

SIF – Generic Web Service Layer

Table of Contents

1	About	4
1.1	Usage	4
1.2	Authentication and ADContextUser.....	4
1.3	Contact identifiers	5
1.4	Validation of personal id number and enterprise number	5
1.5	Code values.....	5
1.6	Common elements.....	6
2	Contact service	7
2.1	SynchronizeContactPerson.....	7
2.1.1	Input	7
2.1.2	Output.....	8
2.2	SynchronizePrivatePerson.....	8
2.2.1	Input	9
2.2.2	Output.....	9
2.3	SynchronizeEnterprise.....	10
2.3.1	Input	11
2.3.2	Output.....	12
2.4	GetContactPersons	12
2.4.1	Input	12
2.4.2	Output.....	12
2.5	GetPrivatePersons.....	13
2.5.1	Input	13
2.5.2	Output.....	13
2.6	GetEnterprises.....	15
2.6.1	Input	15
2.6.2	Output.....	15
2.7	Sub Contracts.....	16
2.7.1	Data Contract: Address.....	16
2.7.2	Data Contract: ContactPersonBase	17
2.7.3	Data Contract: PrivatePersonBase	17

2.7.4	Data Contract: EnterpriseResult	17
3	User service.....	19
3.1	SynchronizeUser	19
3.1.1	Input	19
3.1.2	Output.....	20
4	Case service	20
4.1	CreateCase	20
4.1.1	Input	20
4.1.2	Output.....	23
4.2	UpdateCase.....	24
4.2.1	Input	24
4.2.2	Output.....	26
4.3	GetCases.....	28
4.3.1	Input	28
4.3.2	Output.....	28
4.4	Sub Contracts	30
4.4.1	Data Contract: ClassCodeParameter	30
4.4.2	Data Contract: CaseResult	30
5	Document service	31
5.1	CreateDocument.....	31
5.1.1	Input	31
5.1.2	Output.....	34
5.2	UpdateDocument.....	34
5.2.1	Input	35
5.2.2	Output.....	37
5.3	GetDocuments.....	39
5.3.1	Input	39
5.3.2	Output.....	39
5.4	Sub Contracts	40
5.4.1	Data Contract: DocumentResult	40
5.4.2	Data Contract: DocumentContactResult	41
5.4.3	Data Contract: File	43
6	File service	43
6.1	Upload	43
6.1.1	Input	44

6.1.2	Output.....	44
7	Support methods	45
7.1	Ping	45
8	Dependencies to 360	45
8.1	PDF Converting	45
9	Web Service client responsibility	45

1 About

This document describes the Web service layer of SI Integration Framework (SIF), which can be used by external systems to create, update and read data in 360.

Four types of methods are provided:

- Synchronize methods
- Create methods
- Update methods
- Get methods

Create methods are used to create new cases and documents, update methods to update them. Synchronize methods are used to create *or* update contacts and users, depending on whether they exist already. Get methods search for cases, documents and contacts.

1.1 Usage

There are five web service URL's. The exact URL depends on the server configuration, but in most cases they will be something like

[http://\[servername\]:8088/SI.WS.Core/SIF/ContactService.svc](http://[servername]:8088/SI.WS.Core/SIF/ContactService.svc)

.. or CaseService.svc, DocumentService.svc, UserService.svc, FileService.svc

WSDL's for these services are provided along with this document. WSDL specifies the functionality of a web service, and can be used to build the client code.

The default service endpoints use Soap 1.1, conform to WS-I Basic Profile 1.2, and use Basic authentication. Each service (apart from FileService) also has alternative endpoints with different configurations. These are accessed by adding a postfix to the URL.

Endpoint	Soap	Authentication	Other
.svc	1.1	Basic	
.svc/Basic	1.1	Basic	
.svc/Ntlm	1.1	Ntlm	
.svc/BasicMtom	1.1	Basic	Mtom
.svc/NtlmMtom	1.1	Ntlm	Mtom

Other authentication methods and Soap 1.2 can be configured if required.

1.2 Authentication and ADContextUser

The ADContextuser property is supported by all the web service methods, and contains the login name of a valid 360 user. If provided, the operation will execute as (impersonate) this user. If not provided, the operation will execute as the user that has authenticated against the web service.

Only some users are allowed to impersonate other users. For instance, if the client authenticates against the service with the user "domain\administrator", and specifies "domain\user1" as ADContextUser, "domain\administrator" must have impersonation rights in 360. This is specified in the configuration file of the web service.

Typical scenario 1: The client of the web service is Application X. A service user is created for this application, “domain\appx_svc”, which is configured with impersonation rights. The client authenticates with “domain\appx_svc”, but provides the login of the end user, “domain\user1”, as ADContextUser. The operation is then executed as user1. For instance, if a case is created, user1 will be logged as the user who created it.

Typical scenario 2: Same as 1, except no ADContextUser is provided. In this case, the operation will be executed as appx_svc. If a contact is created, appx_svc will be logged as the user who created it.

It is recommended that a dedicated service user is created either for all clients to use in common (“domain\360sif_svc”), or for each of them to use separately (“appx_svc”, “appy_svc”, ..). If ADContextUser is provided, it’s enough with one common service user. If ADContextUser is not provided, it is recommended to have a service user for each client, so that the 360 log shows where the data came from. For instance, if both Application X and Application Y create contacts in 360, it is useful to know which of them created a specific contact.

1.3 Contact identifiers

There are three types of registered contacts in 360: Enterprises (/companies/departments), private persons and contact persons. A contact person is associated with an enterprise, a private person is not. A user always has a contact person, which contains the contact metadata of that user.

In the SIF web service, contacts are identified by their PersonalIdNumber (“fødselsnummer”) if they’re private persons, EnterpriseNumber (“organisasjonsnummer”) if they’re enterprises, or ExternalId if they’re contact persons. For instance, if you’re synchronizing a private person with PersonalIdNumber 1234567890, the service will look for a contact with that id, and update it if it exists.

When multiple external systems work with 360 contacts, or when there are existing contacts that should be referred to / updated by the web service, care must be taken to prevent duplicates and other problems. This should be analyzed before the system is put into production.

1.4 Validation of personal id number and enterprise number

When synchronizing private persons and enterprises, the service can optionally validate that the values comply with Norwegian fødselsnummer and organisasjonsnummer formats (11 and 9 digits, and satisfying a modulus 11 validation).

This functionality is toggled by the “SI.Data.WS.ValidateContactId” appSetting in the SIF web.config file. By default, it is set to false.

1.5 Code values

Some methods accept input in the form of code table values. For instance, when creating a case, you can set its Status property to a value from the Case status code table, such as “B”.

Since code tables are localized, with different code values for each language, it’s important to use codes from the right language. The default language for the SIF web service is Norwegian, but this may vary. (The default language is configured in the appSetting SI.Data.BL.LanCode in the SIF web.config file.)

