LME Pre-Trade Transparency (PTT) XML Feed Developer Guide

version 1.4

Please respond to:
dataservices@lme.com
+ 44 800 032 77 59

Document History

Version	Date	Description of change
1.0	16th November 2017	First version for distribution
1.1	4th December 2017	Clarified access mechanism IO Quotes
1.2	7th December 2017	Clarified error text when invalid contract Added code example for accessing the feed
1.3	14th December 2017	Standardisation of language across licensing and technical specification.
1.4	3rd January 2018	Corrected URL given in section 2

TABLE OF CONTENTS

1 Introduction

This document is aimed at users of the Pre-Trade Transparency (PTT) XML Feed.

The PTT XML Feed is available daily from 00:10 (xml server time) GMT, however please note that electronic trading on LMEselect does not open until 01:00. Requests can be made throughout the day until 00:10 GMT for 15 minute delayed quote data. Users can make requests at a frequency of up to one request per second.

All subscribers will have access to data for all LME contracts and venues.

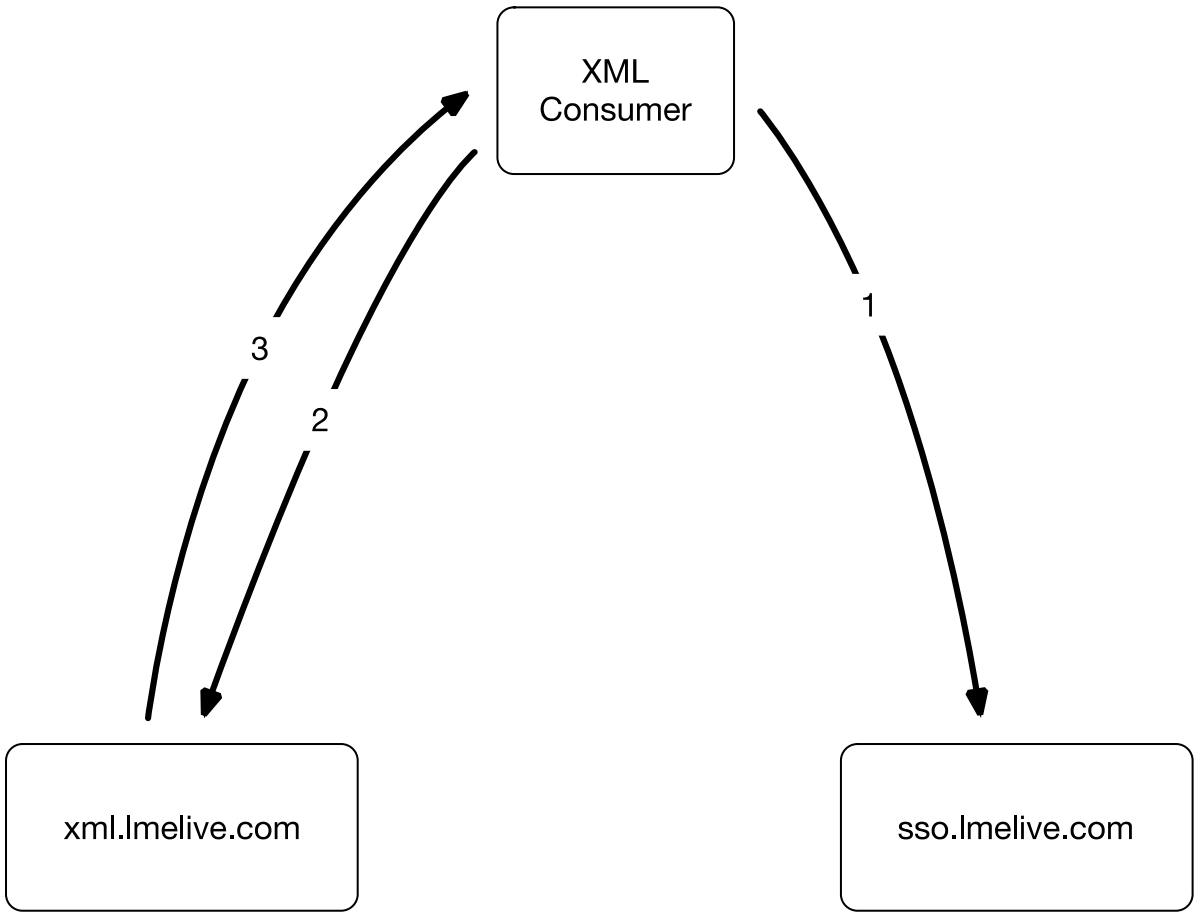
This document is divided into the following sections:

- Access – how to register for the PTT XML Feed and access it using the correct authentication mechanism;
- HTTPS Requests – the format of requests that can be made;
- HTTPS Responses – the format of responses returned by the feed.

