ROSETTANET
POWERED BY GS1 US

Technical Advisory

RosettaNet Implementation Framework V02.00.01

S/MIME Compression of RosettaNet Payload

Issue 01.01.00
17 January 2006

Table of Contents

1 Document Management

1.1 Legal Disclaimer

RosettaNet, its members, officers, directors, employees, or agents shall not be liable for any injury, loss, damages, financial or otherwise, arising from, related to, or caused by the use of this document or the specifications herein, as well as associated guidelines and schemas. The use of said specifications shall constitute your express consent to the foregoing exculpation.

1.2 Copyright

1.3 Trademarks

RosettaNet, Partner Interface Process, PIP and the RosettaNet logo are trademarks or registered trademarks of "RosettaNet," a non-profit organization. All other product names and company logos mentioned herein are the trademarks of their respective owners. In the best effort, all terms mentioned in this document that are known to be trademarks or registered trademarks have been appropriately recognized in the first occurrence of the term.

1.4 Acknowledgments

This document has been prepared by RosettaNet (http://www.rosettanet.org/). Listed below are companies that contributed towards this document:

IBM	Sterling Commerce
ST Microelectronics	

1.5 Related Documents

- RosettaNet Implementation Framework: Core Specification 2.0 [RNIF20]

1.6 Purpose

RosettaNet Implementation Framework (RNIF) 2.0 [RNIF20] describes the use of S/MIME within RosettaNet for the purposes of signing and encrypting RosettaNet Business Messages. This Technical Advisory (TA) describes the use of S/MIME for compressing the Service Content and Attachment, in conjunction with or without signing and encryption.

1.7 Scope

This document contains information describing enhancements to the RNIF 2.0 Specification regarding the use of S/MIME. This document does not contain any other RNIF changes or information regarding PIPs.

1.8 Conformance Statement

Compliance to the enhancements described in this advisory is mandatory if S/MIME compression of payloads is implemented in an RNIF 2.0 implementation. Applications that conform to this TA MUST still conform to all requirements of [RNIF20].

1.9 Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

MIME Headers are described using `courier` format.

Examples have been doctored to improve clarity and reduce space.

1.10 Document Version History

Version	Date	Description
Issue 01.00.00	11 July 2003	Issue
Draft Issue 01.01.00	21 September 2005	Membership Feedback
Issue 01.01.00	17 January 2006	Issue

2 Introduction

This Technical Advisory (TA) prescribes changes to the RosettaNet Implementation Framework (RNIF) 2.0 [RNIF20] when compression and decompression of RosettaNet Business Message is required.

2.1 Terms

The terms RosettaNet Business Message, Service Content, Attachments and Payload are defined in RNIF 2.0, Section 2.3.1.

2.2 Issue

This Technical Advisory is a response to the need to identify a solution to improve the performance issues related to transporting RosettaNet payload over RNIF 2.0. The current performance issues, reported in connection with PIP 4A4 and PIP 7B1, originate from the size (megabytes) of the messages. Applying compression techniques that are described in this document can reduce the size of payload prior to storage, encryption and signing, and transmission. Note that this TA will not solve problems related to memory requirements for constructing payloads prior to compression, or for, parsing, validating, and processing the payload after decompression.

3 S/MIME Compression of RosettaNet Business Message

In RNIF 2.0 [RNIF20], Section 2.2.1 describes the use of S/MIME for the purposes of signing and encrypting RosettaNet Business Messages. This Technical Advisory describes the use of S/MIME for compressing and decompressing the Service Content and Attachment, in conjunction with or without signing and encryption. These enhancements are described as follows:

1. Compression algorithms and format
2. Change in message packaging (enhancements to RNIF 2.0, Section 2.3.3)
3. Change in message unpacking (enhancements to RNIF 2.0, Section 2.3.4)
4. RNIF Error messages

3.1 Compression Algorithms and Format

S/MIME compression in RNIF 2.0 MUST follow the guidelines of RFC 3274 [RFC3274] that describes a Compressed Data Content Type for Cryptographic Message Syntax (CMS). RFC 3274 REQUIRES support for ZLIB compression algorithm [RFC1950].

A compressed Service Content or Attachment will be wrapped within a S/MIME Envelope. The `Content Type` MIME Header for a compressed S/MIME Envelope would have a `smime-type` of `compressed-data.`

To improve compression efficiency and keep the final message size smaller, Content-Transfer-Encoding MUST not be applied to the Service Content and/or Attachments prior to compression. The choice of Content-Transfer Encoding for the compressed data is left to the RNIF implementation and MUST comply with requirements in RNIF 2.0 (sections 2.2 and 2.3).