All properties that accept code values can alternatively accept the recno of the code, on the format "recno:123". It is less risky to specify the code by recno, since it's possible (but rare) for code values to change.

1.6 Common elements

(*)

This tag behind a property name means that the property is mandatory.

(\)

This tag behind property name means that at least one of properties with this symbol needs to be supplied.

2 Contact service

Typical URL: [http://\[servername\]:8088/SI.WS.Core/SIF/ContactService.svc](http://[servername]:8088/SI.WS.Core/SIF/ContactService.svc)

2.1 SynchronizeContactPerson

Creates or updates a contact person contact in 360 based on ExternalId. If the supplied ExternalId does not exist in 360 the contact is created. If the supplied ExternalId exists in 360 the contact is updated to mirror the supplied information.

In the scenario where there exists several contacts with the same ExternalId the oldest (lowest recno) will be updated, while an error message is logged to TraceView and the Windows Event Viewer. It is not possible that this web service method itself creates a duplicate of a contact, given that the ExternalId is correct.

OBS: If a property is not set, meaning it is blank or null; the corresponding value in 360 will be set to an empty value. This behavior is unique to the contact related synchronization methods.

2.1.1 Input

Data Contract: SynchronizeContactPersonParameter

Input Property	360° field	Comment
ADContextUser	-	Read the general information section.
FirstName(*)	Contact – Name2	Mandatory when creating the contact
MiddleName	Contact – Name2	Middle name of the contact, concatenated to FirstName.
LastName(*)	Contact – Name3	Mandatory when creating the contact
ExternalId(*)	Contact – ExternalID	The external id number of the contact used as a unique identifier for the contact in 360. If unique, a new contact is created, otherwise the contact is updated. If more than one external system use the web service, external id's must be unique also between external systems.
Enterprise(*)	Contact – ToEmployer	The OrgNo of the enterprise the contact is connected to, most likely as an employee.
Categories	Contact – Categories	List of categories, identified by their code name in the configured language, or by recno on the format "recno:123". If this property is not specified (null), no changes will be made to the user's existing categories.
Email	Contact – TeleObject 6	
PhoneNumber	Contact – TeleObject 1	

MobilePhone	Contact – TeleObject 27	
PrivateAddress	Contact - Addresses	PrivateAddress is an address object, see fields needed for input under “ Data Contract: Address ”.
PostAddress	Contact - Addresses	PostAddress is an address object, see fields needed for input under “ Data Contract: Address ”.
DataSource	Log tab	DataSource is presented as a log entry under the Log-tab on the contact. The point is to track which systems have made updates to the contact. No logging is done if this property is empty or not set.

2.1.2 Output

Data Contract: SynchronizeContactPersonResult

Output Property	360° field	Comment
Recno	Contact – Recno	Unique ID in 360 for the contact
Successful	-	If true, everything is ok. If false, the operation failed.
ErrorMessage	-	If unsuccessful, this contains a short error message
ErrorDetails	-	If unsuccessful, this contains a full stack trace

2.2 SynchronizePrivatePerson

This creates or updates a private person contact in 360 based on PersonalIdNumber. If the supplied PersonalIdNumber does not exist in 360 the contact is created. If the supplied PersonalIdNumber exists in 360 the contact is updated to mirror the supplied information.

In the scenario where there exists several contacts with the same PersonalIdNumber the oldest (lowest recno) will be updated, while an error message is logged to TraceView and the Windows Event Viewer. It is not possible that this web service method itself creates a duplicate of a contact, given that the PersonalIdNumber is correct.

OBS: If a property is not set, meaning it is blank or null; the corresponding value in 360 will be set to an empty value. This behavior is unique to the contact related synchronization methods.

2.2.1 Input

Data Contract: SynchronizePrivatePersonParameter

Input Property	360° field	Comment
ADContextUser	-	Read the general information section.
FirstName(*)	Contact – Name2	Mandatory when creating the contact
MiddleName	Contact – Name2	Middle name of the contact, concatenated to FirstName.
LastName(*)	Contact – Name3	Mandatory when creating the contact
PersonalIdNumber(*)	Contact – Referencenumber	The personal identity number of the contact used as a unique identifier for the contact in 360. If unique, a new contact is created, otherwise the contact is updated. Has to be 11 digits long and pass a mod11 validation.
Email	Contact – TeleObject 6	
PhoneNumber	Contact – TeleObject 1	
MobilePhone	Contact – TeleObject 27	
PrivateAddress	Contact - Addresses	PrivateAddress is an address object, see fields needed for input under “ Data Contract: Address ”.
PostAddress	Contact - Addresses	PostAddress is an address object, see fields needed for input under “ Data Contract: Address ”.
DataSource	Log tab	DataSource is presented as a log entry under the Log-tab on the contact. The point is to track which systems have made updates to the contact. No logging is done if this property is empty or not set.

2.2.2 Output

Data Contract: SynchronizePrivatePersonResult

Output Property	360° field	Comment
Recno	Contact – Recno	Unique ID in 360 for the contact
Successful	-	If true, everything is ok. If false, the operation failed.
ErrorMessage	-	If unsuccessful, this contains a short error message

ErrorDetails	-	If unsuccessful, this contains a full stack trace
--------------	---	---------------------------------------------------

2.3 SynchronizeEnterprise

This creates or updates an enterprise contact in 360 based on PersonalIdNumber. If the supplied PersonalIdNumber does not exist in 360 the contact is created. If the supplied PersonalIdNumber exists in 360 the contact is updated to mirror the supplied data.

In the scenario where there exists several contacts with the same PersonalIdNumber the oldest (lowest recno) will be updated, while an error message is logged to TraceView and the Windows Event Viewer. It is not possible that this web service method itself creates a duplicate of a contact, given that the PersonalIdNumber is correct.

OBS: If a property is not set, meaning it is blank or null; the corresponding value in 360 will be set to an empty value. This behavior is unique to the contact related synchronization methods.

