



Book Industry Communication

BIC Realtime

Standards for Instant Business Message Exchange

ONIX Product Information Request and Response

Version 2.0, 3 April 2020

This document: <https://www.bic.org.uk/files/pdfs/API/Trade/BICWSONIXProductInformation-V2.0.pdf>

XML schema: https://www.bic.org.uk/files/xml/BICWSONIXProductInformation_V2.0.xsd

WSDL file: https://www.bic.org.uk/files/xml/BICWSONIXProductInformationSOAP_V2.0.wsdl

XML namespace: <https://www.bic.org.uk/webservices/onixProductInfo>

Next review date: 26 February 2021

This document specifies the *BIC Realtime* web service ONIX Product Information Request format and the “payload” for the corresponding ONIX Product Information Response format.

Three alternative formats are specified for Requests:

- an HTTPS query format for use with implementations that use the basic HTTPS protocol¹ and GET method – sometimes referred to as the REST approach
- an XML format for use with both implementations that use either SOAP or the basic HTTPS protocol and POST method.
- a JSON format for use with implementations that use the basic HTTPS protocol and POST method.

The Response payload format options (payload in XML or JSON) will both apply to basic HTTPS exchanges using the POST method, but XML is the only Response payload format supported for HTTPS requests using the GET method. A Request using the HTTPS GET method may be more limited than a Request using the HTTPS POST method, so the Response payload may use only a correspondingly limited subset of the content defined here. SOAP only supports XML as a Request or Response payload format.

The complete specification of these two closely-related *BIC Realtime* web services includes two machine-readable resources that are to be used by implementers in conjunction with this document:

- a WSDL Definition for the SOAP protocol versions of the *BIC Realtime* web services
- an XML Schema for Request and Response payloads in XML format.

It is strongly recommended that SOAP client implementations of these *BIC Realtime* web services be constructed using the BIC WSDL Definitions as a starting point, as this will promote interoperability between SOAP client and server implementations. In some development environments it may be easier to implement a SOAP server without using the BIC WSDL Definitions, but in this case care must be taken to ensure that the WSDL Definitions that describe the actual implementation is functionally equivalent to the BIC WSDL Definitions.

¹ Throughout the term ‘HTTPS protocol’ is to be interpreted as including secure internet protocols that are implemented either at the application layer (e.g. HTTPS) or are implemented at the transport layer (e.g. SSL/TLS).

Business requirements

The formats have been designed to support the implementation of *BIC Realtime* web services that accept product information requests and respond by supplying product information in ONIX format. Given the growing use of ONIX in the UK for the supply of book product information, a business case can be made for such a *BIC Realtime* web service at a number of points in the supply chain. Service providers are likely to include wholesalers, distributors and others supplying ONIX product information to the book trade.

Scope of the proposed formats

The requirements for sending and responding to a Product Information Request can be defined relatively simply. There are only a few options that a service provider would need to decide whether or not to offer.

For the Product Information Request application, the request format is fully specified in this document, in separate versions for HTTPS GET and POST methods.

As well as including a header identifying the source and date-stamping the response, the response format allows error conditions to be reported, and provides an XML “wrapper” within which an ONIX product information record for each successful product request can be sent. The response format also enables the inclusion of a link to each requested ONIX product information record, as an alternative to embedding the record directly in the response, and this is the only supported option when the response is in JSON format².

Implementations may determine the extent of the information they supply about each product, but ONIX product information records returned must be valid and they should, if possible, meet BIC product data accreditation standards. In particular, ONIX product information records must be valid XML in accordance with the specified release of the ONIX DTD or schema.

A service provider is not obliged to implement the associated *BIC Realtime* ONIX Product Information Acknowledgement web service, but is highly likely to find it beneficial to do so.

