

Wire Data Model and Services 4.4

# Document Status

status: Request for Comment (valid values are < Request for Comment, Preliminary Review, Public Review, Architectural Review, Final Review, Published, Deprecated)

# Change Log

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 3.0 | **12/19/2013** | * Initial Document |
| 3.0 | **01/14/2014** | * Updated JSON Examples and Use Case |
| 3.0 | **04/02/2014** | * Updated JSON Examples and data attributes for XSD updates. |
| 3.1 | **07/17/2015** | * Updated to release 3.1 |
| 3.2 | **05/10/2016** | * Updated to release 3.2 |
| 3.3 | **02/15/2017** | * Updated to release 3.3 |
| 4.0 | **02/19/2018** | * Updated to release 4.0, Date Range Global Update, Microsoft Global bug fix. Replaced MicrAccountNumber and RoutingNumber with toAccountMicrAccountNumber, fromAccountMicrAccountNumber, toAccountRoutingNumber, and fromAccountRoutingNumber. |
| 4.1 | **12/10/2018** | * Updated to release 4.1 |
| 4.2 | **03/05/2019** | * Updated to release 4.2, \*\*\* Release 4.2 is a breaking fix release. \*\*\* Errors found in App, ArtifactFilter, and BillFilter required a breaking fix to align with the standard and prevent additional implementation difficulties going forward. |
| 4.3 | **10/07/2019** | * Updated to release 4.3, renamed file removing version as proper version control is being used in Github. |
| 4.4 | **10/20/2020** | * Updated to release 4.4 |

# Overview of Specification

The CUFX Wire Data Model and Service provides a standard description for wire payment definitions and instances of those payments. The CUFX Wire Data Model and Services provide the information necessary to create, read, update or delete Wire transactions between credit union data systems.

# Any know Errors in the document

|  |  |
| --- | --- |
| **Error Description** | Status of Error |
|  |  |

Table of Contents

[Document Status 1](#_Toc54093097)

[Change Log 1](#_Toc54093098)

[Overview of Specification 1](#_Toc54093099)

[Any know Errors in the document 1](#_Toc54093100)

[Document Conventions 2](#_Toc54093101)

[CUFX API and Documentation Support 3](#_Toc54093102)

[Release 4.0 Global Update Notes 3](#_Toc54093103)

[Release 4.4 Global Update Notes 3](#_Toc54093104)

[Definitions related to the specification 4](#_Toc54093105)

[Data Elements 5](#_Toc54093106)

[Filters used when accessing the WIRE data 5](#_Toc54093107)

[Wire Data attributes 5](#_Toc54093108)

[Use Case 6](#_Toc54093109)

[Services 6](#_Toc54093110)

[Authentication 6](#_Toc54093111)

[Wire-transfer Resource Based Create, Read, Update, Delete 7](#_Toc54093112)

[REST-JSON Create Wire-transfer 7](#_Toc54093113)

[REST-JSON Read Wire-transfers 8](#_Toc54093114)

[REST-JSON update Wire-transfer 9](#_Toc54093115)

[REST-JSON Delete pending wire-transfer 10](#_Toc54093116)

[General Error handling For All Services 11](#_Toc54093117)

[Bibliography 11](#_Toc54093118)

# Document Conventions

List any document conventions such as what bold and italics mean and how the document is intended to be read.

“Within this specification, the key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in W3 Working Group (W3C)]. However, for readability, these words do not appear in all uppercase letters in this specification.

At times, this specification recommends good practice for authors and user agents. These recommendations are not normative and conformance with this specification does not depend on their realization. These recommendations contain the expression “We recommend …”, “This specification recommends …”, or some similar wording.”

All formatting in this document utilize Word Styles.

All Citations must utilize Word Citations to automatically show at the end of the document.

All updates after the initial creation must be performed using Tracking Changes turn on and Accepted by the Architecture committee.

# CUFX API and Documentation Support

CUFX is published to SwaggerHub at <https://app.swaggerhub.com/apis/dlacroix01/CUFX> . The latest default version will load automatically.

The purpose of this publication is to demonstrate the full range of CUFX messaging. Provide for complete documentation of the entire message structure and example usage.

