

Book Industry Communication



Library Interoperability Standards LCF version 1.0 Web Services Implementation Version 1.0, 10 January 2014

This document defines a binding of the LCF data communication framework to the HTTP¹ protocol suitable for implementation of LCF in web services, following REST (Representational State Transfer²) design principles,

The use of this document is subject to license terms and conditions that can be found at <http://www.bic.org.uk/files/bicstandardslicence.pdf>.

This document is subject to revision from time to time. The latest versions of this document, the LCF data framework specification, code lists and other resources supporting this and other specific implementations of the LCF standard may be found at <http://www.bic.org.uk/114/LCF/>.

General principles

All RESTful web service implementations of LCF should use standard HTTP features wherever possible, rather than carry the same information in request or response payloads. See the implementation notes below for details.

All web service implementations must identify the version of LCF that is implemented for each function.

Where URIs are shown in examples, the path and query parts of the URI, as defined by IETF RFC 3986³ (i.e. the sub-string of the URI that starts with '/lcf') should be the same in all web service implementations. The authority part of the URI (i.e. the sub-string of the URL to the left of '/lcf') is implementation-specific, but should not be obfuscatory.

Implementation notes

1. Terminal application authentication

RESTful web service implementation of LCF may use any of the following methods for authentication of terminal applications, provided it is practical to do so, but method A is recommended as the most RESTful approach:

- A HTTP challenge-response authentication using status code 401
- B IP address authentication (frequently impractical, for example if IP assignment is dynamic)
- C Public Key Infrastructure (PKI) authentication.

The LCF elements Q00D04 and Q00C05 should not be used in REST implementations.

¹ The Hypertext Transfer Protocol (HTTP), the basic communication protocol of the World Wide Web, is specified by IETF RFC 2616. See <http://www.ietf.org/rfc/rfc2616.txt>.

² The term Representational State Transfer (REST) was originally introduced by Roy Fielding in a doctoral dissertation in 2000, its key principles being scalability and generality in the design of web-based services. See http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.

³ See <http://www.ietf.org/rfc/rfc3986.txt>.

2. User authentication, access rights and privileges

In addition to terminal application authentication, an LMS will frequently require that the user (patron or library staff) be authenticated. The elements Q00C01 and Q00C02 should be used for this purpose. These should be included as (additional) query parameters in the request. It is recommended that HTTPS be used for all requests that contain query parameters for user authentication.,

3. Time-stamped requests and responses

Date and time stamps should be carried as HTTP parameters and the LCF elements Q00D08 and R00D04 should not be used in REST implementations.

4. Exception conditions in RESTful web service responses

In a RESTful web service implementation exception condition responses should either be carried by an HTTP response status code or, if there is no equivalent HTTP status code or if several exception conditions apply, by an HTTP response status code 207 (multi-status) with the exception conditions specified in an XML payload that conforms to the LCF Exception Conditions XML schema.

5. Encoding rules in URI query parts

The URI syntax rules don't allow certain characters in query parts, including the space character. Although these rules allow a space character to be represented by a '+' sign, it is recommended that all non-allowed characters should always be encoded using percent encoding, i.e. '%' followed by hexadecimal digits representing the character's Unicode character number.

Core functions

01 Retrieve entity item information

The request is formulated using the HTTP GET method.

Format for request URI

	<i>Element ID</i>	<i>URI Path segment(s)</i>	<i>URI Query parameter</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
1		/lcf		1		LCF initial segment
2		/1.0		1		LCF version number
3	Q01D01	/{entity-type}		1	Code	The alpha code value is used from code list ENT
4	Q01D02	/{id-value}		1	String	
5	Q00D01.2		user-id	0-1	String	Included if user authentication required in addition to terminal authentication
6	Q00D02.2		user-pwd	0-1	String	

NOTE – LCF element Q01D03 is not implemented in this binding.

Example of a Request

```
GET http://192.168.0.99:80/lcf/1.0/manifestations/1234567890
```

XML payload format for response message

If the request is successful, the HTTP response will contain an XML payload that conforms to the LCF information entity XML schema for the specified entity type.

If the request is unsuccessful, the HTTP response will include an appropriate status code, which may be 207 (Multi-status), in which case the response must contain an XML payload that conforms to the LCF exception conditions XML schema.

02 Retrieve entity instance list

The request is formulated using the HTTP GET method.