Correction and additions for Version 2.0 made January 2020

- | | |
|---------|---|
| General | Version number updated from ‘1.1’ to ‘2.0’ in specification tables and examples.
Support for JSON implementation added to specification tables and examples.
Text corrected in various places to make it clear that the SOAP protocol only supports XML payloads and not JSON payloads.
Deprecated elements and code values removed. |
| Page 4 | HTTPS Request lines 1 and 2: Parameters ClientID and ClientPassword made non-mandatory. It is recommended that HTTPS header-based authentication be used where possible.
HTTPS Request line 12: Parameter DescriptionLanguageCode added to enable preferred language of descriptions to be specified. The value must be a three-letter language code from ONIX code list 74. |
| Page 5 | Request header lines 1 and 2: Elements ClientID and ClientPassword made non-mandatory. It is recommended that HTTPS header-based authentication be used where possible.
Request header line 8: Element DescriptionLanguageCode added to enable preferred language of descriptions to be specified. The value must be a three-letter language code from ONIX code list 74. |

² The expression of ONIX product information records in JSON format is not currently supported by EDItEUR and is therefore not currently supported by this API. For further information on JSON and ONIX see the ONIX for Books Implementation and Best Practice Guide Release 3.0 revision 7 or later.

Page 8 Response header line 8: Element DescriptionLanguageCode added to enable language of description to be specified. The value must be a three-letter language code from ONIX code list 74.

Page 9 Response header line 10: Element ONIXRecordAttachmentType added to support the option in the response of including URI links to ONIX records instead of embedding ONIX records directly. URI links are the only option that may be used if the response payload is in JSON format.

Response detail line 3: Element DescriptionLanguageCode added to enable language of description to be specified. The value must be a three-letter language code from ONIX code list 74.

Response detail line 4: Description of the use of element Record clarified as a result of adding the element RecordURI and support for payloads in JSON format.

Response detail line 5: Element RecordURI added to enable the response to contain links to ONIX records instead of embedding the ONIX records.

ONIX PRODUCT INFORMATION REQUEST

Requests using the basic HTTPS protocol and the GET method

Requests using the basic HTTPS protocol and the GET method should include in the URL a query string containing parameters as specified below.

³	Parameter description	M ⁴	Name	
1	A unique identifier for the sender of the Request. An alphanumeric string not containing spaces or punctuation ⁵ .	D	ClientID	
2	A password to further authenticate the sender of the request ⁵ .	D	ClientPassword	
3	A code value from a BIC-controlled codelist for the scheme used for the account identifier (see ONIX codelist 44). Mandatory if including an account identifier. Permitted values are: 01 Proprietary 06 EAN-UCC GLN 07 SAN 11 PubEasy PIN	D	AccountIDType	
4	Account identifier for this request, using the specified scheme	D	AccountIDValue	
5	Identification number of this request	D	RequestNumber	
6	A date/time reference for this request	D	IssueDateTime	
7	Preferred record type. Permitted values are: 00 Determined by prior agreement (default) 01 Complete ONIX record 02 Block-level update (ONIX 3.0 only)	D	RecordType	
8	Format in which the requestor would prefer product information records to be expressed in the response. Mandatory in every request. 01 ONIX Release 2.1 - reference tags 02 ONIX Release 2.1 - short tags 03 ONIX Release 3.0 - reference tags 04 ONIX Release 3.0 - short tags	M	ONIXRecordFormat	
9	EAN-13 product number (mandatory unless trading partners have agreed to use an alternative product identifier)	D	EAN13	
10	A code value from a BIC-controlled codelist for the type of an alternative identifier of the product (see ONIX codelist 5 - code value '02' excluded).	D	ProductIDType	
11	An alternative product identifier of the specified type. Only one alternative type of identifier may be carried in a Request using the HTTPS protocol.	D	ProductIDValue	
12	Language in which the requester would prefer free-text descriptions to be expressed – use ONIX code list 74.	D	DescriptionLanguageCode	

Example of a Request using the HTTPS protocol and GET method

<https://www.booksupplier.co.uk/ONIXProductInformationService?&ONIXRecordFormat=01&ProductIDType=03&ProductIDValue=9781234567890>

³ The order of parameters in an HTTPS GET Request is insignificant.

⁴ In the column headed "M", "M" means mandatory, and "D" means dependent.

⁵ It is recommended that HTTPS header-based authentication be used where possible.

Requests using the SOAP or basic HTTPS protocol and the POST method

Requests using the HTTPS POST method should include an XML or JSON request document as the body of a request message. Requests using the SOAP protocol must include an XML document.

