

Investment 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)

# Authors and Change Log

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 3.3 | **02/15/2017** | * Initial Creation |
| 4.0 | **02/19/2018** | * Updated to release 4.0, Date Range Global Update, Microsoft Global bug fix, Removed and normalized common:IntervalFrequencyType into common:FrequencyType, Removed investmentAccountStatus and investmentAccountStatus and restructed AccountStatus to choice of Deposit,Loan, and Investment. Consolidated to Acccount.xsd. Corrected element name investmentFilter to investmentHoldingFilter to match naming convention. |
| 4.1 | **12/10/2018** | * Updated to release 4.1, Global removal of choice constructs to eliminate cross language serialization issues. * Global update applied for minOccurs 0 maxOccurs 1. Xsd had 1 or more elements corrected to be consistent with the standard definition. |
| 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. * InvestmentHolding - Added enums to holding type. |
| 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, Account - Deprecated type. Use accountType in accountBase. Documented meta usage. * Added interestPostingFrequency, withholdingsInfoList, overDraftProtectionAccountId, overDraftProtectionAccountType, overDraftProtectionAccountSubType. * Added enums to accountType for Installment, CommercialLoan, CommercialLineOfCredit, GeneralLedger. OtherLoan. * DeprecatedAccountAccountTypeList and AccountAccountTypePair. * Added Type AccountIdentificationList, AccountIdentification, AccountSubType * Added enums ChargedOff, ChargeOffClosed to DepositAccountStatus * Account Filter - includeNotesFlag type to common:IncludeNotesFlag. Added accountIdentificationList * Investment - Added customData to InvestmentParty * InvestmentHolding - Added accountType, accountSubType and customData to InvestmentHolding * InvestmentHoldingFilter - Added accountIdentificationList to filter. |
|  |  |  |

# Overview of Specification

This specification defines the features to read, update or delete an investment account and/or investment holding.

# Any know Errors in the document

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

Table of Contents

[Document Status 1](#_Toc54096310)

[Authors and Change Log 1](#_Toc54096311)

[Overview of Specification 2](#_Toc54096312)

[Any know Errors in the document 2](#_Toc54096313)

[Document Conventions 2](#_Toc54096314)

[CUFX API and Documentation Support 3](#_Toc54096315)

[Release 4.0 Global Update Notes 3](#_Toc54096316)

[Release 4.4 Global Update Notes 4](#_Toc54096317)

[Definitions related to the specification 4](#_Toc54096318)

[Data Elements 4](#_Toc54096319)

[Filters Used When Accessing the Investment Data 4](#_Toc54096320)

[Investment Data Attributes 4](#_Toc54096321)

[Use Case 5](#_Toc54096322)

[Services 5](#_Toc54096323)

[Authentication 5](#_Toc54096324)

[Investment Resource Based Create, Read, Update, Delete investment Account data 6](#_Toc54096325)

[REST-JSON READ Investment ACCOUNT 6](#_Toc54096326)

[Investment Holding priority Resource Based Create, Read, Update, Delete Data 8](#_Toc54096327)

[REST-JSON READ Investment Holding 8](#_Toc54096328)

[General Error handling For All Services 10](#_Toc54096329)

[Bibliography 10](#_Toc54096330)

# 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

Investment Account

An asset account for a financial institution or its clients.

# Data Elements

## Filters Used When Accessing the Investment 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. Include any filter variables related to the request. See **investmentFilter.xsd** or **InvestmentHolding.xsd**

The filters are used to filter based on associated investment account data. Combining filters can be used to get a response or affect specific investment data.

## Investment Data Attributes

All CUFX fields related to an investment are defined in **investment.xsd**.

Note: Fields not listed in the calling specification are not to be returned to the calling specification. i.e. If the field transaction type 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.

# Use Case

CUFX Compliant Application

1

InvestmentMessage Request

2

InvestmentMessage contains messageContext and investmentList

CUFX Core Data Provider

2

InvestmentHoldingMessage contains messageContext and investmentHoldingList

1

InvestmentHoldingMessage Request

# Services

|  |  |
| --- | --- |
| Definition | Investment account |
| Overview of Capabilities | Service to create investment accounts and set dependent data. |
| Dependencies | Security Services, messageContext |
| CUFX REST LINK | https://api.dataprovider.com/investmentmessage |
| CUFX WSDL LINK |  |

## Authentication

See Security Services and messageContext for more detail.