Swaggerhub as a normal support feature also translates into several dozen of the most common and in demand client / server formats. This provides a technology specific version of the CUFX standard for essentially all platforms.

The CUFX Specification documents have been written to provide for limited examples of case usage but do not reflect the entirety of elements present in any given message. Please see the latest version of CUFX in Swaggerhub for the complete message and for superior documentation support.

# Release 4.0 Global Update Notes

CUFX Release 4.0 introduces a number modifications that significantly improves the standard and is not backward compatible with prior versions.

Messaging paradigm shift. Prior to CUFX 4.0 a Message Object would be sent and would expect the Object List to be returned or the error message. The response had to be interrogated to determine what was received. With CUFX 4.0, the Object Message that is sent is also expected to be the Object that is returned. Significant improvements have been made to the Message Context to fully support Success, Informational, Warnings and Error responses. End Points may continue to use the prior methods, but use of the Error.xsd is depreciated; all functionality has transitioned into MessageContext.xsd.

Date Range Filtering. A global update was applied across the standard to remove the pairs of date filter elements for any given range and replaced with a single Common.xsd definition DateRange complex type. This makes date range filtering completely uniform across the standard and associates the startDateTime and endDateTime together as an object set.

As example: elements transactionStartDateTime and transactionEndDateTime were replaced in the AccountFilter.xsd with transactionDateRange.

Microsoft Serialization Bug. We discovered the root cause of a serialization error impacting CUFX. A known Microsoft Serialization error from 2006 is present for single element complex types. It causes a naming error of the serialized constructs. If both endpoints are using a Microsoft compilation the error is consistent and does not present itself, the names are both wrong but pass data successfully. When one end point is not using a Microsoft compilation, the field names are in variance and fails. If both end points are using non-Microsoft compilation the serialization would be correct and match.

CUFX 4.0 has applied a global update across all list types throughout the standard. The CUFX list construct was consistently a single element complex type. For all occurrences we have applied an extension base of common:ListBase. ListBase provides pagination support and also resolves the Microsoft serialization error. No longer being a single element complex type, Microsoft compilation now generates the correct names. This will necessitate prior (Microsoft) implementations to remap to the correct serialized names.

# Release 4.4 Global Update Notes

CUFX Release 4.4 introduces a significant enhancement for complex Account identification and filter navigation. The foundational architectural design premise for account navigation is that the CUFX AccountId would be a unique value unto itself within a given institution, or that a composite unique key would be passed. With the direct support of several core system providers it was established that that later case is predominate. The AccountId is generally not a unique value unless in combination with several other values such as AccountType and AccountSubType. Hence passing a unique AccountId meant that the organization had to overload the element value making filtering implementation specific and forcing the endpoints to map overlay the accountId to unpack the value.

CUFX now fully recognizes this architectural paradigm while continuing to support the original architecture.

A global update was applied to provide the elements accountType and accountSubType in all objects that contained accountId.

A new filter list – AccountIdentificationList has been added to all account related filters so that AccountId, AccountType, and AccountSubType can be structured properly for discreet filtering of complex account keys, support filtering by the sub keys and also support inbound and outbound account filtering using the accountToFromIndicator.

By expanding the architectural paradigm to support a non-unique AccountId CUFX is now positioned to better support core adoption of the standard.

# Definitions related to the specification

Account Data

The CUFX account Data model and Services defines the account data model and Services used by all specifications. Accounts are an abstract summary of loan, share and investment balances at a high level. Any financial product stored in a financial services platform optionally including any transactions associated with that account.

# Data Elements

## Filters used when accessing the WIRE data

Refer to Security Services documentation to understand what may be contained the header and processed by security procedures. When accessing the data include **MessageContext.xsd** so that the service can determine the scope of the request. Refer to recent CUFX messageContext Data and CUFX Security Services for use of MessageContext.xsd. Include any filter variables related to the request. See **WireFilter.xsd.**

## Wire Data attributes

All CUFX fields related to a card are defined in card.xsd. A summary of the attributes is listed here for reference.

