Bindings for OBIX: REST Bindings Version 1.0

Committee Specification Draft ~~02~~03 / Public Review Draft ~~02~~03

~~19 December 2013~~

06 November 2014

Specification URIs

This version:

http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd03/obix-rest-v1.0-csprd03.pdf (Authoritative)
http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd03/obix-rest-v1.0-csprd03.html
http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd03/obix-rest-v1.0-csprd03.doc

Previous version:

http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd02/obix-rest-v1.0-csprd02.pdf (Authoritative)
http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd02/obix-rest-v1.0-csprd02.html
http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd02/obix-rest-v1.0-csprd02.doc

Latest version:

http://docs.oasis-open.org/obix/obix-rest/v1.0/obix-rest-v1.0.pdf (Authoritative)
http://docs.oasis-open.org/obix/obix-rest/v1.0/obix-rest-v1.0.html
http://docs.oasis-open.org/obix/obix-rest/v1.0/obix-rest-v1.0.doc

Technical Committee:

OASIS Open Building Information Exchange (oBIX) TC

Chair:

Toby Considine (toby.considine@unc.edu), University of North Carolina at Chapel Hill

Editors:

Craig Gemmill (craig.gemmill@tridium.com), Tridium, Inc.
Markus Jung (mjung@auto.tuwien.ac.at), Institute of Computer Aided Automation, Vienna University of Technology~~,~~

Related work:

This specification is related to:

- *OBIX Version 1.1*. Edited by Craig Gemmill. Latest version. http://docs.oasis-open.org/obix/obix/v1.1/obix-v1.1.html.
- *Encodings for OBIX: Common Encodings Version 1.0.* Edited by Markus Jung. Latest version. http://docs.oasis-open.org/obix/obix-encodings/v1.0/obix-encodings-v1.0.html.
- *Bindings for OBIX: SOAP Bindings Version 1.0.* Edited by Markus Jung. Latest version~~.~~. http://docs.oasis-open.org/obix/obix-soap/v1.0/obix-soap-v1.0.html.
- *Bindings for OBIX: ~~Web Socket~~WebSocket Bindings Version 1.0*. Edited by Matthias Hub. Latest version. http://docs.oasis-open.org/obix/obix-websocket/v1.0/obix-websocket-v1.0.html~~.~~
- ~~*Encodings for OBIX: Common Encodings Version 1.0.* Edited by Marcus Jung. Latest version.~~.

Abstract:

This document specifies REST bindings for OBIX. OBIX provides the core information model and interaction pattern for communication with building control systems. Specific implementations of OBIX must choose how to bind OBIX interactions. This document describes the REST Binding, an interaction pattern that can be used in conjunction with XML, EXI, CoAP, and JSON encodings, as well as other encodings that may be specified elsewhere.

Status:

This document was last revised or approved by the OASIS Open Building Information Exchange (oBIX) TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=obix#technical.

~~Technical Committee~~TC members should send comments on this specification to the ~~Technical Committee's~~TC's email list. Others should send comments to the ~~Technical Committee~~TC's public comment list, after subscribing to it by ~~using~~following the ~~"~~instructions at the "Send A Comment" button on the ~~Technical Committee's~~TC's web page at https://www.oasis-open.org/committees/obix/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (https://www.oasis-open.org/committees/obix/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[OBIX-REST-v1.0]

Bindings for OBIX: REST Bindings Version 1.0. Edited by Craig Gemmill and Markus Jung. ~~19 December 2013.~~06 November 2014. OASIS Committee Specification Draft ~~02~~03 / Public Review Draft ~~02.~~ 03. http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd03/obix-rest-v1.0-csprd03.html. Latest version: http://docs.oasis-open.org/obix/obix-rest/v1.0/obix-rest-v1.0.html.

Notices

Table of Contents

Table of Tables

1 Introduction

This document specifies the REST bindings for OBIX.