Format for request URI

	Element ID	URI Path segment(s)	URI Query parameter	Card.	Data type	Notes
1		/lcf		1		LCF initial segment
2		/1.0		1		LCF version number
3		/{key-entity-type}		0-1	Code	Key entity type, when retrieving a list of entities relating to a specific key entity, e.g. a list of items relating to a specific manifestation, or a list of charges relating to a specific patron. If included in the request, the identifier of the key entity must also be included. The alpha code value is used from code list ENT
4		/{key-entity-id-value}		0-1	String	
5	Q02D01	/{entity-type}		1	Code	The alpha code value is used from code list ENT
6	Q00D01.2		user-id	0-1	String	Included if user authentication required in addition to terminal authentication
7	Q00D02.2		user-pwd	0-1	String	
8	Q02D02.1		{property-ref}	0-n	Variable	Each instance must be a selection criterion identifier. The parameter value in each case corresponds to Q02D02.2.
9	Q02D04		os:count	0-1	Integer	Implements the OpenSearch 1.1 'count' parameter
10	Q02D05		os:startIndex	0-1	Integer	Implements the OpenSearch 1.1 'startIndex' parameter

NOTE – LCF element Q02D03 is not implemented in this binding.

Examples of a Request

GET <http://192.168.0.99:80/lcf/1.0/manifestations>

GET <http://192.168.0.99:80/lcf/1.0/manifestations/1234567890/items>

GET http://192.168.0.99:80/lcf/1.0/manifestations/1234567890/items?os:count=10
os:startIndex=0

XML payload format for response message

If the request is successful, the HTTP response will contain an XML payload that conforms to the following XML schema.

	Element ID	XML structure	Card.	Data type	Notes
1		lcf-entity-list-response xmlns="http://ns.bic.org/lcf/1.0" xmlns:os="http://a9.com/-/spec/opensource/1.1/" version="1.0"	1		Top-level message element with namespace declarations and mandatory 'version' attribute
2	R02D01	entity-type	1	Code	ENT
3	R02C02	selection-criterion	0-n		If the request contains a key entity reference, a selection-criterion should contain the entity type and identifier of the key entity.
4	R02D02.1	property-ref	1	String	Reference to an instance of the selection criterion entity (E11).
5	R02D02.2	value	1	String	
6	R02D03	os:totalResults	0-1	Integer	
7	R02D04	os:itemsPerPage	0-1	Integer	
8	R02D05	os:startIndex	0-1	Integer	
9	R02C06	entity href="{instance-uri}"	1-n	AnyURI	The 'href' attribute on the element 'entity' contains the URI for retrieving the instance of the specified entity type

NOTE – LCF element R02C07 is not implemented.

Example of a Response XML payload

```
<lcf-entity-list-response xmlns="http://ns.bic.org/lcf/1.0" version="1.0">
  <entity-type>01</entity-type>
  <entity href="http://192.168.0.99:80/lcf/1.0/items/1234567890"/>
</lcf-entity-list-response>
```

If the request is unsuccessful, the HTTP response will include an appropriate status code, which may be 207 (Multi-status), in which case the response must contain an XML payload that conforms to the LCF exception conditions XML schema.

03 Create entity item

The request is formulated using the HTTP POST method. The payload is an XML document that all conform to one of the LCF information entity XML schemas.

Format for request URI

	<i>Element ID</i>	<i>URI Path segment(s)</i>	<i>URI Query parameter</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
1		/lcf		1		LCF initial segment
2		/1.0		1		LCF version number
3		/{key-entity-type}		0-1	Code	Key entity type, when creating an entity relating to a specific key entity, e.g. an item that is a copy of a specific manifestation. If included in the request, the identifier of the key entity must also be included. The alpha code value is used from code list ENT
4		/{key-id-value}		0-1	String	
5	Q03D01	/{entity-type}		1		The alpha code value is used from code list ENT
6	Q00D01.2		user-id	0-1	String	Included if user authentication required in addition to terminal authentication
7	Q00D02.2		user-pwd	0-1	String	

Examples of a Request

```
POST http://192.168.0.99:80/lcf/1.0/manifestation
```

```
POST http://192.168.0.99:80/lcf/1.0/manifestations/1234567890/item
```

XML payload format for response message

If the request is successful, the HTTP response should include status code 201 (Created), in which case the HTTP header must contain a Location field containing the URL for retrieving the created entity (see function 01 above). The response may additionally contain an XML payload that conforms to the LCF information entity XML schema for the specified entity type.

If the request is unsuccessful, the HTTP response will include an appropriate status code, which may be 207 (Multi-status), in which case the response must contain an XML payload that conforms to the LCF exception conditions XML schema.