2.3.1 Input

Data Contract: SynchronizeEnterpriseParameter

Input Property	360° field	Comment
ADContextUser	-	Read the general information section.
Name(*)	Contact – Name	Mandatory when creating the contact
EnterpriseNumber(*)	Contact – Referencenumber	Enterprise number of the company. Represents a unique identifier for the enterprise in 360. Has to be 9 digits long and pass a mod11 validation.
ParentEnterpriseNumber	Contact – ToEmployer	Enterprise number of the parent company.
PhoneNumber	Contact – Teleobject 1	
Email	Contact – Teleobject 6	
PostAddress	Contact – Addresses	Postal address of the company. PostAddress is an address object, see fields needed for input under chapter: “Data Contract: Address” .
OfficeAddress	Contact – Addresses	Office address of the company. OfficeAddress is an address object, see fields needed for input under chapter: “Data Contract: Address” .
DataSource	Log tab	DataSource is presented as a log entry under the Log-tab on the contact. The point is to track which systems have made updates to the contact. No logging is done if this property is empty or not set.
Categories	Contact – Categories	List of categories, identified by their code name in the configured language, or by recno on the format “recno:123”. If this property is not specified (null), no changes will be made to the user’s existing categories.
ContactRelations	Contact - FromMeToOthers	List of contact – contact relations. All relations are hardcoded to relation role “Cooperate with”. Note: Only a private persons or an enterprise may be added as a relation to a Contact.
ReferenceNumber	Contact – FromMeToOthers - Referencenumber	EnterpriseNumber or PersonalIdNumber of the connected contact.

2.3.2 Output

Data Contract: SynchronizeEnterpriseResult

Output Property	360° field	Comment
Recno	Contact - Recno	Unique ID in 360 for the contact
Successful	-	If true, everything is ok. If false, the operation failed.
ErrorMessage	-	If unsuccessful, this contains a short error message
ErrorDetails	-	If unsuccessful, this contains a full stack trace

2.4 GetContactPersons

Retrieves contact persons from 360 based on criteria defined in the input parameter. At least one criteria (Name, Recno or ExternalId) must be given. Returns a list of matching contact persons.

2.4.1 Input

Data Contract: GetContactPersonParameter

Input Property	360° field	Comment
ADContextUser		Read the general information section.
Name()	Contact – SearchName	Name of the contact. Returns all contacts with a name that contains the search name. Example: “Inge” will return “Knut Inge N.” and “Inge Larsen”
ExternalId()	Contact – ExternalID	External identity number (unique identifier) of the contact. Must be an exact match to return a contact.
Recno()	Contact – Recno	Recno (unique identifier in 360) of the contact. Must be an exact match to return a contact.

2.4.2 Output

Data Contract: GetContactPersonResult

Output Property	360° field	Comment
ContactPersons	Contact	List of ContactPersonBase objects. See chapter “ Data Contract: ContactPersonBase ” for details.
Successful	-	Will be true as long as no exception occurred during the execution. This means that

		successful will be true even if no contact persons matching the criteria was found.
ErrorMessage	-	If successful is false, this contains a short error message
ErrorDetails	-	If successful is false, this contains a full stack trace

2.5 GetPrivatePersons

Retrieves private persons from 360 based on criteria defined in the input parameter. At least one criteria (Name, Recno or PersonalIdNumber) must be given. Returns a list of matching private persons.

2.5.1 Input

Data Contract: GetPrivatePersonParameter

Input Property	360° field	Comment
ADContextUser		Read the general information section.
Name()	Contact – SearchName	Name of the contact. Returns all contacts with a name that contains the search name. Example: “Inge” will return “Knut Inge N.” and “Inge Larsen”
PersonalIdNumber()	Contact – ReferenceNumber	Personal identity number (unique identifier) of the contact. Must be an exact match to return a contact.
Recno()	Contact – Recno	Recno (unique identifier in 360) of the contact. Must be an exact match to return a contact.

2.5.2 Output

Data Contract: GetPrivatePersonResult

Output Property	360° field	Comment
PrivatePersons	Contact	List of PrivatePersonBase objects. See chapter “ Data Contract: PrivatePersonBase ” for details.
Successful	-	Will be true as long as no exception occurred during the execution. This means that successful will be true even if no private persons matching the criteria was found.

ErrorMessage	-	If successful is false, this contains a short error message
ErrorDetails	-	If successful is false, this contains a full stack trace

2.6 GetEnterprises

Retrieves enterprises from 360 based on criteria defined in the input parameter. At least one criteria (Name, Recno or EnterpriseNumber) must be given. Returns a list of matching enterprises.

2.6.1 Input

Data Contract: GetEnterpriseParameter

Input Property	360° field	Comment
ADContextUser	-	Read the general information section.
Name(\)	Contact – SearchName	Name of the company. Returns all contacts with a name that contains the search name. Example: “Innovation” will give a hit on “Software Innovation”.
EnterpriseNumber(\)	Contact – ReferenceNumber	Enterprise identity number (unique identifier) of the company. Must be an exact match to return a contact
Recno(\)	Contact – Recno	Recno (unique identifier in 360) of the company. Must be an exact match to return an enterprise

2.6.2 Output

Data Contract: GetEnterpriseResult

Output Property	360° field	Comment
Enterprises	Contact	List of PrivatePersonBase objects. See the chapter “ Data Contract: EnterpriseResult ” for details.
Successful	-	Will be true as long as no exception occurred during the execution. This means that successful will be true even if no enterprises matching the criteria was found.
ErrorMessage	-	If successful is false, this contains a short error message
ErrorDetails	-	If successful is false, this contains a full stack trace

2.7 Sub Contracts

This chapter defines data contracts in use by the main data contracts of the Contact methods.

2.7.1 Data Contract: Address

Property	360° field	Comment
StreetAddress	Address - Address	Example: "Drammensveien 1"
ZipCode	Address - ZipCode	Zip code of address.
ZipPlace	Address - ZipPlace	Zip place (City) of address.
Country(*)	Address - Country	Country of address. To create an address the user has to include "Country" together with another field. E.g. just sending in "Country" will not create an address, but sending in Country and StreetAddress will create an address. Example of input formats: "Norway" or "NOR".
County	Address - County	Municipality of address. Norwegian: "Kommune". Examples of input formats: "BODØ" or "1804", it is recommended that the number format is used, i.e. "1804".
Area	Address - Area	County of address. Norwegian: "Fylke". Examples of input formats: "NORDLAND" or "18", it is recommended that the number format is used, i.e. "18".
State	Address – State	Very optional, only used for some foreign countries.

2.7.2 Data Contract: ContactPersonBase

Property	360° field	Comment
FirstName	Contact – Name2	
LastName	Contact – Name3	Mandatory when creating the contact
ExternalId	Contact – ExternalId	The external identity number of the contact, used as the unique identifier for the contact in 360.
Enterprise	Contact – ToEmployer	The OrgNo of the Enterprise the contact is connected to.
Email	Contact – TeleObject 6	
PhoneNumber	Contact – TeleObject 1	
MobilePhone	Contact – TeleObject 27	
PrivateAddress	Contact - Addresses	PrivateAddress is an address object, see details under the chapter: “Data Contract: Address” .
PostAddress	Contact - Addresses	PostAddress is an address object, see details under the chapter: “Data Contract: Address” .