## Investment Resource Based Create, Read, Update, Delete investment Account data

|  |  |
| --- | --- |
| INPUTS | cufx: investmentMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * Cufx:investmentFilter (for read, update) * cufx:investmentList (for create, update, delete) |
| Outputs | cufx: investmentMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:investmentList |
| Return Values | cufx: investmentMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html)   + statuslist |
| Side Effects | Creation, update or deletion of investment account. |
| Dependencies | Security Services for authentication and security. |
| Fields used | Message Headers : See security services  messageContext: See messageContext.xsd  Filters: See InvestmentFilter.xsd  Attributes: investmentList : See Investment.xsd |
| Testing Procedures for Certification | Send details for a new investment account and verify that it has been created. |

### REST-JSON READ Investment ACCOUNT

This example shows reading an investment account. Note: Not all fields are listed for simplicity of the example.

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/investment>message

{

"investmentMessage": {

"messageContext": {

"cufxVersion": "4.3.0",

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

},

"investmentFilter": {

"accountIdList": {

"accountId": "1234567"

},

"accountStatusList": {

"accountStatusPair": {

"accountStatus": {

"investmentStatus": "Active"

}

}

},

"partyIdList": {

"partyId": "373652"

}

}

}

}

RESPONSE

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"investmentMessage": {

"messageContext": {

"cufxVersion": "4.3.0",

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

"dataSourceId": "dataSourceId1",

"environment": "Development",

"returnDataFilter": "All",

"includeBlankFields": "true",

"includeZeroNumerics": "true",

"user": {

"userId": "userId1",

"processorSessionId": "processorSessionId1",

"userType": "EmployeeId"

},

"statusList": {

"status": {

"statusType": "Success"

}

},

},

"investmentList": {

"investment": {

"accountId": "1234567",

"type": "Investment",

}

}

}

}

}

## Investment Holding priority Resource Based Create, Read, Update, Delete Data

|  |  |
| --- | --- |
| INPUTS | cufx:investmentHoldingMessage (which includes)   * [cufx:messageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:investmentHoldingFilter (for read, update, delete) * cufx:investmentHoldingList (for create, update) |
| Outputs | cufx:investmentHoldingMessage (which includes)   * [cufx:messageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:investmentHoldingList |
| Return Values | cufx:investmentHoldingMessage (which includes)   * [cufx:messageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html)   + statusList |
| Side Effects | Read has no side effects. |
| Dependencies | Security Services for authentication and security. |
| Fields used | Message Headers : See security services  messageContext: See messageContext.xsd  Filters: See InvestmentHoldingFilter.xsd  Attributes: investmentHoldingList : See InvestmentHolding.xsd |
| Testing Procedures for Certification |  |

### REST-JSON READ Investment Holding

Description: Read investment holding for an investment account(s)

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/investmentholdingmessage

{

"investmentHoldingMessage": {

"messageContext": {

"cufxVersion": "4.3.0",

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

}

},

"investmentHoldingFilter": {

"accountIdList": {

"accountId": "7876355"

},

"holdingIdList": {

"holdingId": "987633422"

}

}

}

}

**RESPONSE**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

"investmentHoldingMessage": {

"messageContext": {

"cufxVersion": "4.3.0",

"requestId": "requestId1",

"vendorId": "vendorId1",

"appId": "appId1",

"fiId": "fiId1",

},

"statusList": {

"status": {

"statusType": "Success"

}

},

"investmentHoldingList": {

"investmentHolding": {

"accountId": "7876355",

"holdingID": "987633422",

"holdingCusipNumber": "holdingCusipNumber1",

"holdingAsOfDateTime": "1900-01-01T01:01:01Z",

"holdingType": "Asset",

"holdingDescription": "holdingDescription1",

"holdingQuantity": "1",

"holdingPrice": {

"value": "1",

"currencyCode": "USD",

"exchangeRate": "1"

},

"holdingPriceDateTime": "1900-01-01T01:01:01Z",

"holdingStockSymbol": "holdingStockSymbol1",

"holdingTotalValue": {

"value": "1",

"currencyCode": "USD",

"exchangeRate": "1"

},

"holdingYield": "1",

"holdingYieldAsOfDate": "1900-01-01T01:01:01Z",

"holdingAssetClass": "holdingAssetClass1",

"holdingBondClass": "holdingBondClass1",

"holdingBondType": "holdingBondType1",

"holdingMoodyRating": "holdingMoodyRating1",

"holdingStandardAndPoorsRating": "holdingStandardAndPoorsRating1",

"holdingParValue": {

"value": "1",

"currencyCode": "USD",

"exchangeRate": "1"

},

"holdingCouponRate": "1",

"holdingCouponFrequency": "AmortizedSchedule",

"holdingCallPrice": {

"value": "1",

"currencyCode": "USD",

"exchangeRate": "1"

},

"holdingYieldToCall": "1",

"holdingCallType": "holdingCallType1",

"holdingYieldToMaturity": "1",

"holdingMaturityDate": "1900-01-01T01:01:01Z",

"holdingNextCouponDate": "1900-01-01T01:01:01Z",

"holdingNextCallDate": "1900-01-01T01:01:01Z"

}

}

}

}

# 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.