04 Modify entity item

The request is formulated using the HTTP PUT method. The payload is an XML document that conforms to one of the LCF information entity XML schemas.

Format for request URI

	<i>Element ID</i>	<i>URI Path segment(s)</i>	<i>URI Query parameter</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
1		/lcf		1		LCF initial segment
2		/1.0		1		LCF version number
3	Q04D01	/{entity-type}		1		The alpha code value is used from code list ENT
4	Q04D02	/{item-ref}		1		
5	Q00D01.2		user-id	0-1	String	Included if user authentication required in addition to terminal authentication
6	Q00D02.2		user-pwd	0-1	String	

NOTE – This function replaces the entity item identified in the request with the content of the payload. LCF element Q04D03 is therefore implicitly included with value '01'.

Example of a Request

```
PUT http://192.168.0.99:80/lcf/1.0/manifestation/1234567890
```

XML payload format for response message

If the request is successful, the HTTP response should include status code 200 (OK) and may additionally contain an XML payload that conforms to the LCF information entity XML schema for the specified entity type.

05 Delete entity item

The request is formulated using the HTTP DELETE method.

Format for request URI

	<i>Element ID</i>	<i>URI Path segment(s)</i>	<i>URI Query parameter</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
1		/lcf		1		LCF initial segment
2		/1.0		1		LCF version number
3	Q05D01	/{entity-type}		1		The alpha code value is used from code list ENT
4	Q05D02	/{item-id}		1		
5	Q00D01.2		user-id	0-1	String	Included if user authentication required in addition to terminal authentication
6	Q00D02.2		user-pwd	0-1	String	

Example of a Request

```
DELETE http://192.168.0.99:80/lcf/1.0/manifestation/1234567890
```

XML payload format for response message

If the request is successful, the HTTP response must include the status code 204 (No content) and no payload.

Circulation functions

11 Check-out / renewal

The difference between a check-out and renewal is that in the latter case an existing, active loan of the same item to the same patron must exist. It should not be necessary for the terminal application to know whether an item is already on loan to the patron in question, because the LMS will be able to determine whether this is the case or not. A single function will therefore normally suffice.

Request

The request is formulated using the HTTP POST method.

Format for request URI

	<i>Element ID</i>	<i>URI Path segment(s)</i>	<i>URI Query parameter</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
1		/lcf		1		LCF initial segment
2		/1.0		1		LCF version number
3		/loan		1		
4	Q00D01.2		user-id	0-1	String	Included if user authentication required in addition to terminal authentication
5	Q00D02.2		user-pwd	0-1	String	
6	Q11D01		confirmation	0-1	Y	
7	Q11D07		charge-acknowledged	0-1	Y	Inclusion of this query parameter with any value other than 'n' or 'N' should be interpreted as indicating that a charge may be created for this loan.

A new check-out is performed by creating a new loan record, using LCF function 03 (see above), e.g.

```
POST http://192.168.0.99:80/lcf/1.0/loan
```

Request to confirm a new check-out, which the LMS may not normally deny, is indicated by including the 'confirmation' parameter in the request, e.g.

```
POST http://192.168.0.99:80/lcf/1.0/loan?confirmation=Y
```

If a charge is applicable, the response may report an exception unless the 'charge-acknowledged' parameter is included in the request, e.g.

```
POST http://192.168.0.99:80/lcf/1.0/loan?charge-acknowledged=Y
```

An XML document that conforms to the XML schema for a loan entity (E05) must be uploaded with the request.

In the case of a renewal, the creation of a new loan may be more readily performed by first retrieving the current loan, then using it as a basis for the new loan. However, this could be performed by the LMS, so it should not be necessary for the terminal application to carry out these additional steps.

It is assumed that the other record functions (check that copy can be loaned, check patron status, cancel reservation if any, create charge record if appropriate, update patron and copy records) are performed on the server side. If any of these functions have to be done manually by the terminal application client, a sequence of basic retrieval, modification and deletion functions may be used for this purpose.

Response

The response to a check-out or renewal may be the same response as for creating any entity, i.e. status code 201 (Created) and a Location field in the HTTP header, or it may contain an XML payload that conforms to the following schema. The advantage of including the XML payload in the response is that the terminal application will thereby be alerted by the inclusion of any of R11D03 – R11D05 in the response, rather than having to retrieve the newly-created loan in order to determine whether or not sensitive media are involved or security needs de-sensitization, or what charge has been made for the loan.