2.7.3 Data Contract: PrivatePersonBase

Property	360° field	Comment
FirstName	Contact – Name2	
LastName	Contact – Name3	Mandatory when creating the contact
PersonalIdNumber	Contact – Referencenumber	The personal identity number of the contact, used as the unique identifier for the contact in 360.
Email	Contact – TeleObject 6	
PhoneNumber	Contact – TeleObject 1	
MobilePhone	Contact – TeleObject 27	
PrivateAddress	Contact - Addresses	PrivateAddress is an address object, see details under the chapter: “Data Contract: Address” .
PostAddress	Contact - Addresses	PostAddress is an address object, see details under the chapter: “Data Contract: Address” .

2.7.4 Data Contract: EnterpriseResult

Output Property	360° field	Comment
EnterpriseNumber	Contact – Referencenumber	Unique ID in 360 for the contact, also the contact's personal ID number

Name	Contact – Name	
PhoneNumber	Contact – Teleobject 1	
Email	Contact – Teleobject 6	
OfficeAddress	Contact – Addresses	OfficeAddress is an address object, see details under the chapter: “ Data Contract: Address ”
PostAddress	Contact - Addresses	PostAddress is an address object, see details under the chapter: “ Data Contract: Address ”
ContactRelations	Contact – ToContacts	A list of contacts associated with the enterprise. The list contains EnterpriseContactResult objects.
Name	Contact – ToContacts - Name	Name of a contact person

3 User service

Typical URL: [http://\[servername\]:8088/SI.WS.Core/SIF/UserService.svc](http://[servername]:8088/SI.WS.Core/SIF/UserService.svc)

3.1 SynchronizeUser

Creates or updates a user in 360, identified by its login name. If the user does not exist, it is created. If it exists, it is updated to match the input.

A user must be connected to an existing 360 contact person, so always call SynchronizeContactPerson in the contact service first.

3.1.1 Input

Data Contract: SynchronizeUserParameter

Input Property	360° field	Comment
ADContextUser	-	Read the general information section.
Login (*)	User – UserID	Login name
ContactExternalId (*)	User – ToContact	Reference to the user's contact person
IsActive	User – Active	True/false. Default: True
Profiles	User profile	If the parameter is not specified (null), profiles are not synchronized, and no existing profiles will be overwritten. If it is specified, even with an empty list, existing profiles will be overwritten.
Role (*)	User profile – ToSysrole	Code from the Sysroles code table, (using the configured language), or recno on the format "recno:123"
Enterpriseld (*)	User profile – ToOrgUnit	The role's organization unit. Set to either it's external id or enterprise number, or recno on the format "recno:123"
FromDate	User profile – FromDate	A date the profile is valid from
ToDate	User profile – ToDate	A date the profile is valid to
AccessGroups	Access Group Members	If the parameter is not specified (null), access groups are not synchronized, and no existing group memberships will be overwritten. If it is specified, even with an empty list, existing group memberships will be overwritten.
AccessGroup (*)	AGM – ToAccessGroup	Name of the access group, or recno on the format "recno:123"
FromDate	AGM – FromDate	A date the membership is valid from

ToDate	AGM – ToDate	A date the membership is valid to
---------------	--------------	-----------------------------------

The Profiles and AccessGroups properties are optional. If not specified at all (null), no changes will be made to existing profiles and group memberships, allowing these to be maintained manually.

3.1.2 Output

Data Contract: SynchronizeUserResult

Output Property	360° field	Comment
Recno	User – Recno	Internal 360 id for the user
Successful		If true, everything is ok. If false, the operation failed.
ErrorMessage		If unsuccessful, this contains a short error message
ErrorDetails		If unsuccessful, this contains a full stack trace

4 Case service

Typical URL: [http://\[servername\]:8088/SI.WS.Core/SIF/CaseService.svc](http://[servername]:8088/SI.WS.Core/SIF/CaseService.svc)

4.1 CreateCase

Creates a case in 360 with the case type Case (“Sak/Noark Sak” in Norwegian). The parameters in the data contracts accept either code value of the respective code or recno if nothing else is specified.

4.1.1 Input

Data Contract: CreateCaseParameter

Input Property	360° field	Code table	Comment
ADContextUser			Read the general information section.
Title(*)	Case – Description		
UnofficialTitle	Case – UnofficialTitle		If not set, Title will be used
StartDate	Case – StartDate		Start date of the case progress. Defaults to the creation date of the case.
Status	Case – ToCaseStatus	Case status	Code value or recno of the desired case status. Example of use:

		<p>Status="B"; sets the status to "Under handling"</p> <p>ArchiveCode and ArchiveType needs to be supplied to set the status that closes the case. I.e. "A" – "Avsluttet"</p> <p>Defaults according to the ADUserContext user's role:</p> <ul style="list-style-type: none"> - Archivist = Closed - Case Manager = Reserved
ClosedDate	Case – ClosedDate	<p>The date the case is closed. Should be used together with a closing status and is ignored otherwise.</p> <p>The following happens if the Status property is set to a closing status:</p> <ul style="list-style-type: none"> - If this property is set, the case will gain this date as ClosedDate. - If this property is not set 360 automatically sets the ClosedDate to the current date.
JournalUnit	Case – ToJournalUnit	Can be code value or recno. E.g. JournalUnit="Sentralarkiv"
SubArchive	Case – ToSubArchive	Can be code value or recno. E.g. SubArchive="Sakarkiv", "60001" or "recno:60001"
ArchiveCodes	Case – ToClassCodes	At least one ArchiveCode needs to be set to close the case. This is a list of ClassCodeParameter objects. See the chapter " Data Contract ClassCodeParameter " for details.
FiledOnPaper	Case – Paper	Indicating (true/false) if the case is stored on paper.
Keywords	Case – Keywords	List of keywords connected to the case. If a keyword that does not exist in 360 is used, this is added to the keywords code table.
AccessCode	Case – ToAccessCode	Code value or recno. E.g. AccessCode = "U" or "recno:0"

			Some access codes require Paragraph to be specified as well.
Paragraph	Case - Paragraph		Can be code value or recno. E.g. Paragraph = "OFL § 11"
			Mandatory for some access codes.
AccessGroup	Case – ToAccessGroup		Can be code value or recno. E.g. AccessGroup = "Public"
Notes	Case – Notes		Notes on the case
CaseType	Case - CaseType	Case type	Code or recno value, for instance "Sak" or "recno:2". Default: 2 (Noark case)
ResponsiblePersonIdNumber	Case – OurRef		The personal identity number of the responsible person. Looks for a contact with ExternalID first, then for one with reference number equal to the property input.
ResponsiblePersonRecno	Case – OurRef		This field is a fallback field and is only used if ResponsiblePersonIdNumber is null or empty. Must be recno of a valid private person.
ResponsibleEnterpriseNumber	Case – ToOrgUnit		The enterprise number of the responsible enterprise. Looks for a contact with ExternalID first, then for one with reference number equal to the property input. If neither this nor the ResponsibleEnterpriseRecno property is set, the responsible enterprise will be set to the main enterprise of all the users.
ResponsibleEnterpriseRecno	Case – ToOrgUnit		This field is a fallback field and is only used if ResponsibleEnterpriseNumber is null or empty. Must be recno of a valid enterprise.
Contacts	Case - ToCaseContact		List of contacts with associated roles.

