



# Bindings for oBIX: REST Bindings Version 1.0

## Committee Specification Draft 01

11 July 2013

### Specification URIs

#### This version:

<http://docs.oasis-open.org/obix/obix-rest/v1.0/csd01/obix-rest-v1.0-csd01.pdf> (Authoritative)  
<http://docs.oasis-open.org/obix/obix-rest/v1.0/csd01/obix-rest-v1.0-csd01.html>  
<http://docs.oasis-open.org/obix/obix-rest/v1.0/csd01/obix-rest-v1.0-csd01.doc>

#### Previous version:

N/A

#### 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](mailto:toby.considine@unc.edu)), University of North Carolina at Chapel Hill

#### Editors:

Craig Gemmill ([craig.gemmill@tridium.com](mailto:craig.gemmill@tridium.com)), Tridium, Inc.  
Markus Jung ([mjung@auto.tuwien.ac.at](mailto: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*. 11 July 2013. OASIS Committee Specification Draft 01. <http://docs.oasis-open.org/obix/obix/v1.1/csd01/obix-v1.1-csd01.html>.
- *Encodings for oBIX: Common Encodings Version 1.0*. 11 July 2013. OASIS Committee Specification Draft 01. <http://docs.oasis-open.org/obix/obix-encodings/v1.0/csd01/obix-encodings-v1.0-csd01.html>.
- *Bindings for oBIX: SOAP Bindings Version 1.0*. 11 July 2013. OASIS Committee Specification Draft 01. <http://docs.oasis-open.org/obix/obix-soap/v1.0/csd01/obix-soap-v1.0-csd01.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*. 11 July 2013. OASIS Committee Specification Draft 01. <http://docs.oasis-open.org/obix/obix-rest/v1.0/csd01/obix-rest-v1.0-csd01.html>.

---

## Notices

Copyright © OASIS Open 2013. All Rights Reserved.

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

1	Introduction .....	5
1.1	Terminology .....	5
1.2	Normative References .....	5
1.3	Non-Normative References .....	5
2	HTTP Binding .....	6
2.1	Requests .....	6
2.2	MIME Type.....	6
2.3	Security .....	6
2.4	Localization .....	7
3	CoAP Binding .....	8
3.1	Requests .....	8
3.2	MIME Type.....	8
3.3	Observing resources.....	8
3.4	Security .....	8
4	Conformance .....	9
Appendix A.	Acknowledgments .....	10
Appendix B.	Revision History .....	11

---

# 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 17, 26 May 2013. <http://www.ietf.org/rfc/rfc?????.txt>
- 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

---

## 36 2 HTTP Binding

37 The HTTP binding specifies a simple REST mapping of oBIX requests to HTTP. A read request is a  
38 simple HTTP GET, which means that you can simply read an object by typing its URI into your browser.  
39 Refer to “[RFC 2616 Hypertext Transfer Protocol](#)” for the full specification of HTTP 1.1.

### 40 2.1 Requests

41 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

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

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

### 49 2.2 MIME Type

50  
51 The HTTP client MAY specify the MIME type of the encoding [oBIX Encodings] for the payload of a PUT  
52 or POST request using the HTTP content type header. Content Negotiation  
53 oBIX resources may be encoded using MIME types defined by the according encoding [oBIX Encodings].  
54 Clients and servers SHOULD follow Section 12 of [RFC 2616](#) for content negotiation.

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

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

### 61 2.3 Security

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

- 64 • [RFC 2617](#) - HTTP Authentication: Basic and Digest Access Authentication
- 65 • [RFC 2818](#) - HTTP Over TLS (HTTPS)
- 66 • [RFC 4346/2246](#) – The TLS Protocol (Transport Layer Security)

## 67 **2.4 Localization**

68 Servers SHOULD localize appropriate data based on the desired locale of the client agent. Localization  
69 SHOULD include the `display` and `displayName` attributes. The desired locale of the client SHOULD  
70 be determined through authentication or via the Accept-Language HTTP header. A suggested algorithm  
71 is to check if the authenticated user has a preferred locale configured in the server's user database, and if  
72 not then fallback to the locale derived from the Accept-Language header.

