Bindings for OBIX: REST Bindings Version 1.0

Committee Specification Draft 02 / Public Review Draft 02

19 December 2013

Specification URIs

This 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

Previous version:

http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd01/obix-rest-v1.0-csprd01.pdf (Authoritative)
http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd01/obix-rest-v1.0-csprd01.html
http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd01/obix-rest-v1.0-csprd01.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.
- *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 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. http://docs.oasis-open.org/obix/obix-encodings/v1.0/obix-encodings-v1.0.html.

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.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at http://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 (http://www.oasis-open.org/committees/obix/ipr.php).

Citation format:

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

[OBIX-REST]

Bindings for OBIX: REST Bindings Version 1.0. Edited by Craig Gemmill and Markus Jung. 19 December 2013. OASIS Committee Specification Draft 02 / Public Review Draft 02. http://docs.oasis-open.org/obix/obix-rest/v1.0/csprd02/obix-rest-v1.0-csprd02.html. Latest version: http://docs.oasis-open.org/obix/obix-rest/v1.0/obix-rest-v1.0.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/policies-guidelines/trademark for above guidance.

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.

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.

RFC2618 Aboba, B., Zorn, G., “RADIUS Authentication Client MIB”, RFC2618, June 1999. http://www.ietf.org/rfc/rfc2618.txt.

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.

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-OBSERVE Hartke, K., “Observing Resources in CoAP”, IETF Internet-Draft 08, February 25, 2013

2 HTTP Binding

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 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

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 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
- **RFC2618** - HTTP Over TLS (HTTPS)

- RFC **RFC4346**/**RFC2246** – The TLS Protocol (Transport Layer Security)

2.4 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

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 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

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 Observing resources

An OBIX server that provides a CoAP binding should also support the CoAP 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 observe option set an observation relationship is established on the server. If an observed OBIX Object is updated a CoAP response message is sent to the client according to the **CoAP-OBSERVE** specification.

3.4 Security

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

4 Conformance

An implementation is compliant with this specification if it implements all MUST or REQUIRED level requirements.

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