ReferenceNumber	CaseContact – ToContact	<p>The identification number of the contact. This is PersonIdNumber for private persons and EnterpriseNumber for enterprises.</p> <p>Use either this field OR the ExternalId-field.</p>
ExternalId	CaseContact – ToContactperson	<p>The identification number of the contact. This is ExternalId for ContactPersons.</p> <p>Use either this field OR the ReferenceNumber-field.</p>
Role	CaseContact – ToRole	<p>The contact's role for the specific case. Can be code or recno. E.g. Role = "Sakspart".</p> <p>The possible roles may be restricted based on the case type.</p>
ReferringCases	Case – ReferringCases	<p>List of case numbers of referring cases. E.g. ("14/0001", "14/0002") Ignored if empty</p>
Project	Case – Project	<p>Project number of the project this case should be connected to.</p>
ScrapCode	Case – ToScrapCode	<p>Can be code value or recno. E.g. ScrapCode = "B"</p>

4.1.2 Output

Data Contract: CaseOperationResult

Output Property	360° field	Comment
Recno	Case - Recno	Unique ID in 360 for the case.
CaseNumber	Case - Name	The case number given by 360.
Successful	-	If true, everything is ok. If false, the operation failed.
ErrorMessage	-	If unsuccessful, this contains a short error message
ErrorDetails	-	If unsuccessful, this contains a full stack trace

4.2 UpdateCase

Updates a case in 360. If the case has a closed status this operation can only be done by an archivist.

If a property is set to null, meaning not provided in the parameter passed to the method, the property is not updated. To clear the field in 360 the property has to be explicitly set to "". Only works for string properties. Archive codes supplied to this method is ignored and no changes to an archive code is possible once the case is created.

4.2.1 Input

Data Contract: UpdateCaseParameter

Input Property	360° field	Comment
ADContextUser	-	Read the general information section.
CaseNumber(*)	Case – Name	The case number. This property will be used to determine which case will be updated. A case cannot have its case number updated.
Title	Case – Description	
UnofficialTitle	Case – UnofficialTitle	
StartDate	Case – StartDate	The start date of the case progress.
Status	Case – ToCaseStatus	May only be set to a closing status if ArchiveCode and ArchiveType are also set or have been set. If the status is set to a closing status (e.g. "Closed") the closed date of the case is set to the current date.
JournalUnit	Case – ToJournalUnit	Can be code value or recno. E.g. JournalUnit="Sentralarkiv"
SubArchive	Case – ToSubArchive	Can be code value or recno. E.g. SubArchive="Sakarkiv"
FiledOnPaper	Case – Paper	Indicating (true/false) if the case is stored on paper.
ArchiveCodes	Case – ToClassCodes	At least one ArchiveCode needs to be set to close the case. This is a list of ClassCodeParameter objects. Codes are appended. Existing codes can not be modified. See the chapter " Data Contract ClassCodeParameter " for details.
Keywords	Case – Keywords	When updating keywords this is the update rules: 1) New keywords are added

		2) Existing keywords are ignored 3) Missing keywords are ignored
AccessCode	Case – ToAccessCode	Can be code value or recno. E.g. AccessCode = “U” If the access code requires a paragraph the field paragraph has to offer a valid paragraph. N.B: If a recno value is used, this recno value <u>can not</u> also exist as a code value.
Paragraph	Case – Paragraph	Can be code value or recno. E.g. Paragraph = “OFL § 11”
AccessGroup	Case – ToAccessGroup	Can be code value or recno. E.g. AccessGroup = “Public”
Notes	Case – Notes	Notes of the case. This is appended to notes already present
ResponsiblePersonIdNumber	Case – OurRef	The personal identity number of the responsible person. Looks for a contact with ExternalID first, then for one with reference number equal to the property input.
ResponsiblePersonRecno	Case – OurRef	This field is a fallback field and is only used if ResponsiblePersonIdNumber is null or empty. Must be recno of a valid private person.
ResponsibleEnterpriseNumber	Case – ToOrgUnit	The enterprise number of the responsible enterprise. Looks for a contact with ExternalID first, then for one with reference number equal to the property input. If neither this nor the ResponsibleEnterpriseRecno property is set, the responsible enterprise will be set to the main enterprise of all the users. Normally it is only possible to change the responsible enterprise to an enterprise in the same enterprise hierarchy as the ADUserContext user.
ResponsibleEnterpriseRecno	Case – ToOrgUnit	This field is a fallback field and is only used if ResponsibleEnterpriseNumber

		is null or empty. Must be recno of a valid enterprise.
Contacts	Case - ToCaseContact	Existing contacts will have their case role updated. New contact relations will be added if the contact do not have a previous case role.
ReferenceNumber	CaseContact - ToContact	The identification number of the contact. This is PersonalldNumber for private persons and EnterpriseNumber for enterprises. Use either this field OR the Externalld-field.
Externalld	CaseContact – ToContactperson	The identification number of the contact. This is Externalld for ContactPersons. Use either this field OR the ReferenceNumber-field.
Role	CaseContact - ToRole	The contact's role for the specific case. Can be code or recno. E.g. Role = "Sakspart". The possible roles may be restricted based on the case type.
ReferringCases	Case – ReferringCases	List of case numbers of referring cases. E.g. ("14/0001", "14/0002") Ignored if empty.
Project	Case – Project	Project number of the project this case should be connected to.
ScrapCode	Case – ToScrapCode	Can be code value or recno. E.g. ScrapCode = "B"

4.2.2 Output

Data Contract: CaseOperationResult

Output Property	360° field	Comment
Recno	Case – Recno	Unique ID in 360 for the contact
CaseNumber	Case – Name	The case number given in 360.
Successful	-	If true, everything is ok. If false, the operation failed.
ErrorMessage	-	If unsuccessful, this contains a short error message

ErrorDetails

-

If unsuccessful, this contains a full
stack trace

4.3 GetCases

Retrieves a list of all cases fulfilling the criteria of all the nonblank properties of the supplied parameter. At least one of the properties (CaseNumber, ContactReferenceNumber, and ArchiveCode) must be given. No partial values are allowed.

