

Specification

17 May 2016

CURRENEX

Credit Management Web Service

Please contact Currenex sales representatives and help desk personnel for more information on this documentation.

Table of Contents

1	Introduction	3
1.1	Web Service Call Sequence	3
1.2	Web Service Location	3
1.3	References	4
2	Web Service Requests	5
2.1	LoginRequest	5
2.2	LogoutRequest	5
2.3	ChangePasswordRequest	6
2.4	TestRequest	6
3	Credit Query Service Requests	7
3.1	DslInfoRequest	7
3.2	CreditExposureRequest	7
3.3	DslUpdateRequest	8
4	SOAP Message Examples	10
4.1	LoginRequest	10
4.2	LoginResponse	10
4.3	TestRequest	11
4.4	DslInfoRequest	11
4.5	DslInfoResponse	12
4.6	CreditExposureRequest	12

Currenex Client Support

US Toll Free: + 1 877 939 8723

London: + 0 800 358 5331

Singapore: + 65 6826 7476

Tokyo: + 813 4530 7555

All Other Countries:

Primary: + 1 212-340-1790

Secondary: + 1 212-685-5950

support@currenex.com

Currenex Sales

New York: + 212 340 5220

London: + 44 203 395 7930

Tokyo: + 81 3 4530 7555

Hong Kong: +852 2230 1566

Singapore: + 65 6826 7476

Australia: + 612 8429 1204

sales@currenex.com

V1.2 | 11.0 | PROD | DIR | TO

The products and services outlined herein are only offered to professional clients or eligible counterparties through State Street Bank and Trust Company (a wholly-owned subsidiary of State Street Corporation), or, within certain jurisdictions throughout Europe, through State Street Global Markets International Limited which is authorized and regulated by the Financial Services Authority. Please note, certain foreign exchange business (spot and certain forward transactions) are not regulated by the Financial Services Authority. Clients should be aware of the risks of participating in trading foreign exchange, equities, fixed income or derivative instruments or in investments in non-liquid or emerging markets. Clients should be aware that products and services outlined herein may put their capital at risk. This communication is not intended for retail clients, nor for distribution to, and may not be relied upon by, any person or entity in any jurisdiction or country where such distribution or use would be contrary to applicable law or regulation. This publication or any portion hereof may not be reprinted, sold or redistributed without the prior written consent of State Street Global Markets. This communication is being distributed in Australia by State Street Bank and Trust Company ABN 70 062 819 630, AFSL 239679 and is intended only for wholesale clients, as defined in the Corporations Act 2001.

4.7	CreditExposureResponse	13
4.8	DslUpdateRequest.....	14
4.9	DslUpdateResponse	15
4.9.1	Success	15
4.9.2	Failure.....	15
5	Appendix	16
5.1	Revision history	16

1 INTRODUCTION

This document provides information about the credit management web service.

The service provides a web-based interface for taker-side query and management of daily settlement and credit exposure configuration in the Currenex platform via the industry-standard SOAP protocol, which plugs into existing web services framework supported by Currenex. The web service is running on a server and accepting SOAP calls over HTTP containing an XML request message, and returning data in XML format. The messages and data types used in the SOAP calls are defined in the web service definition (WSDL file) and XML schema (XSD file) that should be given out together with this document.

The goal of providing a web service is to be platform independent on client side. A web service client can be written in any language and in any operation system environment, as long as it is possible to send HTTP POST requests to a remote URL.

1.1 Web Service Call Sequence

To obtain and modify credit-related information, the client application has to make the following SOAP calls:

1. **LoginRequest:** provide username/password to obtain a session ticket (unless already established session on Currenex).
2. **Application requests:** (can be repeated in any sequence as long as session ticket supplied in XML message header is valid)
 - a. **DslInfoRequest:** get daily settlement limit configuration(s) managed by party of authenticated user.
 - b. **CreditExposureRequest:** receive credit utilization information for each settlement date, per given DSL configuration.
 - c. **DslUpdateRequest:** provide new daily settlement limit amount for specific DSL configuration.

LogoutRequest: close the session and destroy the session ticket.

