[MS-FSXTAPI]: XML-RPC Translatable API Structure

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.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail 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
02/19/2010	1.0	Major	Initial Availability
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Editorial	Revised and edited the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.04	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.0	Major	Significantly changed the technical content.
04/11/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the XML-RPC Translatable API Structure, which represents remote method calls, as described in [XML-RPC]. It specifies how this API translates to an XML body structure for those method calls.

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]:

Augmented Backus-Naur Form (ABNF)
base64

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

datetime
XML schema

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 technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

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.

[ISO-8601] International Organization for Standardization, "Data Elements and Interchange Formats - Information Interchange - Representation of Dates and Times", ISO/IEC 8601:2004, December 2004, http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=40874&ICS1=1&ICS2=140&ICS3=30

Note There is a charge to download the specification.

[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

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, http://www.rfc-editor.org/rfc/rfc5234.txt

[XML-RPC] Winer, D., "XML-RPC Specification", June 1999, http://www.xmlrpc.com/spec

[XMLSCHEMA] World Wide Web Consortium, "XML Schema", September 2005, http://www.w3.org/2001/XMLSchema

1.2.2 Informative References

[MS-FSCADM] Microsoft Corporation, "Crawler Administration and Status Protocol Specification".

[MS-FSCX] Microsoft Corporation, "Configuration (XML-RPC) Protocol Specification".

[MS-FSNC] Microsoft Corporation, "Node Controller Protocol Specification".

[MS-FSSPRADM] Microsoft Corporation, "SPRel Administration and Status Protocol Specification".

[MS-FSWAADM] Microsoft Corporation, "WebAnalyzer Administration and Status Protocol Specification".

[MS-FSWASMT] Microsoft Corporation, "WebAnalyzer and SPRel Multinode Transport Protocol Specification".

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

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

1.3 Structure Overview (Synopsis)

This document specifies an application program interface (API) syntax representing remote method calls, as described in [XML-RPC]. It specifies how this remote method call API translates to an XML body structure for those method calls.

1.4 Relationship to Protocols and Other Structures

The interface syntax described in this document is implemented by the protocols as described in [MS-FSCADM], [MS-FSCX], [MS-FSNC], [MS-FSSPRADM], [MS-FSWAADM], and [MS-FSWASMT].

1.5 Applicability Statement

The interface syntax specified in this document is designed for specifying protocols based on remote method calls, as described in [XML-RPC].

1.6 Versioning and Localization

None.

1.7 Vendor-Extensible Fields

None.

2 Structures

This section specifies the syntax for remote method call APIs, and how these APIs map to the **XML schema**, as specified in [XMLSCHEMA].

2.1 XML-RPC API Syntax

The remote method call API MUST use the **Augmented Backus-Naur Form (ABNF)** rules, as specified in [RFC5234]. The following ABNF grammar specifies those syntactic rules.

```
; Specify XML-RPC API syntax
METHODCALL    = METHODCALL1 / METHODCALL2
; Method call with no method response
METHODCALL1   = METHODNAME "(" PARAMETERS ")" *1(%x3b) CRLF
; Method call with a defined a method response
METHODCALL2   = RESPONSE 1*SP METHODNAME "(" PARAMETERS ")" *1(%x3b) CRLF
; Define method call response
RESPONSE      = "i4" / "int" / "double" / "string" / "array" / "struct" / "base64" /
"dateTime.iso8601" / "boolean" / USER-RESP
USER-RESP     = 1*TOKEN
; Define method call parameters
PARAMETERS    = *1(PARAMETER *SP *( "," *SP PARAMETER))
PARAMETER     = INT-PARAM / STRING-PARAM / DOUBLE-PARAM / ARRAY-PARAM / STRUCT-PARAM / BASE64-
PARAM / DATE-PARAM / USER-PARAM
; Define parameter data types
INT-PARAM     = ("int" / "i4") SP TOKEN *1("=" 1*DIGIT)
BOOL-PARAM    = "boolean" SP TOKEN *1("=" 1*1DIGIT)
STRING-PARAM = "string" SP TOKEN *1("=" %x22 *SCHAR %x22)
DOUBLE-PARAM = "double" SP TOKEN *1("=" 1*DIGIT "." 1*DIGIT)
ARRAY-PARAM  = "array" SP TOKEN *1("=" "None")
STRUCT-PARAM = "struct" SP TOKEN *1("=" "None")
BASE64-PARAM = "base64" SP TOKEN *1("=" "None")
DATE-PARAM    = "dateTime.iso8601"  SP TOKEN
USER-PARAM    = ((TOKEN SP TOKEN *1("=" "None")) / (TOKEN *1("=" "None")))
METHODNAME    = 1*TOKEN
TOKEN         = 1*(%x61-7A /%x41-5A / DIGIT / "_" / "." / ":" / "/")
SCHAR         = %x21 / %x23-7E
```

The following table details the preceding ABNF rules, as specified in [RFC5234].