2 Access

In order to access the PTT XML Feed, users must first register on the LME's Online Licensing Portal (OLP)

Once the user has registered, then when attempting to access the PTT XML Feed the authentication flow is as outlined in the diagram and table below (with more detail given on steps 1 and 2 in the following sections).



Step	Description
1	Application makes a call to the PingFederate token endpoint passing username and password and receives an access token in exchange. See section 2.1 below for more details.
2	Application makes a https request to the PTT XML Feed for data (e.g. https://pttxml.lmelive.com/PTTService.svc/ptt.xml?contract=AH) passing the access token received in step 1 as the Authorization header (bearer token)
3	Content returned to user

2.1 Requesting an access token

A HTTP POST must be made to retrieve an access token, the details are:

URL: https://sso.lmelive.com/as/token.oauth2

content-type: application/x-www-form-urlencoded

Body:

Attribute	Value
grant_type	Password
client_id	xmlfeeds
username	<your username>
password	<your password>

For example:

```
POST /as/token.oauth2 HTTP/1.1
Host: sso.lmelive.com:443
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
grant_type=password&username=test@example.com&password=YourPassword&cli
ent_id=xmlfeeds
```

On successful credential validation the return status will be HTTP 200 with a return body containing a JSON document including an access token and number of seconds for which the token is valid, e.g.

```
{
    "access_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IlNQMSJ9”,
    "token_type": "Bearer",
    "expires_in": 86400
}
```

The access_token will used to access the XML interface for the next 24 hours.

A code example for retrieving the access token is provided in Appendix B.

2.2 Making a request of the XML Feed using the access token

A HTTP GET must be made to make requests of the XML Feed. When making requests to the XML Feed the access token received in the previous step must be provided in each request as an authorization header, this takes the format of:

```
Authorization: Bearer <access_token>
```

For example:

```
Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6IlNQMSJ9
```

https://pttxml.lmelive.com/PTTService.svc/ptt.xml?contract=AH

A code example for retrieving the access token is provided in Appendix B.

3 HTTPS Requests

3.1 Request Format

An example of the request format is given below, with details given on the parameters in the following sections.

https://pttxml.lmelive.com/PTTService.svc/ptt.xml?contract=AH

3.2 Parameters

There is one parameter allowed on a request - contract (which is mandatory). The values that can be passed for this parameter are given below.

If an incorrect contract or no contract is provided then the user will receive an error message. See below for more details on response formats.

Contract	Description
AH	Aluminium
AA	Aluminium Alloy
CO	Cobalt
CA	Copper
NA	NASAAC
NI	Nickel
PB	Lead
SN	Tin
ZS	Zinc
AE	Premium Aluminium E Asia
AS	Premium Aluminium SE Asia
AN	Premium Aluminium US
AW	Premium Aluminium W Europe
SC	Steel Scrap
SR	Steel Rebar
AU	LME Gold
AG	LME Silver

Contract	Description
MA	Aluminium Mini
MC	Copper Mini
MZ	Zinc Mini
OA	Aluminium Monthly Average Future
OL	Aluminium Alloy Monthly Average Future
OM	NASAAC Monthly Average Future
OC	Copper Monthly Average Future
OP	Lead Monthly Average Future
ON	Nickel Monthly Average Future
OS	Tin Monthly Average Future
OZ	Zinc Monthly Average Future

4 HTTPS Responses - XML Data Content

4.1 Error Responses

If there is no data for the contract requested then the following response will be received.

```
<QueryResponse>
      <GeneratedTime>2017-06-29 14:55:33</GeneratedTime>
      <ContractRequested>AH</ContractRequested>
      <Response>No data available for contract</Response>
</QueryResponse>
```

If the contract parameter was incorrect (i.e. not one of the single contracts listed above) then the following response will be received (where the value in the <ContractRequested> element is the value originally sent in by the user):

```
<QueryResponse>
      <GeneratedTime>2017-06-29 14:55:33</GeneratedTime>
      <ContractRequested>ALL</ContractRequested>
      <Response>Invalid format for Contract – Should be like ‘CA’ or
      ‘ZS’ for example</Response>
</QueryResponse>
```