Request document name and version

	ONIX Product information request Version 2.0		<ONIXProductInformationRequest version="2.0"> { "ONIXProductInformationRequest": { "version": ...	
--	---	--	--	--

Header

	Request header	M	Header.	6
1	A unique identifier for the sender of the request. An alphanumeric string not containing spaces or punctuation	D	ClientID	
2	A password to further authenticate the sender of the request	D	ClientPassword	
3	Account identifier for this request A code value from a BIC-controlled codelist for the scheme used for the account identifier (see ONIX codelist 44). Mandatory if including an account identifier. Permitted values are: 01 Proprietary 06 EAN-UCC GLN 07 SAN 11 PubEasy PIN Account identifier for this request, using the specified scheme	D M M	AccountIdentifier. AccountIDType IDValue	
4	Identification number / string of this request	D	RequestNumber	
5	Document date/time: the date/time when the request was generated. Permitted formats are: YYYYMMDD YYYYMMDDTHHMM YYYYMMDDTHHMMZ (universal time) YYYYMMDDTHHMM±HHMM (time zone) where "T" represents itself, ie letter T	D	IssueDateTime	
6	Preferred record type. Permitted values are: 00 Determined by prior agreement (default) 01 Complete ONIX record 02 Block update	D	RecordType	
7	Format in which the requestor would prefer product information records to be expressed in the response. Mandatory in every request. 01 ONIX Release 2.1 - reference tags 02 ONIX Release 2.1 - short tags 03 ONIX Release 3.0 - reference tags 04 ONIX Release 3.0 - short tags	M	ONIXRecordFormat	
8	Language in which the requester would prefer free-text descriptions to be expressed – use ONIX code list 74.	D	DescriptionLanguageCode	

⁶ An 'R' in the right-most column means that the element is repeatable. If implementing this API using the JSON format option, all repeatable elements must be represented by JSON array objects.

Request detail

	Product	M	Product.	R
1	EAN-13 product number (mandatory unless trading partners have agreed to use an alternative product identifier)	D	EAN13	
2	Alternative product identifier	D	ProductIdentifier.	R
	Product ID type - see ONIX codelist 5, code value '02' excluded	M	ProductIDType	
	ID type name, only if ID type = proprietary	D	IDTypeName	
	Product number	M	Identifier	

Example of a Request XML payload using either the SOAP or the HTTPS protocol and the POST method:

```
<ONIXProductInformationRequest version="2.0"
xmlns="http://www.bic.org.uk/webservices/onixProductInfo">
  <Header>
    <AccountIdentifier>
      <AccountIDType>01</AccountIDType>
      <IDValue>12345</IDValue>
    </AccountIdentifier>
    <RequestNumber>001</RequestNumber>
    <IssueDateTime>20190418T152500</IssueDateTime>
    <ONIXRecordFormat>01</ONIXRecordFormat>
  </Header>
  <Product>
    <ProductIdentifier>
      <ProductIDType>03</ProductIDType>
      <IDValue>9781234567890</IDValue>
    </ProductIdentifier>
  </Product>
</ONIXProductInformationRequest>
```

Example of a Request JSON payload using the HTTPS protocol and the POST method:

```
{
  "ONIXProductInformationRequest": {
    "version": "2.0",
    "xmlns": "http://www.bic.org.uk/webservices/onixProductInfo",
    "Header": {
      "AccountIdentifier": {
        "AccountIDType": "01",
        "IDValue": "12345"
      },
      "RequestNumber": "001",
      "IssueDateTime": "20190418T152500",
      "ONIXRecordFormat": "01"
    },
    "Product": [
      {
        "ProductIdentifier": [
          {
            "ProductIDType": "03",
            "IDValue": "9781234567890"
          }
        ]
      }
    ]
  }
}
```

ONIX PRODUCT INFORMATION RESPONSE

The Response will use the protocol corresponding to the Request. If the Request uses the basic HTTPS protocol, the Response will be an XML or JSON document as specified below attached to a normal HTTPS header. If the Request uses the SOAP protocol, the Response will contain a SOAP response message whose body will contain the XML document specified below.