	<i>Element ID</i>	<i>XML structure</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
1		lcf-check-out-response version="1.0"	1		Top-level message element with mandatory 'version' attribute
2	R11D01	loan-ref	0-1	String	One of R11D01, R11C02 or R11D03 must be included in the response.
3	R11C02	loan	0-1		See E05
4	R11D03	media-warning	0-1	Code	MEW – Omitted if responding to a renewal
5	R11D04	security-desensitize	0-1	Code	SCD – Omitted if responding to a renewal
6	R11D05	charge-ref	0-1	String	

Example of a Response XML payload:

```
<lcf-check-out-response xmlns="http://ns.bic.org/lcf/1.0" version="1.0">
  <loan-ref>1234567890</loan-ref>
  <sensitive-media-warning>00</sensitive-media-warning>
</lcf-check-out-response>
```

Cancel check-out / renewal

In the case of a new check-out, a cancellation is simply a deletion of a loan, using LCF function 05 (see above), e.g.:

```
DELETE http://192.168.0.99:80/lcf/1.0/loan/1234567890
```

In the case of a renewal, a cancellation involves both deletion of the new loan and modification of the loan that preceded the renewal to modify its status and to remove any reference to the (now deleted) renewal loan.

This also presumes that other record modifications (reset patron and item records, delete any charge record, if any, were created) are performed server-side..

It is implementation-defined as to whether cancellation of a renewal should reset all records as they were prior to the renewal, or treat the cancellation of the renewal as being the same as a check-in.

12 Check-in

Request

The check-in function involves modification of a loan, using function 04 above, to change the status of the loan to '08' (checked in). The loan is first retrieved, then modified, e.g.

1. The URI of the current loan is found:

```
GET http://192.168.0.99:80/lcf/1.0/items/1234567890/loans?status=01
```

2. The current loan is retrieved for modification:

```
GET http://192.168.0.99:80/lcf/1.0/loans/1234567654
```

3. The retrieved loan is modified:

```
PUT http://192.168.0.99:80/lcf/1.0/loans/1234567654
```

This presumes that a number of consequential functions are performed server-side.

Response

A check-in response may be the same response as for modifying any entity, or may contain an XML message that conforms to the following schema. The advantage of including the XML payload in the response is that the terminal application will thereby be alerted by the inclusion of any of R12D04 – R12D08 in the response.

	<i>Element ID</i>	<i>XML structure</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
1		lcf-check-in-response version="1.0"	1		Top-level message element with mandatory 'version' attribute
2	R12D01	loan-ref	1	string	
3	R12D04	return-location-ref	0-1	String	
4	R12D05	media-warning	0-1	Code	MEW
5	R12D06	special-attention	0-1	Code	SPA
6	R12D07	special-attention-note	0-1	String	
7	R12D08	charge-ref	0-n	String	

Example of a Response XML payload:

```
<lcf-check-in-response xmlns="http://ns.bic.org/lcf/1.0" version="1.0">
  <loan-ref>1234567890</loan-ref>
  <return-location-ref>repair-bin</return-location-ref>
  <sensitive-media-warning>00</sensitive-media-warning>
  <special-attention>02</special-attention>
</lcf-check-in-response>
```

Cancel check-in

Cancellation of check-in involves modifying all records affected by the check-in process, to reset them as they were prior to the check-in function being performed. As a minimum, the terminal application will need to reset the status of the loan to '01' (on loan to patron), which could trigger the server/LMS to roll back changes made to other records (item, patron, charge).

13 Patron payment

Request

Making a patron payment involves creating a payment record, assuming that all consequent modifications to charge and patron records are server-side functions.

```
POST http://192.168.0.99:80/lcf/1.0/payment
```

An XML document conforming to the XML schema for payment entities must be attached to the POST request.

Response

The response is the same as for creating any entity – see function 03 above.

14 Block patron account

Request

Blocking a patron account involves a change to the status of a patron and therefore a modification to a specific patron record. No other functions are involved. Normally the patron record would need to be retrieved, then modified, i.e.:

```
GET http://192.168.0.99:80/lcf/1.0/patron/1234567890
```

```
PUT http://192.168.0.99:80/lcf/1.0/patron/1234567890
```

The payload of the PUT request is an XML document containing the modified patron record.

Response

The response is the same as for modifying any entity – see function 04 above.

15 Un-block patron account

Request

Un-blocking a patron account, as with blocking, involves a change in the status of a patron and therefore a modification to a specific patron record, having first retrieved the record. No other functions are involved.

```
GET http://192.168.0.99:80/lcf/1.0/patron/1234567890
```

```
PUT http://192.168.0.99:80/lcf/1.0/patron/1234567890
```

The payload of the PUT request is an XML document containing the modified patron record.