Rule name	Meaning
METHODCALL	Specifies a remote method call.
METHODCALL1	Specifies a remote method call with no method response.
METHODCALL2	Specifies a remote method call with a method response.
RESPONSE	Specifies the response to the remote method call. A response MUST be one of the following data types: **i4**, **int**, double, string, array, **struct**, **base64**, or **dateTime.iso8601**, as specified in [RFC5234], or the user-defined data type **USER-RESP** specified in this section. If processing of the remote method call fails, the response MUST contain a fault response, as specified in section 2.2.2.3.
USER-RESP	Specifies a user-defined remote method call response. If a user response is used, the text that precedes or follows the remote method call declaration MUST specify the data type. The data type of the user response MUST be one of the following types: **i4**, **int**,

Rule name	Meaning
	double, string, array, **struct**, **base64**, or **dateTime.iso8601**, as specified in [RFC5234].
METHODNAME	Specifies the name of the remote method call.
PARAMETERS	Specifies the parameters of the remote method call.
PARAMETER	Specifies a parameter.
INT-PARAM	Specifies an integer parameter, as specified in [RFC5234]. The protocol assigns a default integer value to this parameter.
BOOL-PARAM	Specifies a Boolean parameter, as specified in [RFC5234]. The protocol assigns a default Boolean value to this parameter.
STRING-PARAM	Specifies a string parameter, as specified in [RFC5234]. The protocol assigns a default string value to this parameter.
DOUBLE-PARAM	Specifies a parameter of the type double, as specified in [RFC5234]. The protocol assigns a default floating number value to this parameter.
ARRAY-PARAM	Specifies an array parameter, as specified in [RFC5234]. If an array parameter is used, the text that precedes or follows the remote method call declaration MUST specify the array. The protocol specifies a default empty array with the value "None".
STRUCT-PARAM	Specifies a **structure** parameter, as specified in [RFC5234]. If a structure is used, the text that precedes or follows the remote method call declaration MUST specify the structure. The protocol specifies a default empty structure with the value "None".
BASE64-PARAM	Specifies a parameter of the type **base64**, as specified in [RFC5234]. The protocol specifies a default empty **base64** encoded binary with the value "None".
DATE-PARAM	Specifies a parameter of the type [ISO-8601] **dateTime.iso8601** type, as specified in [RFC5234].
USER-PARAM	Specifies a user-defined parameter. If a user-defined parameter is used, the preceding or following text of the remote method call declaration MUST specify the data type of the parameter, which is one of the following types: **i4**, **int**, double, string, array, **struct**, **base64**, or **dateTime.iso8601**, as specified in [RFC5234].

2.2 XML-RPC Mapping

This section specifies how to map the ABNF rules, specified in section 2.1, to the XML schema for remote method calls.

2.2.1 Global Elements

2.2.1.1 methodCall

This element specifies a remote method call element, as specified in [XML-RPC].

```
<xs:element name="methodCall" type="CT_methodCall"/>
```

2.2.1.2 methodResponse

This element specifies a **methodResponse** element, as specified in [XML-RPC].

```
<xs:element name="methodResponse" type="CT_methodResponse"/>
```

2.2.2 Complex Types

2.2.2.1 CT_array

This complex type referenced by the **CT_type** element specifies an array.

This complex type is defined as follows:

```
<xs:complexType name="CT_array">
  <xs:sequence>
    <xs:element name="data" type="CT_data" />
  </xs:sequence>
</xs:complexType>
```

data: A **CT_data** element that specifies the elements of the array.

2.2.2.2 CT_data

This complex type referenced by the **CT_array** element specifies the elements of an array.

This complex type is defined as follows:

```
<xs:complexType name="CT_data">
   <xs:sequence>
     <xs:element name="value"  type="CT_type"
                 minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
```

value: A **CT_type** element that specifies an array element.

2.2.2.3 CT_fault

This complex type referenced by the **CT_methodResponse** element specifies the fault resulting from a method call that received an error.