1.1 Terminology

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**. When used in the non-capitalized form, these words are to be interpreted with their normal English meaning.

1.2 Normative References

RFC2119	Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.
OBIX 1.1	*OBIX Version 1.1.* See link in "Related work" section on cover page.
RFC2616	Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., “Hypertext Transfer Protocol – HTTP/1.1”, RFC2616, June 1999. http://www.ietf.org/rfc/rfc2616.txt.
RFC2617	Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., Stewart, L., “HTTP Authentication: Basic and Digest Access Authentication”, RFC2617, June 1999. http://www.ietf.org/rfc/rfc2617.txt.
RFC2818	Rescorla, E., “HTTP over TLS”, RFC 2818, May 2000. http://www.ietf.org/rfc/rfc2818.txt**RFC2618** Aboba, B., Zorn, G., “RADIUS Authentication Client MIB”, RFC2618, June 1999. . .
RFC2246	Dierks, T., Allen, C., “The TLS Protocol”, RFC2246, January 1999. http://www.ietf.org/rfc/rfc2246.txt.
RFC4346	Dierks, T., Rescorla, E., “The Transport Layer Security (TLS) Protocol Version 1.1”, RFC4346, April 2006. http://www.ietf.org/rfc/rfc4346.txt.
RFC5246	Dierks, T. and E. Rescorla, ""The Transport Layer Security (TLS) Protocol Version 1.2"", RFC 5246, August 2008. http://www.ietf.org/rfc/rfc5246.txt**COAP**. Shelby, Z., Hartke, K., Bormann, C., “Constrained Application Protocol (CoAP)”, IETF Internet Draft, Version 18, 28 June 2013.
OBIX Encodings	*Encodings for OBIX: Common Encodings Version 1.0*. See link in "Related work" section on cover page.

1.3 Non-Normative References

REST	**RT Fielding** *Architectural Styles and the Design of Network-based Software Architectures,* Dissertation, University of California at Irvine, 2000, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm
CoAP	Shelby, Z., Hartke, K., Bormann, C., “The Constrained Application Protocol (CoAP)”, IETF Internet Draft, June 2014. http://tools.ietf.org/search/rfc7252
CoAP-OBSERVE	Hartke, K., “Observing Resources in CoAP”, IETF Internet-Draft 08, February 25, 2013Version 15, October 27, 2014. http://www.ietf.org/id/draft-ietf-core-observe-15.txt
OBIX 1.1	*OBIX Version 1.1.* See link in "Related work" section on cover page.

1.4 Editing Conventions

All sections of this specification SHALL be considered normative, unless specifically identified as non-normative.

2 HTTP Binding

2.1 Description

The HTTP binding specifies a simple REST mapping of OBIX requests to HTTP. A read request is a simple HTTP GET, which means that you can simply read an Object by typing its URI into your browser. Refer to "**RFC2616**" for the full specification of HTTP 1.1.

~~2.1~~2.2 Requests

The following table summarizes how OBIX requests map to HTTP methods:

OBIX Request	HTTP Method	Target
Read	GET	Any Object with an href
Write	PUT	Any Object with an href and `writable`=true
Invoke	POST	Any op Object
Delete	DELETE	Any Object with an href and `writable`=true

Table 2-1. Mapping of OBIX Requests to HTTP Methods.

The URI used for an HTTP request MUST map to the URI of the Object being read, written, or invoked. Read requests use a simple HTTP GET and return the resulting OBIX document. Write and invoke are implemented with the PUT and POST methods respectively. The input is passed to the server as an OBIX document and the result is returned as an OBIX document.

If the OBIX server processes a request, then it MUST return the resulting OBIX document with an HTTP status code of 200 OK. The 200 status code MUST be used even if the request failed and the server is returning an `err` Object as the result.

~~2.2 MIME Type~~

2.3 Content Negotiation

The HTTP client MAY specify the MIME type of the encoding according to the **OBIX Encodings** specification for the payload of a PUT or POST request using the HTTP content type header.