The S/MIME Envelope of the compressed data MUST have a Content-ID. The value of the Content-ID SHOULD be the Content-ID of the decompressed data with a suffix of "`--z`". Note that the compressed data would contain the MIME headers of the decompressed data, and crosschecking is possible with the Content-ID of the compressed data.

Example 1. Compression Format

Consider a MIME body part containing Service Content:

```
Content-Type: application/xml;
Content-ID: <content-id-of-service-content>
…
```

The corresponding S/MIME Envelope containing compressed data would look like this:


```
Content-Type: application/pkcs7-mime; smime-type=compressed-data;
        name=filename.p7z
Content-ID: <content-id-of-service-content--z>
```

Note that RFC 2311 [RFC2311] limits the filename to 8 characters and file name extension to 3 characters (“8.3 format”).

3.2 Packaging Compressed Messages

S/MIME compression MUST be applied to Service Content or Attachments prior to encryption or signing. Service Content and each Attachment must be compressed individually. It is permissible for only some (one, more than one, or all) of Service Content or Attachments to be compressed. For example, there can be several Attachments with only one of them compressed. Below are multiple packaging scenarios with examples.

3.2.1 No Encryption, No Signature

Figure 1 contains an example where ONLY the Service Content is compressed.

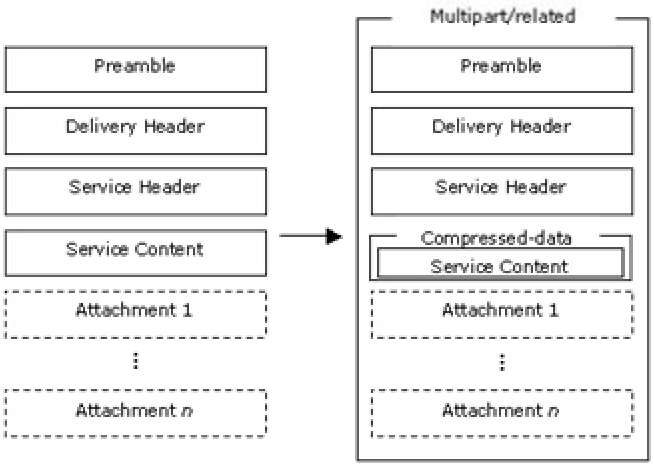

Figure 1: No Encryption, No Signature

Example 2. No Encryption, No Signature

```
Content-Type: multipart/related;
              type="multipart/related";
              boundary="RN-Outer-Boundary"


--RN-Outer-Boundary
Content-Type: multipart/related;
              type=application/xml;
              boundary="RN-Payload-Boundary"
```



```
--RN-Payload-Boundary
Content-Type: application/xml
Content-ID: <Content-ID-of-Preamble>
Content-location: "RN-Preamble"

[Preamble Goes Here]

--RN-Payload-Boundary
Content-Type: application/xml
Content-ID: <Content-ID-of-Delivery-Header>
Content-location: "RN-Delivery-Header"

[DeliveryHeader Goes Here]

--RN-Payload-Boundary
Content-Type: application/xml
Content-ID: <Content-ID-of-Service-Header>
Content-location: "RN-Service-Header"

[Service Header Goes Here]

--RN-Payload-Boundary
Content-Type: Application/pkcs7-mime;
             name=filename.p7z;smime-type=compressed-data
Content-ID: <Content-ID-of-Service-Content—-z>
Content-Transfer-Encoding: base64

MIAGCyqGSIb3DQEJEAEJoIAwgAIBADANBgsqhkiG9w0BCRADCDCABgkqhkiG9w0BbwGggA
A/F4nO1XS3PaSBC+u8r/YZZTchCSwBiWUsg6PFLsxjYb4WS9F9cgNdFUSTPKaGSHf78zQm
+/Xb88gwXtr2bIPvv8j1Weq036D/X5+/Z33Ou1UCj/48/g9BSDmsAAAAAAAAAAAAA==

--RN-Payload-Boundary--

--RN-Outer-Boundary--
```

3.2.2 Encryption, No Signature

RNIF 2.0 specifies two options for encryption. Encrypt only Service Content and Attachments, or encrypt Service Header, Service Content, and Attachments. In either of the cases, the MIME part corresponding to the Service Content and any of the Attachments will be wrapped in an S/MIME Envelope when compressed; and compressed-data S/MIME Envelopes are encrypted. Without compression, corresponding (non-S/MIME) MIME body parts would have been encrypted.