4.3.1 Input

Data Contract: GetCasesQuery

Input Property	360° field	Comment
ADContextUser	-	Read the general information section.
MaxReturnedCases	-	The maximum number of cases returned by the search. Default is 100.
CaseNumber()	Case – Name	The case number of the case you wish to retrieve.
Title()	Case – Description	The title, or part of the title, of the case you wish to retrieve.
ContactReferenceNumber()	Case – ContactConnection	Searches case-contact relations for a private person contact or an enterprise contact with the given property value. Value should be either a PersonalIdNumber or an EnterpriseNumber.
ArchiveCode()	Case - ToClassificationCodes	The archive code of the cases you want returned.

4.3.2 Output

Data Contract: GetCasesResult

Output Property	360° field	Comment
Cases	Case	A list of CaseResult objects. See the chapter “Data Contract: CaseResult” for details.
Successful	-	Will be true as long as no exception occurred during the execution. This means that successful will be true even if no private persons matching the criteria was found.
ErrorMessage	-	If unsuccessful, this contains a short error message
ErrorDetails	-	If unsuccessful, this contains a full stack trace

4.4 Sub Contracts

This chapter defines data contracts in use by the main data contracts of the Case methods.

4.4.1 Data Contract: ClassCodeParameter

When adding class codes to cases the code table: Noark classification code is never changed. Either a reference to an existing code table (code/recno) or a free text value is given. Keep in mind that the archive has to be configured to allow class codes to be free text, and not be references to the code table.

Property	360° field	Comment
Sort(*)	ToClassCodes – Sort	This property describes by what order the classification codes are represented in 360. The top code sort=1, secondary sort=2, etc.
ArchiveType(*)	ToClassCodes – ToClassType	The classification type of this particular classification code. Can be code or recno. E.g. ArchiveType="FAGKODE PRINSIPP"
IsManuelText	-	Boolean field. Needs to be true if ArchiveCode is supposed to be interpreted as a code value and false if ArchiveCode is a text field. It is the responsibility of the user to use a ArchiveType that supports arbitrary ArchiveCodes in the instance of this being false.
ArchiveCode	Contact – ToClassCode/Value	The classification code itself. Can be code or recno if using code values or text otherwise. E.g. ArchiveCode = "100" or ArchiveCode = "1234567891"

4.4.2 Data Contract: CaseResult

Property	360° field	Comment
Recno	Document – Recno	Unique id in 360 for the document connected to the case
Title	Case – Description	
CaseNumber	Case – Name	
Date	Case – StartDate	The start date of the case.
Status	Case – ToCaseStatus	
ResponsibleEnterpriseName	Case - ToOrgUnit	The name of the responsible enterprise
ResponsiblePersonName	Case – OurRef	The name of the responsible person
ArchiveCodes	Case – ToClassCodes	A list of all ArchiveType/ArchiveCode pairs on the case.

Documents	Document	A list with the meta data for the documents connected to the case. Returns max 100 documents.
DocumentNumber	Document – DocumentNumber	The number of the document connected to the case
Files	File	A list with the meta data for the files connected to the current (active) version of the document. Returns max 100 files.
Recno	File – Recno	Unique id in 360 for the file connected to the document
Title	File – Comment	The title of the file connected to the document
Format	File – Format	The format of the file connected to the document

5 Document service

Typical URL: [http://\[servername\]:8088/SI.WS.Core/SIF/DocumentService.svc](http://[servername]:8088/SI.WS.Core/SIF/DocumentService.svc)

The fields "document date", "dispatched date" and "journal date" are made available through this service. These are document specific dates that have defined behavior in line with NOARK. That implies the following:

1. Journal date is required to be after both dispatched date and document date.
2. Dispatched date is required to be before journal date but after document date.
3. Document date is required to be before journal date and dispatched date.

5.1 CreateDocument

5.1.1 Input

Data Contract: CreateDocumentParameter

Input Property	360° field	Comment
ADUserContext	-	Read the general information section.
Title(*)	Document - Title	
UnofficialTitle	Document - UnofficialTitle	If not set, Title will be used
DocumentDate	Document - DocumentDate	Defaults to the time the document was received.
Archive	Document – Archive	The Document archive for this document. Can be code or recno. E.g. Archive = "Generelt dokument"

		Defaults to "Saksdokument" (recno: 2) if not supplied.
Category(*)	Document – ToDocumentCategory	Can be code or recno. E.g. Category = "Referat"
Status(*)	Document – ToJournalStatus	Can be code or recno. E.g. Status = "J"
CaseNumber(*)	Document – ToCase	CaseNumber of the case this document belongs to. <i>Not mandatory in all archives.</i>
ResponsiblePersonIdNumber	Document – OurRef - Referencenumber	The personal identity number of the responsible person. Looks for a contact with ExternalID first, then for one with reference number equal to the property input.
ResponsiblePersonRecno	Document – OurRef - Referencenumber	This field is a fallback field and is only used if ResponsiblePersonIdNumber is null or empty. Must be recno of a valid private person.
ResponsibleEnterpriseNumber	Document – ToOrgUnit	The enterprise number of the responsible enterprise. Looks for a contact with ExternalID first, then for one with reference number equal to the property input. If neither this nor the ResponsibleEnterpriseRecno property is set, the responsible enterprise will be set to the main enterprise of all the users. Normally it is only possible to change the responsible enterprise to an enterprise in the same enterprise hierarchy as the ADUserContext user.
ResponsibleEnterpriseRecno	Document – ToOrgUnit	This field is a fallback field and is only used if ResponsibleEnterpriseNumber is null or empty. Must be recno of a valid enterprise.
Contacts	Document – ToActivityContact	List of contacts with associated roles.

ReferenceNumber	ActivityContact – ToContact	<p>The identification number of the contact. This is PersonalIdNumber for private persons and EnterpriseNumber for enterprises.</p> <p>Use either this field OR the ExternalId-field.</p>
ExternalId	ActivityContact – ToContactperson	<p>The identification number of the contact. This is ExternalId for ContactPersons.</p> <p>Use either this field OR the ReferenceNumber-field.</p>
Role	ActivityContact – ToRole	<p>The contact's role for the specific document. Can be code or recno. E.g. Role = "Contact".</p> <p>The possible roles may be restricted based on the case type.</p>
SendersReference	Document - ReferenceNo	External reference number.
Access code	Document – Access code	<p>Can be code value or recno. E.g. AccessCode = "U" or "recno:0"</p> <p>Some access code require Paragraph to be specified as well.</p>
Paragraph	Document - ToAuthorization	<p>Can be code or recno E.g. Paragraph = "OFL § 11"</p> <p>Defaults to the same paragraph as the case.</p>
AccessGroup	Case – ToAccessGroup	<p>Can be code value or recno. E.g. AccessGroup = "Public"</p>
JournalDate	Document - JournalDate	<p>The date the document is closed. Should be used together with a closing status and is ignored otherwise.</p> <p>The following happens if the Status property is set to a closing status:</p> <ul style="list-style-type: none"> - If this property is set, the document will gain this date as JournalDate <p>If this property is not set 360 automatically sets the JournalDate to the current date.</p>
DispatchedDate	Document - DispatchedDate	