1.2 Web Service Location

The web service is running at the following locations:

- INT1: <https://integration-dl.currenex.com/web/service/request/CreditQueryService>
- INT2: <https://integration2-dl.currenex.com/web/service/request/CreditQueryService>
- PROD: <https://fxtrades-dl.currenex.net/web/service/request/CreditQueryService>
- LDFX: <https://dl.lofx.currenex.com/web/service/request/CreditQueryService>
- BRET: <https://bret-dl.currenex.com/web/service/request/CreditQueryService>
- DRET: <https://dret-dl.currenex.com/web/service/request/CreditQueryService>
- PRET: <https://pret-dl.currenex.com/web/service/request/CreditQueryService>
- TKFX: <https://dl.tkfx.currenex.com/web/service/request/CreditQueryService>

The products and services outlined herein are only offered to professional clients or eligible counterparties through State Street Bank and Trust Company (a wholly-owned subsidiary of State Street Corporation), or, within certain jurisdictions throughout Europe, through State Street Global Markets International Limited which is authorized and regulated by the Financial Services Authority. Please note, certain foreign exchange business (spot and certain forward transactions) are not regulated by the Financial Services Authority. Clients should be aware of the risks of participating in trading foreign exchange, equities, fixed income or derivative instruments or in investments in non-liquid or emerging markets. Clients should be aware that products and services outlined herein may put their capital at risk. This communication is not intended for retail clients, nor for distribution to, and may not be relied upon by, any person or entity in any jurisdiction or country where such distribution or use would be contrary to applicable law or regulation. This publication or any portion hereof may not be reprinted, sold or redistributed without the prior written consent of State Street Global Markets. This communication is being distributed in Australia by State Street Bank and Trust Company ABN 70 062 819 630, AFSL 239679 and is intended only for wholesale clients, as defined in the Corporations Act 2001.

1.3 References

Used references / specifications:

Definition	Protocol / Name	Reference / Specification
	Web Service	http://www.w3.org/2002/ws/
HTTP	Hypertext Transfer Protocol (HTTP/1.1)	http://www.w3.org/Protocols/
SOAP	Simple Object Access Protocol (1.2)	http://www.w3.org/TR/soap/
WSDL	Web Services Description Language (1.1)	http://www.w3.org/TR/wsdl
XML	eXtensible Markup Language	http://www.w3.org/XML/
XSD	XML Schema (1.0 / 1.1)	http://www.w3.org/XML/Schema

2 WEB SERVICE REQUESTS

This part of the document gives an overview over the available request and response messages and the data contained in the message. For each request, the contained elements, their XML Schema data types and possible restrictions are indicated. For optional data types, the default value is assumed if no value is given.

2.1 LoginRequest

LoginRequest allows to log in to the web services and to obtain a session ticket. It supports two different login modes: "GetSession" to obtain an already existing ticket of a session established through a Currenex FX/FIX client, and "CreateSession" to create a new session that can only be used for the web service. For white label side administration, CreateSession is most likely the one to be used.

The response contains the session ticket to be used in further requests. It also contains the session expire time, which can be null if the session never expires.

Request parameters:

- userID (xs:string)
- password (xs:string)
- oneTimePassword(xs:string) – One time password for 2FS implementation. Must be kept null or blank unless provided.
- loginType (xs:string) – can be "CreateSession" or "GetSession".

Response content:

- userID (xs:string)
- sessionTicket (xs:string)
- webserviceSession (xs:boolean) – true if the session was created with CreateSession, false if not
- expires (xs:dateTime) – Session expire time. If the session never expires, the element will not exist.

Error response:

- errorMessage (xs:string)
- errorType (xs:string) – can have one of the following values:
 - "BadUserOrPassword"
 - "UserNotActive" – user inactive on the system
 - "PasswordNotSet" – user has no password; send change pw request
 - "NoActiveSession" – if "GetSession" is used but no session exists
 - "OtherError" – general authentication problem or other error

2.2 LogoutRequest

LogoutRequest closes a web service session. It needs the username and session ticket set in the SOAP header.

Request parameters:

- userID (xs:string)

Response content:

- logoutResponse (xs:string) – message confirming that the user has successfully been logged out.

