[MC-NBFSE]:
.NET Binary Format:
SOAP Extension

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
08/10/2007	0.1	Major	Initial Availability
09/28/2007	0.2	Minor	Updated the technical content.
10/23/2007	0.2.1	Editorial	Revised and edited the technical content.
11/30/2007	0.2.2	Editorial	Revised and edited the technical content.
01/25/2008	0.2.3	Editorial	Revised and edited the technical content.
03/14/2008	0.2.4	Editorial	Revised and edited the technical content.
05/16/2008	0.2.5	Editorial	Revised and edited the technical content.
06/20/2008	0.2.6	Editorial	Revised and edited the technical content.
07/25/2008	0.2.7	Editorial	Revised and edited the technical content.
08/29/2008	0.2.8	Editorial	Revised and edited the technical content.
10/24/2008	0.2.9	Editorial	Revised and edited the technical content.
12/05/2008	0.3	Minor	Updated the technical content.
01/16/2009	0.3.1	Editorial	Revised and edited the technical content.
02/27/2009	0.3.2	Editorial	Revised and edited the technical content.
04/10/2009	0.4	Minor	Updated the technical content.
05/22/2009	0.5	Minor	Updated the technical content.
07/02/2009	0.5.1	Editorial	Revised and edited the technical content.
08/14/2009	0.5.2	Editorial	Revised and edited the technical content.
09/25/2009	0.6	Minor	Updated the technical content.
11/06/2009	0.6.1	Editorial	Revised and edited the technical content.
12/18/2009	0.6.2	Editorial	Revised and edited the technical content.
01/29/2010	0.7	Minor	Updated the technical content.
03/12/2010	0.7.1	Editorial	Revised and edited the technical content.
04/23/2010	1.0	Major	Updated and revised the technical content.
06/04/2010	1.0.1	Editorial	Revised and edited the technical content.
07/16/2010	2.0	Major	Significantly changed the technical content.

Release: Friday, October 25, 2013

Date	Revision History	Revision Class	Comments
08/27/2010	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
05/06/2011	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
06/17/2011	2.1	Minor	Clarified the meaning of the technical content.
09/23/2011	2.1	No change	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	3.0	Major	Significantly changed the technical content.
03/30/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/31/2013	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
08/08/2013	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	3.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1   Introduction

In many types of communication, there are often repeated **strings**, both within a document and within the context of related documents. For example, a request for a record set yields a list of records in one document or a series of documents, each bearing a record. Each record uses the same application-defined **XML** tags. This pattern of repeated custom XML tags would be repeated each time as a fully expanded **string** using [MC-NBFS].

This specification defines the .NET Binary Format: SOAP Extension, which is a new format built by extending the format specified in [MC-NBFS]. [MC-NBFS] specifies a way to efficiently encode **strings** that are common to many **SOAP messages**, but it does not specify a way to efficiently encode repeated **strings** that are custom to applications. The purpose of this specification is to define a mechanism by which **strings** may be transmitted once and referred to by subsequent documents. Thus, applications with some recurrence of **strings** may result in documents that are smaller using [MC-NBFSE].

The .NET Binary Format: SOAP Extension does not specify a context in which documents are related. Rather, it is designed to work within the .NET Message Framing Protocol, as specified in [MC-NMF], which provides a context that defines related documents.

Sections 1.7 and 2 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. All other sections and examples in this specification are informative.

1.1   Glossary

The following terms are defined in [MS-GLOS]:

.NET Framework
XML

The following terms are defined in [MC-NBFX]:

DictionaryString
MultiByteInt31
record
String

The following terms are specific to this document:

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2   References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

A reference marked "(Archived)" means that the reference document was either retired and is no longer being maintained or was replaced with a new document that provides current implementation details. We archive our documents online [Windows Protocol].

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MC-NBFS] Microsoft Corporation, ".NET Binary Format: SOAP Data Structure".

[MC-NBFX] Microsoft Corporation, .NET Binary Format: XML Data Structure".

[MC-NMF] Microsoft Corporation, ".NET Message Framing Protocol".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[SOAP1.2-1/2007] Gudgin, M., Hadley, M., Mendelsohn, N., et al., "SOAP Version 1.2 Part 1: Messaging Framework (Second Edition) ", W3C Recommendation 27, April 2007, http://www.w3.org/TR/2007/REC-soap12-part1-20070427/

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

1.3 Overview

This .NET Binary Format: SOAP Extension is used to efficiently represent SOAP documents, as specified in [SOAP1.2-1/2007]. When multiple documents are being processed, this format may be more efficient than the .NET Binary Format: SOAP Data Structure, as specified in [MC-NBFS].

1.4 Relationship to Protocols and Other Structures

The .NET Binary Format: SOAP Extension uses the .NET Binary Format: SOAP Data Structure, as specified in [MC-NBFS].

The .NET Message Framing Protocol, as specified in [MC-NMF], uses the .NET Binary Format: SOAP Extension.

1.5 Applicability Statement

The .NET Binary Format: SOAP Extension is a general-purpose way to process a series of SOAP documents, and is applied in the same way as specified in [MC-NBFS] section 1.5. Additionally, the format is particularly well-suited for environments where a series of SOAP documents that share a common or repetitive vocabulary are processed.

1.6 Versioning and Localization

For information on versioning and localization, see [MC-NBFS] section 1.6.

1.7 Vendor-Extensible Fields

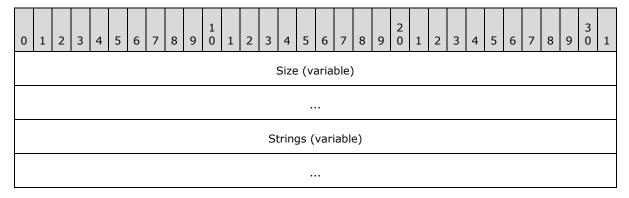The .NET Binary Format: SOAP Extension has no vendor-extensible fields.