FiledOnPaper	Document – Paper	
Keywords	Document – Keywords	Adds the given keywords to the case. If they are not found in 360 they are added to the Keyword code table.
Notes	Document – Notes	
ReferringCases	-	List of referring cases, referred by case number. The document-case relation is hardcoded to “reference”. Ignored if empty
ReferringDocuments	-	List of referring documents, referred by document number. Ignored if empty
Files	-	List of files. See the chapter “ Data Contract: File ” for details

5.1.2 Output

Data Contract: DocumentOperationResult

Output Property	360° field	Comment
Recno	Document – Recno	Unique ID in 360 for the document.
DocumentNumber	Document - DocumentNumber	The document number given by 360.
Successful	-	If true, everything is ok. If false, the operation failed.
ErrorMessage	-	If unsuccessful, this contains a short error message
ErrorDetails	-	If unsuccessful, this contains a full stack trace

5.2 UpdateDocument

Updates a document in 360. If a property is set to null, meaning not provided in the parameter passed to the method, the property is not updated. To be able to clear the field in 360 the property has to be set to “”. Only works for string properties.

What is possible and not possible with this method:

- It is **not** possible to change or delete files.
- It is **not** possible to add a new version of files.
- It is possible to add files to a document by supplying the Files and DocumentNumber property.

5.2.1 Input

Data Contract: UpdateDocumentParameter

Property	360° field	Comment
ADUserContext	-	Read the general information section.
DocumentNumber(*)	Document – DocumentNumber	Search field. This document will be updated.
DocumentDate	Document - DocumentDate	
Title	Document – Title	
UnofficialTitle	Document – UnofficialTitle	
Category	Document – ToDocumentCategory	Can be code or recno. E.g. Category = "J"
ResponsiblePersonIdNumber	Document – OurRef – Referencenumber	The personal identity number of the responsible person. Looks for a contact with ExternalID first, then for one with reference number equal to the property input.
ResponsiblePersonRecno	Document – OurRef – Referencenumber	This field is a fallback field and is only used if ResponsiblePersonIdNumber is null or empty. Must be recno of a valid private person.
ResponsibleEnterpriseNumber	Document – ToOrgUnit	The enterprise number of the responsible enterprise. Looks for a contact with ExternalID first, then for one with reference number equal to the property input. If neither this nor the ResponsibleEnterpriseRecno property is set, the responsible enterprise will be set to the main enterprise of all the users. Normally it is only possible to change the responsible enterprise to an enterprise in

		the same enterprise hierarchy as the ADUserContext user.
ResponsibleEnterpriseRecno	Document – ToOrgUnit	This field is a fallback field and is only used if ResponsibleEnterpriseNumber is null or empty. Must be recno of a valid enterprise.
Contacts	Document - ToActivityContact	Existing contacts will have their document role updated. New contact relations will be added if the contact do not have a previous document role.
ReferenceNumber	ActivityContact – ToContact	<p>The identification number of the contact. This is PersonalIdNumber for private persons and EnterpriseNumber for enterprises.</p> <p>Use either this field OR the ExternalId-field.</p>
ExternalId	ActivityContact – ToContactperson	<p>The identification number of the contact. This is ExternalId for ContactPersons.</p> <p>Use either this field OR the ReferenceNumber-field.</p>
Role	ActivityContact - ToRole	<p>The contact's role for the specific document. Can be code or recno.</p> <p>E.g. Role = "Mottaker".</p>
SendersReference	Document - ReferenceNo	External reference number.
Access code	Document – Access code	<p>Can be code value or recno. E.g. AccessCode = "U"</p> <p>If the access code requires a paragraph the field paragraph has to offer a valid paragraph.</p> <p>N.B: If a recno value is used, this recno value <u>can not</u> also exist as a code value.</p>
Paragraph	Document - ToAuthorization	<p>Can be code or recno</p> <p>E.g. Paragraph = "OFL § 11"</p>

		Ignored if used with an access code that doesn't require a paragraph.
AccessGroup	Case – ToAccessGroup	Can be code value or recno. E.g. AccessGroup = "Public"
JournalDate	Document- JournalDate	<p>The date the document is closed. Should be used together with a closing status and is ignored otherwise.</p> <p>The following happens if the Status property is set to a closing status:</p> <ul style="list-style-type: none"> - If this property is set, the document will gain this date as JournalDate <p>If this property is not set 360 automatically sets the JournalDate to the current date</p>
DispatchedDate	Document - DispatchedDate	
FiledOnPaper	Document – Paper	
Keywords	Document – Keywords	<p>Updating the keywords on the case by these rules:</p> <ol style="list-style-type: none"> 1) New keywords are added 2) Existing keywords are ignored 3) Missing keywords are ignored
Notes	Document – Notes	This is appended to notes already present
ReferringCases	Document – ToCase	List of referring cases, referred by case number. Ignored if empty.
ReferringDocuments	?	List of referring documents, referred by document number. Ignored if empty.
Files	?	List of files that will be added to the document. It is not possible to update previously added files. See the chapter "Data Contract: File" for details.

5.2.2 Output

Data Contract: DocumentOperationResult

Property	360° field	Comment
Recno	Document – Recno	Unique ID in 360 for the document.
DocumentNumber	Document - DocumentNumber	The document number in 360.
Successful	-	If true, everything is ok. If false, the operation failed.
ErrorMessage	-	If unsuccessful, this contains a short error message
ErrorDetails	-	If unsuccessful, this contains a full stack trace

5.3 GetDocuments

Take note that this method in addition to getting meta data on documents; may also be used to get file meta data and the physical file(s) inside the document.

5.3.1 Input

Data Contract: GetDocumentsQuery

Input Property	360° field	Comment
ADContextUser		Read the general information section.
MaxReturnedDocuments	-	Maximum number of documents to return. Default 100.
IncludeFileData	-	If the file list should include an output property that contain the files base 64 encoded data. Defaults to false.
CaseNumber(\)	Case – Name	Searches using case number.
DocumentNumber(\)	Document – DocumentNumber	Searches using document number.
Title(\)	Document – Title	Searches using document title.
ContactReferenceNumber(\)	Document – ToActivityContact – ToContact - Referencenumber	Find documents with a relation to this contact. Value should be either the PersonalIdNumber of a private person or the EnterpriseNumber of an enterprise contact.
ContactExternalId	Document – ToActivityContact – ToContact - ExternalID	Find documents with a relation to this contact, identifies by its ExternalId.