If the Service Content and/or Attachments are encrypted, then Content-Transfer-Encoding, prior to as well as right after compression, MUST be completely avoided, as only the final encrypted MIME part is exposed to the transfer protocol.

3.2.3 No Encryption, Signature

RNIF 2.0 specifies the use of `multipart/signed` Content Type for the purpose of signing. While the Service Content or Attachments may be compressed and wrapped inside an S/MIME envelope as needed, the

corresponding signature (calculated over compressed data) contained in the S/MIME part with Content Type `application/pkcs7-signature` MUST not be compressed.

Example 3. No Encryption, Signature

```
Content-Type: multipart/related;
              type="multipart/signed";
              boundary="RN-Outer-Boundary"


--RN-Outer-Boundary
Content-Type: multipart/signed;
              protocol="application/pkcs7-signature";
              micalg=sha1;
              boundary="RN-Signature-Boundary"


--RN-Signature-Boundary
Content-Type: multipart/related;
              type=application/xml;
              boundary="RN-Payload-Boundary"


--RN-Payload-Boundary
Content-Type: application/xml
Content-ID: <Content-ID-of-Preamble>
Content-location: "RN-Preamble"

[Preamble Goes Here]

--RN-Payload-Boundary
Content-Type: application/xml
Content-ID: <Content-ID-of-Delivery-Header>
Content-location: "RN-Delivery-Header"

[Delivery Header Goes Here]

--RN-Payload-Boundary
Content-Type: application/xml
Content-ID: <Content-ID-of-Service-Header>
Content-location: "RN-Service-Header"

[Service Header Goes Here]

--RN-Payload-Boundary
Content-Type: Application/pkcs7-mime;
             name=something.p7z;
             smime-type=compressed-data
Content-ID: <Content-ID-of-Service-Content—-z>
Content-Transfer-Encoding: base64

MIAGCyqGSIb3DQEJEAEJoIAwgAIBADANBgsqhkiG9w0BCRADCDCABgkqhkiG9w0BBwGggA
A/F4nO1XS3PbNhC+e8b/AdUpOVB82JJcDaPU0SOjNrbVUG7rXjwUuQoxAwIMCNrRvy9A8W
7r/fnq8M47Vtzw4Q8irXZ6nTfoP+f33+N+tz0a0SePTn8T+MUDlyAAAAAAAAAAAAAA==

--RN-Payload-Boundary--

--RN-Signature-Boundary
Content-Type: Application/pkcs7-signature;
              name=somesig.p7s
Content-Transfer-Encoding: base64
```

```
MIIDFQYJKoZIhvcNAQcCoIIDBjCCAwICAQExCzAJBgUrDgMCGgUAMAsGCSqGSIb3DQEHAa
AfwwggH4MIIBYaADAgECAgEBMA0GCSqGSIb3DQEBBQUAMDgxFDASBgNVBAMTC1JOSUYyMC
ZeVdn511Fxt17HtdmJPwqPCWHGRgUvj7SblZha4GhEwbfNTCAbg==

--RN-Signature-Boundary--

--RN-Outer-Boundary--
```

3.2.4 Encryption and Signature

If both encryption and digital signature are applied to a Business Message, RNIF 2.0 requires encryption to be performed prior to signing. Therefore, all the rules stated in the earlier section, “Encryption with no Signature” apply to this case. The encrypted S/MIME Envelope is signed as described in RNIF 2.0 specification.

3.3 Unpacking Compressed Message

The Service Content and any Attachments are decompressed and extracted in the reverse order of compression. There is no assumption that decompression needs to be performed prior to persistent storage. Entire message or individually compressed MIME body parts may be stored. The choice of which approach to take is left to the RNIF implementations and associated applications. See the next section for error messages during decompression.

3.3.1 RNIF Error Messages during unpacking

The errors during unpacking of the compressed message are in the table below.

	ERROR	
	Service Content	**Attachment**
Decompressing	UNP.SCON.UNCERR	UNP.ATT.UNCERR
Decompression not supported by receiving RNIF implementation	UNP.SCON.NOUNCERR	UNP.ATT.NOUNCERR
Reading	UNP.SCON.UREADERR	UNP.ATT.UREADERR
Note: The reading errors may be used for debugging purposes.		