If the original request is badly formatted (e.g. contract parameter completely omitted), then the following response will be received:

```
<QueryResponse>
      <GeneratedTime>2017-06-29 14:55:33</GeneratedTime>
      <ContractRequested/>
      <Response>Contract is a required parameter</Response>
</QueryResponse>
```

The elements in all error responses are:

Element	Description
GeneratedTime	The time that the PTT XML Feed created this response (in local system time)
ContractRequested	The value from the contract parameter in the original request. This can be empty if the user omitted the contract.
Response	Additional information on the error

4.2 Data Responses

If there is data available then the response contains the following hierarchy of elements:

- `<QueryResponse>` – top level bounding element
  - `<GeneratedTime>`
  - `<ContractRequested>`
  - `<Instrument>` – one instrument for each distinct combination of product, contract type, prompt code/date, currency for futures and one for each distinct combination of product, contract type, expiry, strike, call/put indicator and currency for options
    - `<Venues>` – within each instrument this bounding element for the venues included for the instrument
      - `<Venue>` - data is presented for each Venue for that instrument.
      - `<DepthLevels>` – within each venue this bounding element demarcates the different DepthLevel elememts
        - `<DepthLevel>` – data for a single depth level for a single instrument on a single venue. Both bid and ask data is given in the DepthLevel element.

An example is shown below. Note only two instrument are shown in this example (one future, one option), but there would be multiple Instrument records. Similarly only two depth levels are shown, but there could be up to five for the LMEselect venue (EL) and one each for the Ring (RK) and Interoffice (IO) venues.

```
<QueryResponse>

<GeneratedTime>2017-06-29 14:55:33</GeneratedTime>
<ContractRequested>NI</ContractRequested>

<Instrument>

    <Product>NI</Product>
    <ContractType>F</ContractType>
    <PromptCode>3M</PromptCode>
    <PromptDate>20170920</PromptDate>
    <Currency>USD</Currency>

    <Venues>
          <Venue Code=EL>
              <DepthLevels>
                <DepthLevel>
                    <Level>1</Level>
                    <Bid>1235.00</Bid>
                    <BidSize>3</BidSize>
                    <BidNumOrders>2</BidNumOrders>
                    <BidTime>20170620 15:14:22.123</BidTime>
                    <Ask>1245.00</Ask>
                    <AskSize>2</AskSize>
                    <AskNumOrders>1</AskNumOrders>
                    <AskTime>20170620 15:13:22.456</AskTime>
                </DepthLevel>

                <DepthLevel>
                    <Level>2</Level>
                    <Bid>1245.00</Bid>
                    <BidSize>5</BidSize>
                    <BidNumOrders>3</BidNumOrders>
                    <BidTime>20170620 15:13:22.123</BidTime>
                    <Ask>1255.00</Ask>
                    <AskSize>8</AskSize>
                    <AskNumOrders>2</AskNumOrders>
                    <AskTime>20170620 15:12:22.456</AskTime>
                </DepthLevel>
              </DepthLevels>
          </Venue>

          <Venue Code=RK>
              <DepthLevels>
                <DepthLevel>
                    <Level>1</Level>
                    <Bid>1235.00</Bid>
                    <BidSize>3</BidSize>
                    <BidNumOrders>2</BidNumOrders>
                    <BidTime>20170620 15:14:22.123</BidTime>
                    <Ask>1245.00</Ask>
                    <AskSize>2</AskSize>
                    <AskNumOrders>1</AskNumOrders>
                    <AskTime>20170620 15:13:22.456</AskTime>
                </DepthLevel>
              </DepthLevels>
          </Venue>
```

```
          <Venue Code=IO>
              <DepthLevels>
                <DepthLevel>
                    <Level>1</Level>
                    <Bid>1235.00</Bid>
                    <BidSize>3</BidSize>
                    <BidNumOrders>2</BidNumOrders>
                    <BidTime>20170620 15:14:22.123</BidTime>
                    <Ask>1245.00</Ask>
                    <AskSize>2</AskSize>
                    <AskNumOrders>1</AskNumOrders>
                    <AskTime>20170620 15:13:22.456</AskTime>
                </DepthLevel>
              </DepthLevels>
          </Venue>

      </Venues>

</Instrument>

<Instrument>

    <Product>NI</Product>
    <ContractType>T</ContractType>
    <PutOrCallIndicator>C</CallPut>
    <StrikePrice>1000</StrikePrice>
    <Expiry>20170920</Expiry>
    <Currency>USD</Currency>

    <Venues>
          <Venue Code=EL>
              <DepthLevels>
                <DepthLevel>
                    <Level>1</Level>
                    <Bid>1235.00</Bid>
                    <BidSize>3</BidSize>
                    <BidNumOrders>2</BidNumOrders>
                    <BidTime>20170620 15:14:22.123</BidTime>
                    <Ask>1245.00</Ask>
                    <AskSize>2</AskSize>
                    <AskNumOrders>1</AskNumOrders>
                    <AskTime>20170620 15:13:22.456</AskTime>
                </DepthLevel>
                <DepthLevel>
                    <Level>2</Level>
                    <Bid>1245.00</Bid>
                    <BidSize>5</BidSize>
                    <BidNumOrders>3</BidNumOrders>
                    <BidTime>20170620 15:13:22.123</BidTime>
                    <Ask>1255.00</Ask>
                    <AskSize>8</AskSize>
                    <AskNumOrders>2</AskNumOrders>
                    <AskTime>20170620 15:12:22.456</AskTime>
                </DepthLevel>
              </DepthLevels>
          </Venue>
    </Venues>
</Instrument>
</QueryResponse>
```

