

Party Association Data and Services 4.2

# Document Status

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

This version: **Assembla**.com. Files Tag = CUFX\_4.2\_RFC\_Active

Previous Version: **Assembla**.com. Files Tag = CUFX\_4.1\_RFC\_Archive

# Authors and Change Log

|  |  |  |
| --- | --- | --- |
| Version | Date | Changes |
| 0.0.01 |  | * Initial Draft |
| 0.0.02 |  | * Updated Overview of Specification |
| 0.0.03 |  | * Updated XSD filenames to PascalCase |
| 0.0.04 |  | * Updated filters to use simple data types |
| 0.0.05 |  | * Fix issue with update URL |
| 0.0.06 |  | * Update Id’s to be consistent with the XSD |
| 3.0 | **10/29/2013** | * Switch to use X-HTTP-METHOD-OVERRIDE standard rather than subMethod non-Standard method for overriding request types * Create a partyAssociationMessage wrapper for every message to increase ability for infrastructure to serialize the data * Versioning and format change with release CUFX 3.0 |
| 3.0 | **12/16/2013** | * Update examples X-API-Version to >=3.0.0 |
| 3.0 | **12/19/2013** | * Cleanup definitions, fix typos and prep for release |
| 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 | **02/01/2018** | * Updated to release 4.0, Date Range Global Update, Microsoft Global bug fix |
| 4.1 | **12/10/2018** | * Updated to release 4.1, 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. |
| 4.2 | **05/17/2019** | * Removed version from file naming. Github version control in place from this date forward. See <https://github.com/cufx/documenthistory> for prior document publication history. |

# Overview of Specification

The CUFX Party Association Data Model and Services defines the Party Association Data object and services for use by all specifications. The party association allows an organization to manage and view if parties work for, are related to, or have a legal responsibility to another party.

# Any knowN Errors in the document

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

# Table of Contents

[Document Status 1](#_Toc8979444)

[Authors and Change Log 1](#_Toc8979445)

[Overview of Specification 2](#_Toc8979446)

[Any knowN Errors in the document 2](#_Toc8979447)

[Table of Contents 2](#_Toc8979448)

[Document Conventions 2](#_Toc8979449)

[Release 4.0 Global Update Notes 3](#_Toc8979450)

[Definitions related to the specification 3](#_Toc8979451)

[Data Elements 3](#_Toc8979452)

[Filters used when accessing the party association data 3](#_Toc8979453)

[Party Data attributes 4](#_Toc8979454)

[Party Services 4](#_Toc8979455)

[Overview 4](#_Toc8979456)

[Party Resource based create, read, update, delete services 4](#_Toc8979457)

[REST-JSON CREATE Party Association example 5](#_Toc8979458)

[REST-JSON READ Party Association example 5](#_Toc8979459)

[REST-JSON UPDATE Party Association example 6](#_Toc8979460)

[REST-JSON DELETE Party Association example 7](#_Toc8979461)

[General Error handling For All Services 8](#_Toc8979462)

[Bibliography 8](#_Toc8979463)

# 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 utilizes Word Styles.

All Citations must utilize Word Citations so that it automatically shows at the end of the document.

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

# 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.

# Definitions related to the specification

**Party Association**

Any person or entity can be associated to another person or entity through a party association. For instance a party association would identify if a person is an officer of a company or has power of attorney over another party.

# Data Elements

## Filters used when accessing the party association data

Refer to Security Services documentation to understand what may be contained in 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 **PartyAssociationFilter.xsd.**

## Party Data attributes

All CUFX fields related to a party are defined in PartyAssociation.xsd.

Note: Fields not listed in the calling specification are not to be returned to the calling specification. That is, 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 a bloat of information being returned to lightweight applications.

# Party Services

## Overview

|  |  |
| --- | --- |
| Definition | Collection of services to manage a party association |
| Overview of Capabilities | Create, read, update and delete a party associations. You can programmatically filter party associations using the filter object for read, update and delete. It is assumed that the parties already exist on the back end system and can be connected through two party Id’s. It is possible to use the services to define more than one association. |
| Dependencies | Security Services, messageContext, partyAssociation |
| Sample CUFX REST LINK | https://api.dataprovider.com/partyassociationmessage |
| CUFX SOAP LINK |  |
| CUFX WaDL LINK |  |

## Party Resource based create, read, update, delete services

|  |  |
| --- | --- |
| INPUTS | cufx:partyAssociationMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:partyAssociationFilter (for read, update, delete) * cufx:partyAssociationlist (for create, update) |
| Outputs | cufx:partyAssociationMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:partyAssociationlist |
| Return Values | cufx:partyAssociationMessage (which includes)   * [cufx:MessageContext](file:///\\files2\users\CMarjaniemi\Projects\CUFX\MessageContext.html) * cufx:partyAssociationFilter (for read, update, delete)   + statusList |
| Side Effects | Creation, update or deletion of partyAssociation data; read has no side effects |
| Dependencies | Security Services for authentication and security. Both parties already exist on the back end system. |
| Fields used | Message Headers : See security services  messageContext: See MessageContext.xsd  Filters: See PartyAssociationFilter.xsd  Attributes: partyAssociation : See PartyAssociation.xsd |

### REST-JSON CREATE Party Association example

This example shows how to create an association between two parties. One is an individual and one is an organization.