Response

The response is the same as for modifying any entity – see function 04 above.

16 Reserve manifestation / item

Request

The request is formulated using the HTTP POST method.

Format for request URI

	<i>Element ID</i>	<i>URI Path segment(s)</i>	<i>URI Query parameter</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
1		/lcf		1		LCF initial segment
2		/1.0		1		LCF version number
3		/reservation		1		
4	Q00D01.2		user-id	0-1	String	Included if user authentication required in addition to terminal authentication
5	Q00D02.2		user-pwd	0-1	String	
6	Q11D01		confirmation	0-1	Y	
7	Q16D10		charge-acknowledged	0-1	Y	Inclusion of this query parameter with any value other than 'n' or 'N' should be interpreted as indicating that a charge may be created for this loan.

A reservation is performed by creating a new reservation record, using LCF function 03 (see above), e.g.

```
POST http://192.168.0.99:80/lcf/1.0/reservation
```

Request to confirm a reservation, which the LMS may not normally deny, is indicated by including the 'confirmation' parameter in the request, e.g.

```
POST http://192.168.0.99:80/lcf/1.0/reservation?confirmation=Y
```

If a charge is applicable, the response may report an exception unless the 'charge-acknowledged' parameter is included in the request, e.g.

```
POST http://192.168.0.99:80/lcf/1.0/reservation?charge-acknowledged=Y
```

An XML document that conforms to the XML schema for a reservation entity (E06) must be uploaded with the request.

Response

A reservation response may be the same response as for creating any entity, i.e. status code 201 (Created) and a Location field in the HTTP header, or it may contain an XML message that conforms to the following schema. The advantage of including the XML payload in the response is that the terminal application will thereby be alerted by the inclusion of R16D03 in the response, rather than having to retrieve the newly-created reservation in order to determine what charge has been made for the reservation.

	<i>Element ID</i>	<i>XML structure</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
--	-------------------	----------------------	--------------	------------------	--------------

	<i>Element ID</i>	<i>XML structure</i>	<i>Card.</i>	<i>Data type</i>	<i>Notes</i>
1		lcf-reservation-response version="1.0"	1		Top-level message element with mandatory 'version' attribute
2	R16D01	reservation-ref	0-1	String	Either R16D01 or R16D02 must be included in the response.
3	R16C02	reservation	0-1		See E06
7	R16D03	charge-ref	0-1	String	

Example of a Response XML payload:

```
<lcf-reservation-response xmlns="http://ns.bic.org/lcf/1.0" version="1.0">
  <reservation-ref>R1234</reservation-ref>
  <charge-ref>C12345</charge-ref>
</lcf-reservation-response>
```

Stock management functions

21 Retrieve location list

This function is the same as core function 02, applied to the retrieval of a list of location entities, for example:

```
GET http://192.168.0.99:80/lcf/1.0/locations?{selection-criteria}
```

22 Retrieve title classification scheme list

This function is the same as core function 02, applied to the retrieval of a list of title classification scheme entities, for example:

```
GET http://192.168.0.99:80/lcf/1.0/class-schemes
```

23 Retrieve title classification list

This function is the same as core function 02, applied to the retrieval of a list of title classification code entities, for example:

```
GET http://192.168.0.99:80/lcf/1.0/class-codes?scheme=xxxxx
```

24 Retrieve (stock) item list

This function combines the core functions for retrieval of a list of manifestations and list of items. A list of titles is first retrieved matching selection criteria that relate to titles. This list, coupled with further selection criteria that relate to copies, is used to retrieve a list of copies. The two can be combined in a single request that includes both manifestation-specific and copy-specific selection criteria.

The following selects all items for a given set of selection criteria:

```
GET http://192.168.0.99:80/lcf/1.0/items?{all-selection-criteria}
```

The following selects all items that are copies of the same manifestation, for a given set of selection criteria:

```
GET http://192.168.0.99:80/lcf/1.0/manifestations/1234567890/items?{all-selection-criteria}
```

25 Retrieve selection criterion type list

This function is the same as the core function 02 for retrieving a list of selection criterion entities. A list of selection criterion types can be retrieved for a specific entity type or for all entity types, e.g.:

```
GET http://192.168.0.99:80/lcf/1.0/properties?entity-type=manifestation
```

```
GET http://192.168.0.99:80/lcf/1.0/properties?entity-type=location
```