The elements/attributes within the response are as follows.

Query Response:

Element/Attribute	Description
GeneratedTime	The time that the PTT XML Feed created this response (in local system time)

Instrument:

Element/Attribute	Description
Product	The contract value originally requested
ContractType	One of: F (Future) T (Traded Option) A (TAPO)
PromptCode	If the instrument has one then this will be provided, e.g. 3M. Else this element will be omitted.
ToPromptCode	If the instrument is a carry and has a prompt code for the far leg then this will be provided, e.g. 3M. Else this element will be omitted.
PromptDate	In format YYYYMMDD
ToPromptDate	If the instrument is a carry this will be provided, else this element will be omitted
PromptAverage	If the instrument is an average this will be provided, else this element will be omitted, e.g. 1Y18, 2Q19, 1H20
ToPromptAverage	If the instrument is a carry average this will be provided, else this element will be omitted
Expiry	If the instrument is an option this will be provided, either in format YYYYMMDD or the average format as given above (e.g. 1Y18, 2Q19, 1H20)
Currency	One of USD, GBP, JPY or EUR
StrikePrice	If the ContractType is T or A this will be provided
PutOrCallIndicator	One of: C (Call) P (Put) If the ContractType is T or A this will be provided

Venue:

Element/Attribute	Description
Code (attribute)	EL = LMEselect (Electronic) RK = Ring IO = Interoffice (Telephone)

DepthLevel:

Element/Attribute	Description
Level	Number identifying the depth level
Bid	Bid price at this level
BidSize	Volume of bid lots at this level
BidNumOrders	Number of bid orders that make up the BidSize at this level
BidTime	UTC time that this bid price was received by the source system
Ask	Ask price at this level
AskSize	Volume of ask lots at this level
AskNumOrders	Number of ask orders that make up the AskSize at this level
AskTime	UTC time that this ask price was received by the source system

5 Appendix A – Code Example

The following is a C# code example for retrieving an access token and calling the PTT XML Feed.

```
HttpClient client = new HttpClient();

// setup the content for the http post to request for an access token
// Note: the “yourusername” and “yourpassword” items should be set
// with your specific credentials. Everything else is as is here.
var values = new Dictionary<string, string>
{
        { "grant_type", "password" },
        { "client_id", "xmlfeeds" },
        { "username", "yourusername"  },
        { "password", "yourpassword"  }
};

var content = new FormUrlEncodedContent(values);

// setup the http post request for an access token
var authRequest = new HttpRequestMessage()
{
        RequestUri = new Uri("https://sso.lmelive.com/as/token.oauth2"),
        Content = content,
        Method = HttpMethod.Post
};

// send the http post request for an access token
var authResponse = await client.SendAsync(authRequest);

// read the response
var authResponseContent = await authResponse.Content.ReadAsStringAsync();

// parse the response into a json object
var json = JObject.Parse(authResponseContent);

// Get the access token from the response
var accessToken = json["access_token"].ToObject<string>();

// Set the access token in the header for the GET request
// of the feed itself, preceded by text “Bearer”
client.DefaultRequestHeaders.Authorization = new
AuthenticationHeaderValue("Bearer", accessToken);

// Send the Get request to the PTT XML Feed (in this example a
// request for the AH contract)
var response = await
client.GetAsync("https://pttxml.lmelive.com/XMLFeed.svc/ptt.xml?contract=AH");

// Read the response
var responseContent = await response.Content.ReadAsStringAsync();
```