2.3 ChangePasswordRequest

ChangePasswordRequest allows changing the user password. This also has to be used if a PasswordExpired message is returned upon login. New password has to have a length of at least 8 characters, including at least one uppercase character and one number, and be different from the old one.

Request parameters:

- userID (xs:string)
- oneTimePassword(xs:string) – One time password for 2FS implementation. Must be kept null or blank unless provided.
- oldPassword (xs:string)
- newPassword (xs:string)

Response content:

- changePasswordResponse (xs:string) – confirmation on success

Error response:

- errorMessage (xs:string)
- errorType (xs:string) – can have one of the following values:
 - "BadUserOrPassword" (username or old password not correct)
 - "UserAuthenticationFailed" (user cannot be authenticated)
 - "NewPasswordIsOldPassword" (same password not allowed)
 - "NewPasswordEmpty" (empty password not allowed)
 - "NewPasswordInvalid" (password must have at least 8 characters, including at least 1 uppercase and 1 number)

2.4 TestRequest

The test request can be sent to test the client implementation and verify a session ticket. No action is done on the server. The user ID and session ticket must be given in the request header. The test request takes a message which is returned in the response along with some system version information.

3 CREDIT QUERY SERVICE REQUESTS

3.1 DslInfoRequest

DslInfoRequest provides a way to obtain daily settlement limit information to privileged account (owner). The information returned in response contains a collection of aggregate daily settlement limits across all possible settlement pairs in each of the credit segments the owner oversees, as well as the settlement pairs within each such segment.

Request parameters:

- `userId(xs:string)` – A valid user id
- `sessionTicket(xs:string)` – Unique token returned upon successful login
- A single empty element, named **DslInfoRequest** is expected in request body.

Response content:

Contains zero or more elements with daily settlement limit information.

DslInfo is DslInfoType[] – Complete daily settlement limit information managed by party of requesting user. DslInfoType schema definition.

- `Id(xs:string)` – optional identifier for this configuration
- `ownerId(xs:string)` – id of party managing the settlement limit configuration
- `dslRef(xs:long)` – unique key for this settlement limit configuration, used in update request (below)
- `dailySettlementLimit` of `DailySettlementLimitType` – Daily settlement limit amount and type, definition:-
 - `settlementLimitType (xs:string)` – One of:
 - NETTED
 - ACCUMULATED
 - NET_OPEN_POSITION
 - ACCUMULATED_OPEN_POSITION
 - `Amount(xs:decimal)` – Settlement limit
- `Counterparties` of `CounterPartiesType` – Counterparties representing possible settlements, `settlementPair` of `SettlementPairType`:-
 - `Entity1Id(xs:string)` – party id of first counterparty
 - `Entity2Id(xs:string)` – party id of second counterparty

3.2 CreditExposureRequest

CreditExposureRequest allows one to see utilized credit exposure broken down by settlement date, along with corresponding daily settlement limit configuration, for each of the credit segments. It should be noted that no exposure amount of any settlement date for a given DSL configuration can exceed the configured daily settlement limit amount.

Request parameters:

A single empty element, named **CreditExposureRequest** is expected in request body.

Response

Contains zero or more elements with utilized credit exposure for specific settlement days

CreditExposureInfo of CreditExposureInfoType[] – One or more CreditExposureInfoType elements

- DslInfo, DslInfoType – DslInfo for all the credit exposures returned:-
 - Id(xs:string) – optional identifier for this configuration
 - ownerId(xs:string) – id of party managing the settlement limit configuration
 - dslRef(xs:long) – unique key for this settlement limit configuration, used in update request (below)
 - dailySettlementLimit of DailySettlementLimitType – Daily settlement limit amount and type, definition:-
 - settlementLimitType(xs:string) – One of:
 - NETTED
 - ACCUMULATED
 - NET_OPEN_POSITION
 - ACCUMULATED_OPEN_POSITION
 - Amount(xs:decimal) – Settlement limit
 - Counterparties of CounterPartiesType – Counterparties representing possible settlements, settlementPair of SettlementPairType[]:-
 - Entity1Id(xs:string) – party id of first counterparty
 - Entity2Id(xs:string) – party id of second counterparty
 - DailyExposureInfo of DailyExposureInfoType[] – Credit exposure for each settlement date:-
 - date (xs:date) – Settlement date
 - exposure (xs:decimal) – Utilized exposure for settlement date
 - If settlementLimitType = NETTED or ACCUMULATED then DailyExposureInfo of DailyExposureInfoType is returned with date and exposure
 - If settlementLimitType = NET_OPEN_POSITION or ACCUMULATED_OPEN_POSITION then OpenExposureInfo is returned with exposure

3.3 DslUpdateRequest

This request allows DSL configuration owner to update daily settlement limit amount for specific DSL configuration.