~~2.2.1 Content Negotiation~~

OBIX resources MUST be encoded using MIME types defined by the corresponding encodingas defined by the **OBIX Encodings** specification. Clients and servers SHOULD follow Section 12 of **RFC2616** for content negotiation.

If a client wishes to GET a resource using a specific encoding, then it SHOULD specify the desired MIME type in the Accept header.

If the server does not support the MIME type of a client request, then it SHOULD respond with the 406 Not Acceptable status code. There are two use cases for a 406 failure: 1) the client specifies an unsupported MIME type in the Accept header of a GET (read) request, or 2) the client specifies an unsupported MIME type in the Content-Type of a PUT (write) or POST (invoke) request.

~~2.3~~2.4 Security

Numerous standards are designed to provide authentication and encryption services for HTTP. Existing standards SHOULD be used when applicable for OBIX HTTP implementations including:

- **RFC2617** - HTTP Authentication: Basic and Digest Access Authentication
- **RFC2818** - HTTP Over TLS (HTTPS)
- **RFC5246**~~RFC /~~ – The TLS Protocol (Transport Layer Security~~)~~. An OBIX HTTP implementation MAY support superseded versions of this standard, including **RFC2246** and **RFC4346**~~)~~.

~~2.4~~2.5 Localization

Servers SHOULD follow the localization approach outlined in the core OBIX Specification. If the desired locale of the client cannot be determined through authentication, it SHOULD be determined via the Accept-Language HTTP header. As a fallback, the locale MAY be derived from the Accept-Language header.

3 CoAP Binding

3.1 Description

The Constrained Application Protocol (CoAP) is a specialized Web transfer protocol for use within constrained nodes and constrained (e.g., low-power, lossy) networks [CoAP]. CoAP is designed for nodes operated by microcontrollers and networks such as 6LoWPAN, which often have a high packet error rate and low bandwidth (10s of kbits/s). It is intended to be used within building automation systems.

CoAP can be seen as optimized HTTP equivalent that uses UDP for packet exchange instead of TCP. Since UDP is a non-reliable packet oriented transport protocol CoAP provides custom facilities for reliable messaging and includes a CoAP specific acknowledgement mechanism to provide reliable point-to-point communication. Through the use of UDP it enables additional interaction patterns like asynchronous and group communication.

~~3.1~~3.2 Requests

The following table summarizes how OBIX requests map to CoAP methods:

OBIX Request	CoAP Method	Target
Read	GET	Any Object with an href
Write	PUT	Any Object with an href and `writable`=true
Invoke	POST	Any op Object
Delete	DELETE	Any Object with an href and `writable`=true

Table 3-1. ***Mapping of OBIX Requests to CoAP Methods.***

~~3.2 MIME Type~~

3.3 Content Negotiation

The CoAP client MAY specify the MIME type of the encoding according to the **OBIX Encodings** specification for the payload of a PUT or POST request using the CoAP header content format option to a value according to the CoAP content-format registry defined by **CoAP** which maps standard MIME types to a numeric value.Content negotiation

OBIX resources may be encoded using either the “text/xml“ or the “application/x-obix-binary“ MIME types defined by the corresponding encoding defined by the **OBIX Encodings** specification. Clients and servers SHOULD follow Section 12 of **RFC2616** for content negotiation.

If a client wishes to GET a resource using a specific encoding, then it SHOULD specify the desired MIME type content-format identifier in the Accept header CoAP header accept option according to the CoAP content-format registry which maps standard MIME types to a numeric value..

If the server does not support the MIME type of a client request, then it SHOULD respond with the 406 Not Acceptable status code. There are two use cases for a 406 failure: 1) the client specifies an unsupported MIME type in the Accept header of a GET (read) request, or 2) the client specifies an unsupported MIME type in the Content-Type of a PUT (write) or POST (invoke) request.