**Required**: messageContext, two parties with the association type between the two parties

**REQUEST:**

Headers:

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

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

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

X-API-Version: >=4.2.0

**POST h**ttps://api.datasource.com/partyassociationmessage

{

“partyAssociationMessage”:{

“messageContext”: { <see MessageContext.xsd>

},

“partyAssociationList”: [

{“parentPartyId”:”1583471356asdflkj154”,

“childPartyId”:”75834165634n534h”,

“partyAssociationType”:

{“officerManager”:

{“qualifier”:”President”}

}

}

]

}

}

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

“partyAssociationMessage”:{

“messageContext”: { <see MessageContext.xsd>

},

“partyAssociationList”: [

{“partyAssociationId”:”153434”,

“parentPartyId”:”1583471356asdflkj154”,

“childPartyId”:”75834165634n534h”,

“partyAssociationType”:

{“officerManager”:

{“qualifier”:”President”}

}

}

]

}

### REST-JSON READ Party Association example

Note: This example reads a party association based on a known party association id.

**Required**: messageContext, at least one valid filter in partyAssociationFilter

**REQUEST:**

Headers:

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

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

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

**X-HTTP-Method-Override: GET**

X-API-Version: >=4.2.0

**POST h**ttps://api.datasource.com/partyassociationmessage

{

“partyAssociationMessage”:{

“messageContex”: { <see MessageContext.xsd>

},

“partyAssociationFilter”:{

“partyAssociationIdList”:[”153434”]

}

}

}

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

“partyAssociationMessage”:{

“messageContext”: { <see MessageContext.xsd>

},

“partyAssociationList”: [

{“partyAssociationId”:”153434”,

“parentPartyId”:”1583471356asdflkj154”,

“childPartyId”:”75834165634n534h”,

“partyAssociationType”:

{“officerManager”:

{“qualifier”:”President”}

}

}

]

}

}

### REST-JSON UPDATE Party Association example

Note: This example updates a party association as the individual is now CEO of the company.

**Required**: messageContext, at least one valid filter in partyAssociationFilter, the specific partyAssociationId of the partyAssociation to be updated in the message

**REQUEST:**

Headers:

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

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

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

X-API-Version: >=4.2.0

**PUT h**ttps://api.datasource.com/partyassociationmessage

{

“partyAssociationMessage”:{

“messageContex”: { <see MessageContext.xsd>

},

“partyAssociationFilter”:{

“partyAssociationIdList”:[ ”153434” ]

},

“partyAssociationList”: [

{“partyAssociationId”:”153434”,

“partyAssociationType”:

{“officerManager”:

{“qualifier”:”CEO”}

}

}

]

}

}

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

“partyAssociationMessage”:{

“messageContext”: { <see MessageContext.xsd>

},

“partyAssociationList”: [

{“partyAssociationId”:”153434”,

“partyAssociationType”:

{“officerManager”:

{“qualifier”:”CEO”}

}

}

]

}

}

### REST-JSON DELETE Party Association example

Note: This example deletes a party association.

**Required**: messageContext, at least one valid filter in partyAssociationFilter, the specific partyAssociationid of the partyAssociation to be deleted in the message

**REQUEST:**

Headers:

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

Accept: application/json

Accept-Charset: utf-8

Accept-Language: en-us (IANA – language codes)(W3C, HTTP Protocols)

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

**X-HTTP-Method-Override: DELETE**

X-API-Version: >=4.2.0

**PUT h**ttps://api.datasource.com/partyassociationmessage

{

“partyAssociationMessage”:{

“messageContex”: { <see MessageContext.xsd>

},

“partyAssociationFilter”:{

“partyAssociationIdList”:[ ”153434” ]

},

}

}

**RESPONSE:**

Headers:

Status Code: 200 Ok

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

Content-Language: en-us

Payload:

{

“partyAssociationMessage”:{

“messageContext”: { <see MessageContext.xsd>

}

}

}

# General Error handling For All Services

Refer to latest CUFX documentation *Error Mapping*.

Bibliography

*E.164.* (n.d.). Retrieved 06 28, 2012, from International PUblic Telecommunications Number Plan: http://www.itu.int/rec/T-REC-E.164/en

*North American Number Plan Administration.* (n.d.). Retrieved 06 28, 2012, from North American Number Plan Administration: http://www.nanpa.com/

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