Request parameters:

- dslRef (xs:long) – unique id for DSL configuration, from *DslInfoType* (see 0)
- amount (xs:decimal) – New daily settlement limit amount
- isResetUsedAmountToZero (xs:boolean) – optionally used to control utilized credit.

Response:

- dslRef (xs:long)

- result (xs:string) – One of:
 - “SUCCESS”
 - “ERROR”
- errorReason (xs:string) – If value of result is ERROR, this will contain error description.

4 SOAP MESSAGE EXAMPLES

4.1 LoginRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <loginRequest xmlns="http://currenex.com/web service/shared">
      <userID>adminuser</userID>
      <password>password</password>
      <loginType>CreateSession</loginType>
    </loginRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

4.2 LoginResponse

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <loginResponse xmlns="http://currenex.com/web service/shared">
      <userID>adminuser</userID>
      <sessionTicket>1f7245e034ac2d600ffe08b92249a390f1</sessionTicket>
      <web serviceSession>true</web serviceSession>
      <expires>2008-04-24T15:01:25.503Z</expires>
    </loginResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

The session ticket obtained in the login response is to be used in all data requests. If the session is set to automatically expire, an element “expires” is included in the response giving the time the session will expire (in GMT). If no “expires” element is given, the session won’t automatically expire.

4.3 TestRequest

The test request can be sent to test the client implementation and verify a session ticket. No action is done on the server. The SOAP header containing the session information is highlighted in the request.

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <session xmlns="http://currenex.com/web service/shared">
      <userID>adminuser</userID>
      <sessionTicket>1f7245e034ac2d600ffe08b90239a390f1</sessionTicket>
    </session>
  </soapenv:Header>
  <soapenv:Body>
    <testRequest xmlns="http://currenex.com/web service/shared">
      <testMessage>Test Message</testMessage>
    </testRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

4.4 DslInfoRequest

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns1:session
      soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
      soapenv:mustUnderstand="0"
      xmlns:ns1="http://currenex.com/web service/retail/schema">
      <ns1:userID>metropolis_c4u0</ns1:userID>
      <ns1:sessionTicket>
        9905d7a894194b5a349bf521fcef3d3bc1b
      </ns1:sessionTicket>
    </ns1:session>
  </soapenv:Header>
  <soapenv:Body>
```

```

    <DslInfoRequest
      xmlns="http://currenex.com/web service/retail/schema">

    </DslInfoRequest>

  </soapenv:Body>
</soapenv:Envelope>

```

4.5 DslInfoResponse

```

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <DslInfoResponse xmlns="http://currenex.com/web service/retail/schema">
      <ns1:DslInfo ns1:dslRef="1" id="metropolis_PB0"
        ownerId="metropolis_C4" xsi:type="ns1:DslInfoType"
        xmlns:ns1="http://currenex.com/web service/retail/schema">
        <ns1:dailySettlementLimit amount="100000000"
          settlementLimitType="NETTED"/>
        <ns1:counterParties>
          <ns1:settlementPair>
            <ns1:Entity1Id>metropolis_PB0</ns1:Entity1Id>
            <ns1:Entity2Id>metropolis_C4</ns1:Entity2Id>
          </ns1:settlementPair>
          <ns1:settlementPair>
            <ns1:Entity1Id>metropolis_B4</ns1:Entity1Id>
            <ns1:Entity2Id>metropolis_PB0</ns1:Entity2Id>
          </ns1:settlementPair>
        </ns1:counterParties>
      </ns1:DslInfo>
    </DslInfoResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