Note: Fields not listed in the calling specification are not to be returned to the calling specification. i.e. If the field occurrenceMemo is not listed in the calling specification, then do not return the data field to alleviate issues with unexpected information and bloat of information being returned to light weight applications.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| occurrenceId | ID to uniquely identify the occurrence of the wire |
| recurringId | Not applicable to wire transactions |
| occurrenceAmount | Amount of the wire |
| occurrenceFromAccountId | The account from which the transfer or payment was taken. |
| occurrenceToAccountId | The account to which the transfer or payment was made. |
| occurrenceStatus | Status of the wire occurrence. |
| fundsWithdrawalType | When will the funds be withdrawn from the account |
| occurrenceMemo | The memo represents a short description that can be applied to the comment of the wire. |
| estimatedProcessDateTime | The date on which to process the wire. |
| createdDateTime | The date that the user submitted the wire. |
| processingStartedDateTime | The date on which the wire processing started. |
| processedDateTime | The date on which the wire was actually processed by the system. |
| completedDateTime | The date on which the core completed the wire. |
| lastUpdatedDateTime | The date of the last known status of the wire. |
| occurrenceConfirmationCode | Unique confirmation code for the wire. |
| queuedForPosting | Indicates whether the wire is pending for posting until system is on-line |
| note | A user's "note to self." |
| batchId | An identifier used to group a set of wires together to process them at the same time |
| customData | Name/data pairs list of fields that are not yet available in the general specification. |
| wireTransferType | Wire transfer type. |
| wireTransferSpecialInstructions | Special instructions for wire transfer. |
| escrowNumber | Escrow number for the wire transfer. |
| escrowOfficerName | Escrow officer name for the wire transfer. |

Fields in blue extended from FundsTransferOccurrenceBase.

# Use Case

CUFX Compliant Application

1

Create Wire Transfer Message

2

Wire Transfer Data Response

Contains Confirmation Code

3

Read Wire Transfer Message

Mapped to Account Id

4

Wire Transfer Data Message Response

Contains confirmation or error

CUFX Core Data Provider

5

Update Wire Transfer Message

6

Wire Transfer Data Message Response

Contains confirmation Code or error

7

Delete Wire Transfer Message

8

Wire Transfer Data Response

Contains Confirmation Code or error

# Services

|  |  |
| --- | --- |
| Definition | WireTransfer Service |
| Overview of Capabilities | Service to create, read, update and delete Wire-Transfer |
| Dependencies | Security Services, messageContext |
| CUFX REST LINK | https://api.dataprovider.com/wiretransfermessage |
| CUFX WSDL LINK |  |

## Authentication

See Security Services and messageContext for more detail.

## Wire-transfer Resource Based Create, Read, Update, Delete

|  |  |
| --- | --- |
| INPUTS | cufx: WireMessage (which includes)   * [cufx:messageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:wireFilter (for read, update, delete) * cufx:wireList (for create, update) |
| Outputs | cufx: WireMessage (which includes)   * [cufx:messageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:wireList (for create, update) |
| Return Values | cufx: WireMessage (which includes)   * [cufx:messageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html)   + statusList |
| Side Effects | Creation, read, update or deletion of single wire-transfer. |
| Dependencies | Security Services for authentication and security. |
| Fields used | Message Headers : See security services  messageContext: See messageContext.xsd  Filters: See DepositFilter.xsd  Attributes: Transfer : See Wires.xsd |
| Testing Procedures for Certification |  |

### REST-JSON Create Wire-transfer

REQUEST:

**<Security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us

Content-type: application/json; charset=utf-8

X-API-Version: >=4.3.0

**POST** https://api.dataprovider.com/wiretransfermessage

{

wireMessage: {

"messageContext" : {<See MessageContext.xsd>

},

"wireList" : {

"Wire”: [{

"occurrenceId": "35464736635",

"occurrenceAmount": {

"value": "234.89",

"currencyCode": "USD"

},

"occurrenceFromAccountId" : account:AccountId,

"createdDateTime" : "2014-01-01T01:01:01Z",

"WireTransferType”: "Escrow",

"WireTransferSpecialInstructions”: xs:string,

"EscrowNumber”: xs:string,

"EscrowOfficerName”: xs:string

}]