73

74 Localization MAY include auto-conversion of units. For example if the authenticated user has configured  
75 a preferred unit system such as English versus Metric, then the server might attempt to convert values  
76 with an associated `unit` facet to the desired unit system.

---

## 77 3 CoAP Binding

78 The Constrained Application Protocol (CoAP) is a specialized Web transfer protocol for use within  
79 constrained nodes and constrained (e.g., low-power, lossy) networks [CoAP]. CoAP is designed for  
80 nodes operated by microcontrollers and networks such as 6LoWPAN, which often have a high packet  
81 error rate and low bandwidth (10s of kbits/s). It is intended to be used within building automation systems.  
82 CoAP can be seen as optimized HTTP equivalent that uses UDP for packet exchange instead of TCP.  
83 Since UDP is a non-reliable packet oriented transport protocol CoAP provides custom facilities for reliable  
84 messaging and includes a CoAP specific acknowledgement mechanism to provide reliable point-to-point  
85 communication. Through the use of UDP it enables additional interaction patterns like asynchronous and  
86 group communication.

### 87 3.1 Requests

88 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

### 89 3.2 MIME Type

90 The CoAP client MAY specify the MIME type of the encoding [oBIX Encodings] for the payload of a PUT  
91 or POST request using the CoAP header content format option to a value according to the CoAP content-  
92 format registry [CoAP] which maps standard MIME types to a numeric value. Content negotiation

93 oBIX resources may be encoded using either the "text/xml" or the "application/x-obix-binary" MIME types  
94 defined by the according encoding [oBIX Encodings]. Clients and servers SHOULD follow Section 12 of  
95 RFC 2616 for content negotiation.

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

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

### 103 3.3 Observing resources

104 An oBIX server that provides a CoAP binding should also support the CoAP observe option on CoAP  
105 GET requests. This provides an alternative to the concept of oBIX watches, since no polling for updates  
106 on a resource is required. If the client issues a CoAP GET request with the observe option set an  
107 observation relationship is established on the server. If an observed oBIX object is updated a CoAP  
108 response message is sent to the client according to the [CoAP Observe] specification.

### 109 3.4 Security

110 For securing the CoAP binding the DTLS binding of CoAP as specified in [CoAP] should be used.

---

111 **4 Conformance**

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

---

## 114 Appendix A. Acknowledgments

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

### 117 **Participants:**

118 Ron Ambrosio, IBM  
119 Brad Benson, Trane  
120 Ron Bernstein, LonMark International\*  
121 Rich Blomseth, Echelon Corporation  
122 Anto Budiardjo, Clasma Events, Inc.  
123 Jochen Burkhardt, IBM  
124 JungIn Choi, Kyungwon University  
125 David Clute, Cisco Systems, Inc.\*  
126 Toby Considine, University of North Carolina at Chapel Hill  
127 William Cox, Individual  
128 Robert Dolin, Echelon Corporation  
129 Marek Dziedzic, Treasury Board of Canada, Secretariat  
130 Brian Frank, SkyFoundry  
131 Craig Gemmill, Tridium, Inc.  
132 Wonsuk Ko, Kyungwon University  
133 Perry Krol, TIBCO Software Inc.  
134 Corey Leong, Individual  
135 Ulf Magnusson, Schneider Electric  
136 Brian Meyers, Trane  
137 Jeremy Roberts, LonMark International  
138 Thorsten Roggendorf, Echelon Corporation  
139 Anno Scholten, Individual  
140 John Sublett, Tridium, Inc.  
141 Dave Uden, Trane  
142 Ron Zimmer, Continental Automated Buildings Association (CABA)\*  
143 Rob Zivney, Hirsch Electronics Corporation  
144 Markus Jung, Institute of Computer Aided Automation

145

146

## Appendix B. Revision History

147

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

148

149