Response document name and version

ONIX Product information response Version 2.0	<ONIXProductInformationResponse version="2.0"> { "ONIXProductInformationResponse": { "version": ...
--	---

Header

	Payload header	M	Header.	
1	Document date/time: the date/time when the response was generated. Permitted formats are: YYYYMMDD YYYYMMDDTHHMM YYYYMMDDTHHMMZ (universal time) YYYYMMDDTHHMM±HHMM (time zone) where "T" represents itself, i.e. letter T	M	IssueDateTime	
2	Sender (<i>BIC Realtime</i> web service host) Sender ID type - see ONIX codelist 92 ID type name, only if ID type = proprietary Identifier	M M D M	SenderIdentifier. SenderIDType IDTypeName IDValue	
3	Identification number / string of this response	D	ResponseNumber	
4	Account identifier, required if included in the request A code value from a BIC-controlled codelist for the scheme used for the account identifier (see ONIX codelist 44). Must be specified if an account identifier is specified. Permitted schemes are: 01 Proprietary 06 EAN-UCC GLN 07 SAN 11 PubEasy PIN Account identifier for this request, using the specified scheme	D M M	AccountIdentifier. AccountIDType IDValue	
5	References: request number of request must be quoted if included in the request; request date or date and time must be quoted in this composite if both number and date/time are included in the request. Reference type 01 Number or date/time of associated ONIX product information request Reference number / string Reference date or date and time	D M D D	ReferenceCoded ReferenceTypeCode ReferenceNumber ReferenceDateTime	R
6	Reference date or date and time: must be quoted separately if and only if included in the request <i>without</i> a request number. See header line 5 for permitted formats.	D	ReferenceDateTime	
7	Default currency of prices given in the response	D	CurrencyCode	

Response Header (continued)

	Payload header	M	Header.	
8	<p>Response code, if there are exception conditions that affect the response as a whole</p> <p>Response type code. Suggested code values: <i>01</i> Service unavailable <i>02</i> Invalid ClientID or ClientPassword <i>03</i> Server unable to process request– a reason should normally be given as a free text description – see below <i>08</i> Product record format not as requested</p> <p>Free text description / reason for response. If the request used the basic HTTPS protocol, and there was an unprocessable error in the request (e.g. invalid account ID type code), the request string may be repeated here for reference and debugging purposes.</p> <p>Language of description. Mandatory if included in the request. See request header line 8.</p>	D M D D	<p>ResponseCoded.</p> <p>ResponseType</p> <p>ResponseTypeDescription</p> <p>DescriptionLanguageCode</p>	R
9	<p>Record type</p> <p>Record type code <i>00</i> See description for details <i>01</i> ONIX record without supply P&A <i>02</i> ONIX record with supply P&A <i>04</i> ONIX 3.0 block-level update without supply P&A <i>05</i> ONIX 3.0 block-level update with supply P&A</p> <p>Record type description</p>	D M D	<p>RecordType</p> <p>RecordTypeCode</p> <p>RecordTypeDescription</p>	
10	<p>Format in which product information records are expressed in the response detail. Mandatory in every request that contains response detail elements.</p> <p><i>01</i> ONIX Release 2.1 - reference tags <i>02</i> ONIX Release 2.1 - short tags <i>03</i> ONIX Release 3.0 - reference tags <i>04</i> ONIX Release 3.0 - short tags</p>	D	ONIXRecordFormat	
11	<p>ONIX record attachment type. If not included, the default is that ONIX records, if any, are embedded in the response.</p> <p><i>01</i> ONIX records embedded in response. Only supported if the Response payload as a whole is in XML format <i>02</i> URI links to ONIX records included in response</p>	D	ONIXRecordAttachmentType	

Response detail

	ONIX Product information record: mandatory unless the header reports an exception condition that prevents any response	D	ONIXProductInformationRecord.	R
1	EAN-13 product number as specified in the request detail, if any (mandatory unless trading partners have agreed to use an alternative product identifier). Mandatory if included in the request detail to which this response relates.	D	EAN13	
2	<p>Alternative product identifier as specified in the request detail. Mandatory if included in the request detail to which this response relates.</p> <p>Product ID type - see ONIX codelist 5, code value '02' excluded. Use code value '01' (proprietary) if invalid in the request.</p> <p>ID type name, only if ID type = proprietary</p> <p>Product number</p>	D M D M	<p>ProductIdentifier.</p> <p>ProductIDType</p> <p>IDTypeName</p> <p>Identifier</p>	R

Response detail (continued)