Figure 2 shows a portion of Figure 18 in the RNIF 2.0 specification modified to include decompression related flows, including error handling.

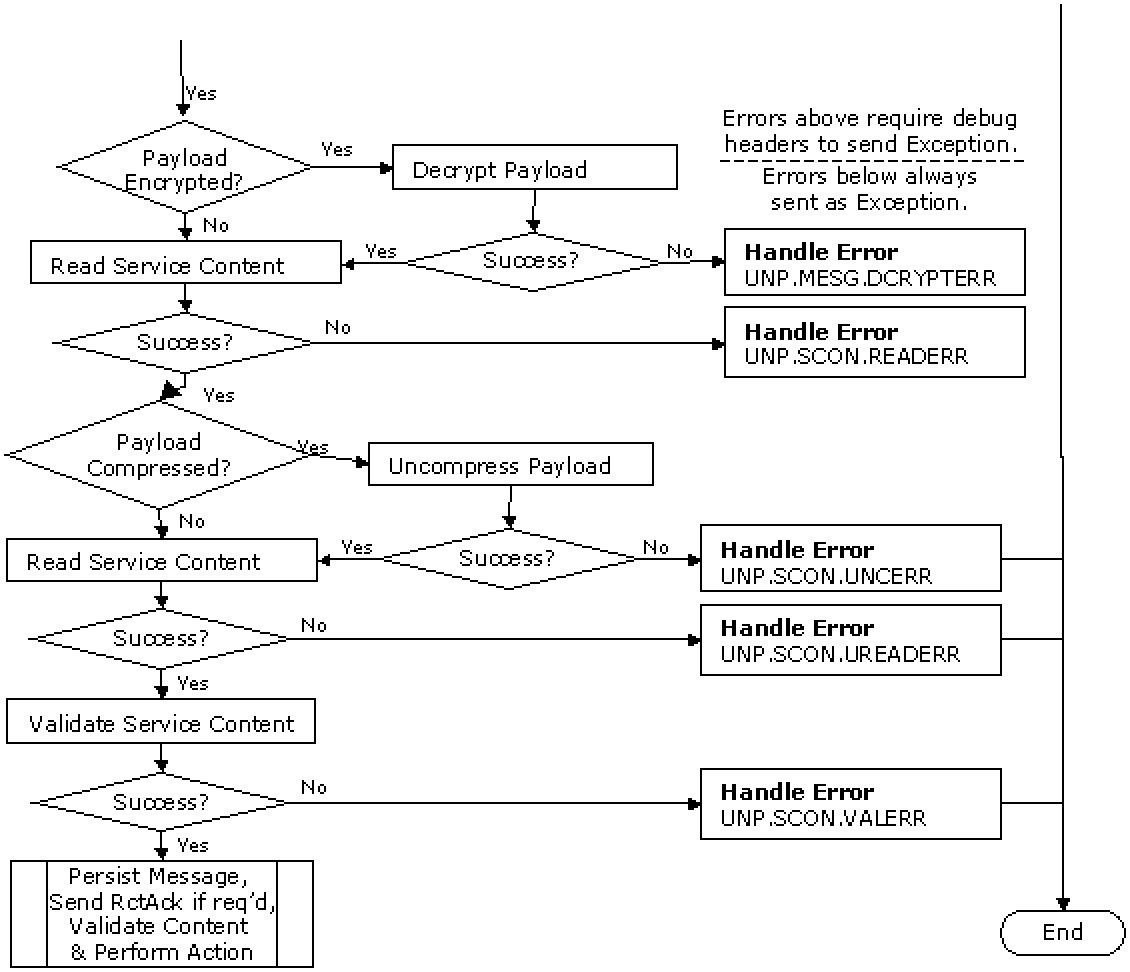

Figure 2: Modified Message Processing Flow

If a receiving application does not support compressed payloads, an UNP.SCON.NOUNCERR error code MAY be generated when the compressed content is encountered, and the error SHOULD be handled according to RNIF section 2.3.4.

4 Benefits

Using this compression technique has the following benefits:

1. The RosettaNet Business Message transmission time and/or bandwidth required for such transmission would be lower because of the reduced size of payload.

2. Compressing the payload prior to signing and encryption, as this advisory recommends, reduces the size of data transmitted. Reduced data implies more efficient and faster processing of cryptographic or other processing.