2   Structures

The structures in the .NET Binary Format: SOAP Extension are identical to those specified in [MC-NBFS], except for the addition of one new structure and a further extension of the DictionaryString structure, as specified in [MC-NBFS] section 2.1.

The format is identical to the .NET Binary Format: SOAP Data Structure, as specified in [MC-NBFS], except that the document MUST be preceded by a StringTable structure, as specified in section 2.1.

2.1   StringTable

The StringTable structure describes a set of new **strings** to which subsequent **records** can refer. As specified in [MC-NBFS] (section 2.2), the static DictionaryStrings defined by [MC-NBFS] map even integers to a static set of characters, leaving odd integers for extension. The StringTable structure takes advantage of the extensibility point by defining a mapping of odd integers to strings.
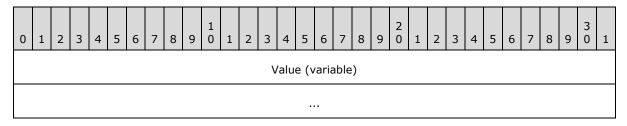
0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Size (variable)																															
...																															
Strings (variable)																															
...																															

Size (variable): An integer value that is encoded by using the **MultiByteInt31** structure that indicates the overall size in bytes of the String structures that follow.

Strings (variable): A series of strings. The **Strings** MUST fit exactly within the **Size** specified. Each **String** MUST be unique and MUST NOT have been present in a prior StringTable.

The first **String** of the first StringTable is assigned an ID of 1, and each subsequent **String** is assigned the next-higher odd number. A consumer of this format MUST maintain this mapping until there are no further documents to process. Subsequent records and documents will refer to this **String** by this ID, as specified in section 2.2.

2.2   DictionaryString

The DictionaryString structure describes a reference to a set of characters.

0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Value (variable)																															
...																															

Value (variable): An integer value that is encoded by using the MultiByteInt31 structure. If this value is even, then it MUST be interpreted as specified in [MC-NBFS] section 2. If this value is

odd, then it MUST refer to a **String** from a StringTable structure that has been processed and MUST be interpreted as the set of characters identified by that **String**.

3 Structure Examples

The following example is of a SOAP document that uses the .NET Binary Format: SOAP Extension format.

Note that the document is the same document that is encoded in [MC-NBFS] section 3.

```
<s:Envelope xmlns:a="http://www.w3.org/2005/08/addressing"
            xmlns:s="http://www.w3.org/2003/05/soap-envelope">
    <s:Header>
        <a:Action s:mustUnderstand="1">action</a:Action>
    </s:Header>
    <s:Body>
        <Inventory>0</Inventory>
    </s:Body>
</s:Envelope>
```

Because the content "action" in the <Action> element is not present in the string structure as specified in [MC-NBFS] section 2, the string was written out fully.

Bytes (hex)	Records	Characters represented
99 06 61 63 74 69 6F 6E	Chars8TextWithEndElement	action</a:Action>

Similarly, because the Inventory string was also not present, the <Inventory> element was written out verbosely.

Bytes (hex)	Records	Characters represented
40 09 49 6E 76 65 6E 74 6F 72 79	ShortElement	<Inventory>

If this document, or a similar document, is processed repeatedly, then it is advantageous not to have to repeat these bytes.

Using the .NET Binary Format: SOAP Extension, a StringTable (section 2.1) is used first to describe the two strings.

- action

- Inventory

According to the StringTable structure, these strings MUST be encoded as shown in the following table. The Bytes column contains the bytes of the structure shown in hex, while the Chars column contains the same bytes shown as characters.

Bytes (hex)	Chars
11 06 61 63 74 69 6F 6E **09** 49 6E 76 65 6E 74 6F 72 79	..action.Inventory

The bolded items in the preceding table are the size (0x11) of the overall structure encoded using MultiByteInt31 and the start of the two strings (0x06, 0x09) encoded using MultiByteInt31 as well. The string "action" is assigned a value of 1 and Inventory is assigned a value of 3, as specified in section 2.

Now, the two preceding records can be encoded more compactly, as shown in the following tables.

Bytes (hex)	Records	Characters represented
AB **01**	DictionaryTextWithEndElement	action</a:Action>

Bytes (hex)	Records	Characters represented
42 **03**	ShortDictionaryElement	<Inventory>

The references to the string from the StringTable are in bold.

4   Security Considerations

The .NET Binary Format: SOAP Extension has the same security considerations as the .NET Binary Format: SOAP Data Structure, as specified in [MC-NBFS] section 4, and the .NET Binary Format: XML Data Structure, as specified in [MC-NBFX] section 4.

Consumers of this format should consider limiting the size of the StringTable (section 2.1) structures that are processed. In addition to requiring the consumer to allocate memory to store the information, StringTable structures represent a way to refer to a potentially large piece of information by using a small piece of information. Without any limits, it might be possible for a malicious producer to leverage this behavior and devise a document that causes an unsuspecting consumer to use significantly more memory than expected.

5   Appendix A: Product Behavior

This document specifies version-specific details in the Microsoft .NET Framework. The following versions of .NET Framework are available in the following released Windows product or as supplemental software, see **.NET Framework**.

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft .NET Framework 3.0

- Microsoft .NET Framework 3.5

- Microsoft .NET Framework 4.0

- Microsoft .NET Framework 4.5

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

6 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

7 Index