	ONIX Product information record	D	ONIXProductInformationRecord.
3	<p>Response code, if no information can be sent for this product. If present, no further elements may be included in this product information record.</p> <p>Response type code. Suggested code values: 06 Invalid product ID 07 No information for this product</p> <p>Free text description If the request used the basic HTTPS protocol, and there was an unprocessable error in the request (e.g. invalid product ID type code), the request string may be repeated here for reference and debugging purposes.</p> <p>Language of description. Mandatory if included in the request. See request header line 13.</p>	D M D D	ResponseCoded. ResponseType ResponseTypeDescription DescriptionLanguageCode
4	<p>The content of a record must comprise the content of an ONIX product information record that conforms to the record format specified in the header (i.e. it is valid according to the corresponding DTD or Schema). Each record must also be consistent with the specified record type. It is recommended that the content of this element should be <i>either</i> a Product element <i>or</i> an ONIXMessage element containing a single Product element, to facilitate validation and processing. This element may only be included if consistent with the attachment type if specified in the header – see header line 11. This element may not be included if the response payload is in JSON format.</p>	D	Record
5	<p>Location of the ONIX record expressed as a URI. Either this element or a Record element may be included, but not both. This element may only be included if consistent with the attachment type specified in the header – see header line 11.</p>	D	RecordURI

Example of a Response XML payload containing an embedded ONIX record:

```

<ONIXProductInformationResponse version="2.0"
xmlns="http://www.bic.org.uk/webservices/onixProductInfo">
  <Header>
    <IssueDateTime>20190424T1145</IssueDateTime>
    <SenderIdIdentifier>
      <SenderIDType>01</SenderIDType>
      <IDValue>XYZ</IDValue>
    </SenderIdIdentifier>
    <AccountIdentifier>
      <AccountIDType>01</AccountIDType>
      <IDValue>12345</IDValue>
    </AccountIdentifier>
    <ReferenceCoded>
      <ReferenceTypeCode>01</ReferenceTypeCode>
      <ReferenceNumber>001</ReferenceNumber>
      <ReferenceDateTime>20190418T152500</ReferenceDateTime>
    </ReferenceCoded>
    <RecordType>
      <RecordTypeCode>01</RecordTypeCode>
    </RecordType>
    <ONIXRecordFormat>01</ONIXRecordFormat>
  </Header>

```

```

<ONIXProductInformationRecord>
  <ProductIdentifier>
    <ProductIDType>03</ProductIDType>
    <IDValue>9781234567890</IDValue>
  </ProductIdentifier>
  <Record>
    <Product>
      <RecordReference>00101</RecordReference>
      <NotificationType>04</NotificationType>
      <ProductIdentifier>
        <ProductIDType>03</ProductIDType>
        <IDValue>9781234567890</IDValue>
      </ProductIdentifier>
      <ProductForm>BA</ProductForm>
      <Title>
        <TitleType>01</TitleType>
        <TitleText>Title of book</TitleText>
      </Title>
      <Contributor>
        <ContributorRole>A01</ContributorRole>
        <PersonNameInverted>Author, A.N.</PersonNameInverted>
      </Contributor>
      <ImprintName>Imprint</ImprintName>
    </Product>
  </Record>
</ONIXProductInformationRecord>
</ONIXProductInformationResponse>

```

Example of a Response JSON payload containing a URI link to an ONIX record:

```

{
  "ONIXProductInformationResponse": {
    "version": "2.0",
    "xmlns": "http://www.bic.org.uk/webservices/onixProductInfo",
    "Header": {
      "IssueDateTime": "20190424T1145",
      "SenderIdentifier": {
        "SenderIDType": "01",
        "IDValue": "XYZ"
      },
      "AccountIdentifier": {
        "AccountIDType": "01",
        "IDValue": "12345"
      },
      "ReferenceCoded": [
        {
          "ReferenceTypeCode": "01",
          "ReferenceNumber": "001",
          "ReferenceDateTime": "20190418T152500"
        }
      ],
      "RecordType": {"RecordTypeCode": "01"},
      "ONIXRecordFormat": "01"
    },
    "ONIXProductInformationRecord": [
      {
        "ProductIdentifier": [
          {
            "ProductIDType": "03",
            "IDValue": "9781234567890"
          }
        ],
        "Record": {
          "RecordURI": "https://onixserver.com/9781234567890.xml"
        }
      }
    ]
  }
}

```