~~3.3~~3.4 Observing resources [non-normative]

An OBIX server that provides a CoAP binding ~~should~~SHOULD also support the CoAP ~~o~~Observe option on CoAP GET requests. This provides an alternative to the concept of OBIX watches, since no polling for updates on a resource is required. If the client issues a CoAP GET request with the ~~o~~Observe option set,

an observation relationship ~~is~~SHOULD be established on the server. If an observed OBIX Object is updated, a CoAP response message ~~is~~SHOULD be sent to the client according to the **CoAP-OBSERVE** specification.

~~3.4~~3.5 Security

For securing the CoAP binding the DTLS binding of CoAP as specified in **CoAP** ~~should~~ SHOULD be used.

4 Conformance

4.1 Conditions for a Conforming Server Binding

An implementation ~~is compliant with~~ conforms to this specification as a Server if it ~~implements~~provides one of the bindings described in this specification, and meets all of the requirements specified in the Section describing that binding. All MUST and REQUIRED elements MUST be implemented in order to comply with the binding specification. In particular, a Server MUST be able to perform content negotiation as described in Sections 1.1 and 1.1~~or~~ to arrive at a common agreement for the MIME type to be used in encoding OBIX requests and responses.

4.2 Conditions for a Conforming Client Binding

An implementation conforms to this specification as a Client if it makes requests using one of the bindings described in this specification, and meets all of the MUST and REQUIRED level requirements described for the client request generation and response processing. In particular, a Client MUST be able to perform content negotiation as described in Sections 1.1 and 1.1 to arrive at a common agreement for the MIME type to be used in encoding OBIX requests and responses.

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Ron Ambrosio, IBM
Brad Benson, Trane
Ron Bernstein, LonMark International*
Rich Blomseth, Echelon Corporation
Anto Budiardjo, Clasma Events, Inc.
Jochen Burkhardt, IBM
JungIn Choi, Kyungwon University
David Clute, Cisco Systems, Inc.*
Toby Considine, University of North Carolina at Chapel Hill
William Cox, Individual
Robert Dolin, Echelon Corporation
Marek Dziedzic, Treasury Board of Canada, Secretariat
Brian Frank, SkyFoundry
Craig Gemmill, Tridium, Inc.
Wonsuk Ko, Kyungwon University
Perry Krol, TIBCO Software Inc.
Corey Leong, Individual
Ulf Magnusson, Schneider Electric
Brian Meyers, Trane
Jeremy Roberts, LonMark International
Thorsten Roggendorf, Echelon Corporation
Anno Scholten, Individual
John Sublett, Tridium, Inc.
Dave Uden, Trane
Ron Zimmer, Continental Automated Buildings Association (CABA)*
Rob Zivney, Hirsch Electronics Corporation
Markus Jung, Institute of Computer Aided Automation, Vienna University of Technology

Appendix B. Revision History

Revision	Date	Editor	Changes Made
wd01	26 Mar 13	Markus Jung	Initial creation with HTTP binding taken out of OBIX 1.1 WD07 working draft.
wd02	27 Mar 2013	Craig Gemmill	Add HTTP DELETE, references
wd03	10 Apr 2013	Craig Gemmill	Upper case SHOULD keywords
wd04	23 May 2013	Markus Jung	First draft on CoAP binding, Updated MIME and content negotiation of HTTP binding to reference the encodings document.
wd05	13 Jun 2013	Markus Jung	Updated CoAP reference
wd06	28 Jun 2013	Markus Jung	Updated reference section
wd07	04 Dec 2013	Craig Gemmill	Localization moved to core spec
wd08	16 Dec 2013	Markus Jung	Merge with changes of Craig
wd09	16 Dec 2013	Markus Jung	PR doc
wd10	5 Nov 2014	Craig Gemmill	Address several PR issues
wd11	6 Nov 2014	Craig Gemmill	Address remaining PR issues
wd12	6 Nov 2014	Craig Gemmill	Fix references in Section 4