5.3.2 Output

Data Contract: GetDocumentsResult

Output Property	360° field	Comment
Documents	-	List of DocumentResult objects. See the chapter “Data Contract: DocumentResult” for details.
Successful	-	Will be true as long as no exception occurred during the execution. This means that successful will be true even if no private persons matching the criteria was found.
ErrorMessage	-	If unsuccessful, this contains a short error message

ErrorDetails	-	If unsuccessful, this contains a full stack trace
---------------------	---	---------------------------------------------------

5.4 Sub Contracts

This chapter defines data contracts in use by the main data contracts of the Document methods.

5.4.1 Data Contract: DocumentResult

Output Property	360° field	Comment
Recno	Document – Recno	
DocumentNumber	Document – DocumentNumber	
CaseNumber	Document – ToCase	
Title	Document – Title	
DocumentDate	Document – DocumentDate	
JournalDate	Document – JournalDate	
StatusCode	Document – ToJournalStatus – Code	Examples: “J”, “R”
StatusDescription	Document – ToJournalStatus – Description	Examples: «Journalført og/eller kontrollert av arkivet», «Reservert av/for saksbehandler»
ResponsibleEnterpriseName	Document – ToOrgUnit	Name on the responsible enterprise.
ResponsiblePersonName	Document – OurRef	Name on the responsible person.
Contacts	Document	List of DocumentContactResult objects. See the chapter “ Data Contract: DocumentContactResult ” for details.
Files		List of files from the current (active) version of the document.
Recno	File - Recno	
Title	File – Comment	
Format	File – Format	
Base64Data		The file data in a base 64 encoded string. Will only be

		returned if the input property IncludeFileData is set to true.
URL	-	A 360 URL that can be used by authenticated users to view the file.
Note	File - Note	
Paragraph	File – ToAuthorization	
RelationTypeDescription	File – ToRelationType - Description	
RelationTypeCode	File – ToRelationType - Code	
VersionFormatDescription	File – ToVersionFormat - Description	
VersionFormatCode	File – ToVersionFormat - Code	
Type	File – Type	
ModifiedBy	File – ToFileModifiedBy – ToContact	
CheckedOutBy	File – ToCheckedOutBy – ToContact	
CategoryCode	File – ToFileCategory - Code	
CategoryDescription	File – ToFileCategory - Description	
StatusDescription	File – ToFileStatus - Description	
StatusCode	File – ToFileStatus - Code	
AccessCodeDescription	File – ToAccessCode - Code	
AccessCodeCode	File – ToAccessCode - Description	
DegradeCode	File – ToDegradeCode	
DegradeDate	File - DegradeDate	
FiledOnPaper	File – Paper	
PaperLocation	File – PaperLocation	
SignDate	File – ToFileVersion – ToContact	

5.4.2 Data Contract: DocumentContactResult

Output Property	360° field	Comment
ReferenceNumber	Contact - ReferenceNumber	Reference number of the contact – if private person or enterprise.
ExternalId	Contact – ExternalID	External id of the contact – if contact person.

Role	ActivityContact – ToRole	Role of the contact on this document in code value.
SearchName	Contact - SearchName	The name of this contact.

5.4.3 Data Contract: File

There are two ways to upload a file to 360, either as Bas64-encoded data, or by first calling the [FileService.Upload\(\)](#) method, and then providing the file reference it returns in UploadedFileReference.

Property	360° field	Comment
Title(*)	File - FileName	
Format(*)	File - FileFormat	Can be code or recno. E.g. Format = "pdf"
Base64Data()		
UploadedFileReference()	-	This value is returned by FileService.Upload
RelationType	File - ToRelationType	Can be code or recno. E.g. RelationType = "H"
VersionFormat	File - ToVersionFormat	Can be code or recno. E.g. VersionFormat = "A"
Note	File – Note	
Category	File – ToFileCategory	Can be code or recno. E.g. Category = "Brev"
Status	File – ToFileStatus	Can be code or recno. E.g. Status = "B"
AccessCode	File – ToAccessCode	
DegradeCode	File – ToDegradeCode	The "Downgrade" values will only show up in the GUI if both properties DegradeCode and DegradeDate is set. Norwegian: "Avgradering".
DegradeDate	File – DegradeDate	The "Downgrade" field will only show up in the GUI if both properties DegradeCode and DegradeDate is set.
FiledOnPaper	File – Paper	
PaperLocation	File – PaperLocation	

6 File service

Typical URL: [http://\[servername\]:8088/SI.WS.Core/SIF/FileService.svc](http://[servername]:8088/SI.WS.Core/SIF/FileService.svc)

6.1 Upload

Streams a file to the 360° server, where it is stored temporarily, and should be referred to in CreateDocument() or UpdateDocument().

6.1.1 Input

Input Property	Comment
FileStream	The file data
User	The uploaded file is stored in this users 360 cache. This user <i>must</i> be the same as the user you authenticate (ADUserContext) with when referring to the file, otherwise it will not be able to locate the it.

6.1.2 Output

Data Contract: UploadFileResult

Output Property	Comment
Successful	If true, the file was uploaded.
FileReference	Use this value to refer to the uploaded file.
ErrorMessage	If unsuccessful, this contains a short error message
ErrorDetails	If unsuccessful, this contains a full stack trace

7 Support methods

7.1 Ping

Each web service endpoint has a Ping() method that can be used to test that you have access to the service, that it is correctly installed, and that your user has at least read permissions to 360°. If it returns without throwing an exception, everything is ok.

8 Dependencies to 360

8.1 PDF Converting

Converting of files to PDF (PDF/A) is not done by the web service. This should be done using the 360 Format Converter module which uses a third party application for the conversion. How the conversion is done is best explained with an example:

1. A web service client calls the CreateDocument() or UpdateDocument() with files.
2. The web service creates/updates a document and adds files to it in 360.
3. Part of the 360 Format Converter module is a timer job that copies files to a third party programs TODO folder. Only files that fit preconfigured properties are copied to this folder, a typical property is that the file's document status is "Closed".
4. The third party application converts the files in the TODO folder. The files are then uploaded to 360 to the correct document.
5. The result is a 360 document with the original file(s) and PDF file(s).

The web service has in other words no responsibility of converting files to PDF. The third party application is PixEdit and supports a wide range of file formats for conversion.

9 Web Service client responsibility

A web service client, for example external systems is responsible for the following:

- Creating cases, documents and contacts in the correct order. This means that the web service methods that are available has to be called in the correct logical order. For example you cannot create a document without first creating the case that it belong to.
- Files has to be in the correct format. PDF files is not verified to be PDF/A. No file format validation is done by the web service
- A general rule, it is client responsibility to validate and clean input to its own needs. Some validation is still done by the 360 business logic, but errors will be a stack trace.
 - o If customer specific validation has to be implemented in the webservice this can be done in an customer specific plugin.