This complex type is defined as follows:

```
<xs:complexType name ="CT_fault">
   <xs:sequence>
     <xs:element name="value" type="CT_fault_value" />
   </xs:sequence>
</xs:complexType>
```

value: A **CT_fault_value** element that specifies the fault of the method call.

2.2.2.4 CT_fault_value

This complex type referenced by the **CT_fault** element specifies the cause of the fault that resulted from the remote method call.

This complex type is defined as follows:

```
<xs:complexType name="CT_fault_value">
   <xs:sequence>
     <xs:element name="struct" type="CT_struct"
                 maxOccurs="unbounded"/>
   </xs:sequence>
</xs:complexType>
```

struct: A **CT_struct** element that specifies the cause of the fault. The **faultCode** and **faultString** elements contain information about the cause of the fault, as specified in [RFC5234].

2.2.2.5 CT_member

This complex type referenced by the **CT_struct** element specifies a member of a **structure**.

This complex type is defined as follows:

```
<xs:complexType name="CT_member">
  <xs:sequence>
    <xs:element name="name"  type="xs:string" />
    <xs:element name="value" type="CT_type"/>
  </xs:sequence>
</xs:complexType>
```

name: A **CT_member** element that specifies the data type of the member.

value: A **CT_type** element that specifies the data type of the parameter.

2.2.2.6 CT_methodCall

This complex type referenced by the **methodCall** element specifies a call to a remote method.

This complex type is defined as follows:

```
<xs:complexType name="CT_methodCall">
  <xs:all>
    <xs:element name="methodName" type="ST_methodName" minOccurs="1" maxOccurs="1"/>
    <xs:element name="params" type="CT_params" minOccurs="1" maxOccurs="1">
    </xs:element>
  </xs:all>
</xs:complexType>
```

methodName: A **ST_methodName** element that specifies the name of the method call.

params: A **CT_params** element that specifies the parameters of the method call.

2.2.2.7 CT_methodResponse

This complex type referenced by the **methodResponse** element specifies the response from the method call.

This complex type is defined as follows:

```
<xs:complexType name="CT_methodResponse">
  <xs:all>
    <xs:element name="params" type="CT_params" minOccurs="1" maxOccurs="1">
    </xs:element>
  </xs:all>
</xs:complexType>
```

params: A **CT_params** element that specifies the returned parameters.

2.2.2.8 CT_param

This complex type referenced by the **CT_params** element specifies a parameter, as specified in section 2.1.

This complex type is defined as follows:

```
<xs:complexType name="CT_param">
  <xs:sequence>
    <xs:element name="value" type="CT_type"/>
  </xs:sequence>
</xs:complexType>
```

value: A **CT_type** element that specifies the data type of the parameter.

2.2.2.9 CT_params

This complex type referenced by the **CT_methodCall** and **CT_methodResponse** elements contains the parameters, as specified in section 2.1.

This complex type is defined as follows:

```
<xs:complexType name="CT_params">
  <xs:sequence>
    <xs:element name="param" type="CT_param"
                minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

param: A **CT_param** element that specifies a parameter.

2.2.2.10 CT_struct

This complex type referenced by the **CT_type** element specifies a **structure**.

This complex type is defined as follows:

```
<xs:complexType name="CT_struct">
```

```
    <xs:sequence>
      <xs:element name="member" type="CT_member"
                  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
```

member: A **CT_member** element that specifies a member of the **structure**.

2.2.2.11 CT_type

This complex type referenced by the **CT_param** element specifies a data type.

This complex type is defined as follows:

```
  <xs:complexType name="CT_type" mixed="true">
    <xs:choice>
      <xs:element name="i4" type="xs:int"/>
      <xs:element name="int" type="xs:int"/>
      <xs:element name="string" type="xs:string"/>
      <xs:element name="double" type="xs:decimal"/>
      <xs:element name="base64" type="xs:base64Binary"/>
      <xs:element name="boolean" type="ST_boolean"/>
      <xs:element name="dateTime.iso8601" type="xs:dateTime"/>
      <xs:element name="array" type="CT_array"/>
      <xs:element name="struct" type="CT_struct"/>
    </xs:choice>
  </xs:complexType>
```

i4: Specifies that the parameter data type is an integer.

int: Specifies that the parameter data type is an integer.

string: Specifies that the parameter data type is a string.

double: Specifies that the parameter data type is a floating point.

base64: Specifies that the parameter data type uses base64 encoding.

boolean: An **ST_boolean** element that specifies a Boolean parameter data type.

dateTime.iso8601: Specifies that the parameter data type is a **datetime**.

array: A **CT_array** element that specifies an array for the parameter.

struct: A **CT_struct** element that specifies a structure for the parameter.

2.2.3 Simple types

2.2.3.1 ST_boolean

This simple type referenced by the **CT_type** element specifies a **Boolean** value.