4.6 CreditExposureRequest

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <ns1:session
      soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
      soapenv:mustUnderstand="0"
      xmlns:ns1="http://currenex.com/web service/retail/schema">
      <ns1:userID>metropolis_c4u0</ns1:userID>
      <ns1:sessionTicket>
        87a7aa35ed73286733ea1e44d31d2f57fa
      </ns1:sessionTicket>
    </ns1:session>
  </soapenv:Header>
  <soapenv:Body>
    <CreditExposureRequest
      xmlns="http://currenex.com/web service/retail/schema">
    </CreditExposureRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

4.7 CreditExposureResponse

```

<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <CreditExposureResponse
      xmlns="http://currenex.com/web service/retail/schema">
      <CreditExposureInfo>
        <ns1:DslInfo ns1:dslRef="1" id="metropolis_PB0"
          ownerId="metropolis_C4"
          xmlns:ns1="http://currenex.com/web service/retail/schema">
          <ns1:dailySettlementLimit amount="100000000"
            settlementLimitType="NETTED"/>
          <ns1:counterParties>
            <ns1:settlementPair>
              <ns1:Entity1Id>metropolis_PB0</ns1:Entity1Id>
              <ns1:Entity2Id>metropolis_C4</ns1:Entity2Id>
            </ns1:settlementPair>
          </ns1:counterParties>
        </ns1:DslInfo>
      </CreditExposureInfo>
    </CreditExposureResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

```

        </ns1:settlementPair>
        <ns1:settlementPair>
            <ns1:Entity1Id>metropolis_B4</ns1:Entity1Id>
            <ns1:Entity2Id>metropolis_PB0</ns1:Entity2Id>
        </ns1:settlementPair>
    </ns1:counterParties>
</ns1:DslInfo>
<DailyExposureInfo>
    <date>2007-12-11</date>
    <exposure>20878000</exposure>
</DailyExposureInfo>
</CreditExposureInfo>
</CreditExposureResponse>
</soapenv:Body>
</soapenv:Envelope>

```

4.8 DslUpdateRequest

```

<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Header>
        <ns1:session
            soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
            soapenv:mustUnderstand="0"
            xmlns:ns1="http://currenex.com/web service/retail/schema">
            <ns1:userID>metropolis_c4u0</ns1:userID>
            <ns1:sessionTicket>
                58cf997fae1a2d88ef1fde130956ea10f9
            </ns1:sessionTicket>
        </ns1:session>
    </soapenv:Header>
    <soapenv:Body>
        <ns2:DslUpdateRequest ns2:dslRef="1"
            xmlns="http://currenex.com/web service/retail/schema"
            xmlns:ns2="http://currenex.com/web service/retail/schema">
            <ns2:amount>80000000</ns2:amount>
        </ns2:DslUpdateRequest>
    </soapenv:Body>
</soapenv:Envelope>

```

```

    </soapenv:Body>
</soapenv:Envelope>

```

4.9 DslUpdateResponse

4.9.1 Success

```

<?xml version="1.0" encoding="utf-8"?>

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <soapenv:Body>

    <ns1:DslUpdateResponse ns1:dslRef="1"
      xmlns="http://currenex.com/webservice/retail/schema"
      xmlns:ns1="http://currenex.com/webservice/retail/schema">

      <ns1:result>SUCCESS</ns1:result>

    </ns1:DslUpdateResponse>

  </soapenv:Body>

</soapenv:Envelope>

```

4.9.2 Failure

```

<?xml version="1.0" encoding="utf-8"?>

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <soapenv:Body>

    <ns1:DslUpdateResponse ns1:dslRef="2"
      xmlns="http://currenex.com/webservice/retail/schema"
      xmlns:ns1="http://currenex.com/webservice/retail/schema">

      <ns1:result>ERROR</ns1:result>

      <ns1:errorMessage>
        DSL update not allowed - not owner
      </ns1:errorMessage>

    </ns1:DslUpdateResponse>

  </soapenv:Body>

</soapenv:Envelope>

```

5 APPENDIX

5.1 Revision history

Revision Number	Revision Date	Page Number	Update
1	15 May 2013	7-8	Section 3.1 and Section 3.2 : Support added for Net Open Position and Accumulated Open Position.
2	24 July 2014	8	Section 3.3 : Added isResetUsedAmountToZero.