3. Compressing the payload would considerably reduce the storage at the trading partner sites. This savings become even more significant when the data is stored in multiple locations to prevent data loss.

Note that this TA will not solve problems related to memory requirements for constructing payloads prior to compression, or for, parsing, validating, and processing the payload after decompression.

5 Implementation Considerations

1. The ZLIB compression algorithm has no known intellectual property restrictions and has at least one freely available reference implementation.

2. The need for Content-Transfer-Encoding the compressed data is dependent on whether an 8bit safe transport mechanism is used. See section 2.3 of [RNIF20] for details on Content-Transfer-Encoding. In particular, Content-Transfer-Encoding of Service Content and/or Attachments MUST be avoided prior to compression.

3. Compressing the payload prior to encryption removes redundancy in the data, and may reduce security vulnerabilities related to plaintext data redundancy. However, implementations should be aware of security risks associated with combining security sensitive data with non-security sensitive data.

4. Compressing already compressed data does not usually reduce size, and therefore is not advised. Therefore, it is advised to compress the topmost MIME envelope that needs all its parts to be compressed, in order to avoid multiple compressions of the same data (from multiple MIME parts).

5. Compression/decompression may deliver any performance gains only when used for sufficiently large enough data due to the execution time required for compression and decompression. Weigh your options carefully.

6. When to use compression on Service Content and/or Attachments is a decision that the trading partners would make based on the specific implementation performance and capacity requirements. If the RNIF implementation has capacity restrictions, such as maximum size of send/receive transmission message, compression may be used to accommodate that. On the other hand, compressing every message may not be the answer, since compression and decompression do take some finite amount of processing time, and performance considerations may take precedence over message size considerations. In any case, only some or all messages will be compressed can be answered only by trading partners based on specific implementation requirements.

7. This TA does not address other means of reducing the size of messages such as breaking a message into multiple parts.

8. Compression/decompression of archived files is valid. Since Service Content and Attachment need to be compressed individually, as stated in Section 3.2, they also must be archived in the same fashion. The files being archived may also be compressed, even though for reasons stated in (5) above, it is not advisable.

   An example of how MIME Headers might change at various stages of compression is below. We only show the Service-Content in this example. Example 4 is the decompressed content, which is shown archived in zip format in Example 5, followed by compressed format in Example 6. It is not necessary that the intermediate step of “decompressed, archived” be preserved in MIME format – the individual implementations may use libraries to convert the MIME parts from “decompressed” to “compressed and archived” format in a single step. The inclusion of this intermediary stage here is for illustration only.

Example 4. Decompressed

```
--RN-Payload-Boundary
Content-Type: Application/xml;
Content-ID: <Content-ID-of-Service-Content>
Content-Transfer-Encoding: 7bit

[Service Content goes here]

--RN-Payload-Boundary
```

Example 5. Decompressed, Archived

```
--RN-Payload-Boundary
Content-Type: Application/zip;
Content-ID: <Content-ID-of-Service-Content--a>

[Archived Service Content goes here]

--RN-Payload-Boundary
```

Example 6. Compressed

```
--RN-Payload-Boundary
Content-Type: Application/pkcs7-mime;
              name=something.p7z;
              smime-type=compressed-data
Content-ID: <Content-ID-of-Service-Content—-z>
Content-Transfer-Encoding: base64

MIAGCyqGSIb3DQEJEAEJoIAwgAIBADANBgsqhkiG9w0BCRADCDCABgkqhkiG9w0BBwGggA
A/F4nO1XS3PbNhC+e8b/AdUpOVB82JJcDaPU0SOjNrbVUG7rXjwUuQoxAwIMCNrRvy9A8W
7r/fnq8M47Vtzw4Q8irXZ6nTfoP+f33+N+tz0a0SePTn8T+MUDlyAAAAAAAAAAAAAA==

--RN-Payload-Boundary--
```

6 References

[RFC1950] "ZLIB Compressed Data Format Specification version 3.3," May 1996, http://www.ietf.org/rfc/rfc1950.txt

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels," March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2311] "S/MIME Version 2 Message Specification" March 1998, http://www.ietf.org/rfc/rfc2311.txt

[RFC3274] "Compressed Data Content Type for Cryptographic Message Syntax (CMS)," June 2002, http://www.faqs.org/rfcs/rfc3274.html

[RNIF20] "RosettaNet Implementation Framework: Core Specification," Version V02.00.01, March 6, 2002.