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

wireMessage: {

"messageContext" : {<See MessageContext.xsd>

},

"wireList" : {

"Wire”: [{

"occurrenceId": "35464736635",

"occurrenceAmount": {

"value": "234.89",

"currencyCode": "USD"

},

"occurrenceFromAccountId" : account:AccountId,

"createdDateTime" : "2014-01-01T01:01:01Z",

"occurrenceConfirmationCode": "occurrenceConfirmationCode1",

"WireTransferType”: "Escrow",

"WireTransferSpecialInstructions”: xs:string,

"EscrowNumber”: xs:string,

"EscrowOfficerName”: xs:string

}]

}

}

}

### REST-JSON Read Wire-transfers

REQUEST:

**<Security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us

Content-type: application/json; charset=utf-8

X-HTTP-Method-Override:GET

X-API-Version: >=4.3.0

**POST** https://api.dataprovider.com/wiretransfermessage

{

wireMessage: {

“messageContext”: { <see messageContext.xsd>

},

“wireFilter”: {

“occurrenceIdList” : {

“occurrenceId” : [ “452344”]

},

“accountIDList” {

“accountId”: [ “1234” ]

},

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

wireMessage: {

"messageContext" : {<See MessageContext.xsd>

},

"wireList" : {

"Wire”: [{

"occurrenceId": "452344",

"occurrenceAmount": {

"value": "234.89",

"currencyCode": "USD"

},

"occurrenceFromAccountId" : “1234”,

"createdDateTime" : "2014-01-01T01:01:01Z",

"occurrenceConfirmationCode": "occurrenceConfirmationCode1",

"WireTransferType”: "Escrow",

"WireTransferSpecialInstructions”: xs:string,

"EscrowNumber”: xs:string,

"EscrowOfficerName”: xs:string

}]

}

}

}

### REST-JSON update Wire-transfer

REQUEST:

**<Security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us

Content-type: application/json; charset=utf-8

X-API-Version: >=4.3.0

**PUT** https://api.dataprovider.com/wiretransfermessage

{

wireMessage: {

“messageContext”: { <see messageContext.xsd>

},

“wireFilter”: {

"occurrenceIdList": {

"occurrenceId": ["452344"}

},

"Wire”: [{

"occurrenceId": "452344",

"occurrenceAmount": {

"value": "310.89",

"currencyCode": "USD"

},

"occurrenceFromAccountId" : “1234”,

"createdDateTime" : "2014-01-01T01:01:01Z",

"WireTransferType”: "Escrow",

"WireTransferSpecialInstructions”: xs:string,

"EscrowNumber”: xs:string,

"EscrowOfficerName”: xs:string

}]

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

wireMessage: {

"messageContext" : {<See MessageContext.xsd>

},

"wireList" : {

"Wire”: [{

"occurrenceId": "452344",

"occurrenceAmount": {

"value": "310.89",

"currencyCode": "USD"

},

"occurrenceFromAccountId" : “1234”,

"createdDateTime" : "2014-01-01T01:01:01Z",

"occurrenceConfirmationCode": "occurrenceConfirmationCode1",

"WireTransferType”: "Escrow",

"WireTransferSpecialInstructions”: xs:string,

"EscrowNumber”: xs:string,

"EscrowOfficerName”: xs:string

}]

}

}

### REST-JSON Delete pending wire-transfer

REQUEST:

**<Security related header parameters... see Security Services>**

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us

Content-type: application/json; charset=utf-8

X-HTTP-Method-Override: DELETE

X-API-Version: >=4.3.0

**PUT** https://api.dataprovider.com/wiretransfermessage

{

wireMessage: {

“messageContext”: { <see messageContext.xsd>

},

“wireFilter”: {

“occurrenceIdList” : {

“occurrenceId” : [ “452344”]

},

“accountIDList” {

“accountId”: [ “1234” ]

},

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

Content-type: application/json; charset=utf-8

Content-Language: en-us

Payload:

{

wireMessage: {

"messageContext" : {<See MessageContext.xsd>

}

}

# General Error handling For All Services

Refer to latest CUFX documentation *Error Mapping*.

# Bibliography

W3C. (n.d.). *Key words for use in RFCs to Indicate Requirement Levels [RFC2119].* Retrieved Sept. 8th, 2011, from W3C.