as many as 2.47 x 10^173 different values.
Appendix C. Meeting regulatory requirements

Government agencies are introducing new requirements for retention of email, such as United State’s SEC rule 17a-4 which defines preservation requirements. Archiving and preservation of email is satisfied by XAM Storage. Rule 17a-4 contains at least three important requirements accommodated by XAM storage:

1. Immutable storage,
2. Verifiable accuracy, and
3. Deterministic retention.

Authors’ Addresses

Douglas Otis  
Trend Micro, NSSG  
10101 N. De Anza Blvd  
Cupertino, CA  95014  
USA  
Phone: +1.408.257-1500  
Email: doug_otis@trendmicro.com

John Leslie  
JLC.net  
10 Souhegan Street  
Milford, NH  03055  
USA  
Phone: +1.603.673.6132  
Email: john@jlc.net