```
  <xs:simpleType name="ST_boolean">
    <xs:restriction base="xs:boolean">
      <xs:pattern value="0|1" />
    </xs:restriction>
  </xs:simpleType>
```

2.2.3.2 ST_methodName

This simple type referenced by the **CT_methodCall** element specifies a string value. This string is the name of the method call, as specified in section 2.1.

```
    <xs:simpleType name="ST_methodName">
      <xs:restriction base="xs:string">
      </xs:restriction>
    </xs:simpleType>
```

3 Structure Examples

3.1 SimpleMethodCall

This example describes a simple remote method call.

```
SimpleMethodCall(int i=1711);
```

The remote method call results in the following XML.

```
<?xml version="1.0"?>
<methodCall>
  <methodName>SimpleMethodCall</methodName>
  <params>
    <param>
        <value><int>1711</int></value>
    </param>
  </params>
</methodCall>
```

3.2 MethodCall

This example describes a remote method call. In this example, the return value named **res** is of type string.

```
res MethodCall(int i, string j);
```

This call results in the following XML for the remote method call.

```
<?xml version="1.0"?>
<methodCall>
  <methodName>MethodCall</methodName>
  <params>
    <param>
        <value><int>"value_of_the_i_parameter"</int></value>
        <value><string>"value_of_the_j_parameter"</string></value>
    </param>
  </params>
</methodCall>
```

The following is the response from the method call.

```
<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
        <value><string>"value_of_the_returned_string"</string></value>
    </param>
  </params>
</methodResponse>
```

4 Security Considerations

None.

5 Appendix A: XML-RPC XML Schema

For ease of implementation, the following complete XML schema for the remote method call API is provided.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="methodCall" type="CT_methodCall"/>

  <xs:element name="methodResponse" type="CT_methodResponse"/>

  <xs:complexType name="CT_methodCall">
    <xs:all>
      <xs:element name="methodName" type="ST_methodName"
                  minOccurs="1" maxOccurs="1"/>
      <xs:element name="params" type="CT_params"
                  minOccurs="1" maxOccurs="1"/>
    </xs:all>
  </xs:complexType>

  <xs:complexType name="CT_methodResponse">
    <xs:all>
      <xs:element name="params" type="CT_params"
                  minOccurs="0" maxOccurs="1"/>
      <xs:element name="fault" type="CT_fault"
                  minOccurs="0" maxOccurs="1"/>
    </xs:all>
  </xs:complexType>

  <xs:complexType name="CT_params">
    <xs:sequence>
      <xs:element name="param"  type="CT_param"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CT_param">
  <xs:sequence>
    <xs:element name="value" type="CT_type"/>
  </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CT_type" mixed="true">
    <xs:choice>
      <xs:element name="i4" type="xs:int"/>
      <xs:element name="int" type="xs:int"/>
      <xs:element name="string" type="xs:string"/>
      <xs:element name="double" type="xs:decimal"/>
      <xs:element name="base64" type="xs:base64Binary"/>
      <xs:element name="boolean" type="ST_boolean"/>
      <xs:element name="dateTime.iso8601" type="xs:dateTime"/>
      <xs:element name="array" type="CT_array"/>
      <xs:element name="struct" type="CT_struct"/>
    </xs:choice>
  </xs:complexType>

  <xs:complexType name="CT_struct">
```

```
    <xs:sequence>
      <xs:element name="member" type="CT_member"
                   maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CT_member">
    <xs:sequence>
      <xs:element name="name"  type="xs:string" />
      <xs:element name="value" type="CT_type"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CT_array">
    <xs:sequence>
      <xs:element name="data" type="CT_data" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="CT_data">
    <xs:sequence>
      <xs:element name="value"  type="CT_type"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CT_fault_value">
    <xs:sequence>
      <xs:element name="struct" type="CT_struct"
                   maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name ="CT_fault">
      <xs:sequence>
        <xs:element name="value" type="CT_fault_value"/>
      </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ST_methodName">
    <xs:restriction base="xs:string">
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="ST_boolean">
    <xs:restriction base="xs:boolean">
      <xs:pattern value="0|1" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

6 Appendix B: Product Behavior

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

- Microsoft® FAST™ Search Server 2010

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.

7 Change Tracking

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

8 Index