

ACTA

Servizi ACARIS

Manuale Integrazione Fruitore

Versione 3.1

STATO DELLE VARIAZIONI

VERS.	DATA VALIDITÀ	PARAGRAFO O PAGINA	DESCRIZIONE DELLA VARIAZIONE
3.1	23/01/2024	Cap. 7.2	Correzione Tabella 4
		Cap. 11.1	Correzione Tabella 43
		Cap. 17	Correzione Tabella 86
3	16/01/2023	Cap. 1.3	Aggiunto riferimento a documento che descrive i servizi legati all'invio della segnatura di una registrazione di protocollo
		Cap. 2.10	Aggiunto scenario relativo al recupero dell'ora della registrazione di protocollo
		Cap. 2.11	Aggiunto scenario sull'invio della segnatura di protocollo e scenari secondari
		Cap. 12.2	Nuovi oggetti su cui effettuare la ricerca
		Cap. 12.4, 12.5 e 12.6	Descritti i nuovi servizi legati all'invio della segnatura di una registrazione di protocollo
		Cap. 7.3	Corretto nome dati della tabella di output (tabella 7)
2.3	13/12/2021	Cap. 2.9	Aggiunto scenario che descrive le implicazioni della verifica asincrona della firma
		Cap. 12.1	Aggiornato tabella 52 InfoCreazioneCorrispondente (per nuove linee guida AgID). Aggiunti ulteriori dati per distinguere le varie tipologie di corrispondenti (PA, AE, PF e PG) e dati di maggior dettaglio per la PA
2.2	22/04/2021	Cap. 2.1	Deprecato e sostituito dallo scenario 2.8 (per nuove linee guida AgID)
		Cap. 2.8	Aggiunto scenario d'uso in sostituzione del 2.1 (per nuove linee guida AgID)
		Cap. 12.1	Aggiunto nota sui "corrispondenti" (per nuove linee guida AgID)
2.1	26/06/2020	BackofficeService ObjectService DocumentService NavigationService OfficialBookService MultifilingService	Completamento servizi esposti
2	08/06/2020	Cap.3,4,5	Inseriti capitoli relative alle specifiche tecniche
1	08/06/2020	Versione iniziale	Stesura del documento

Sommario

1	Introduzione	7
1.1	Scopo del documento	7
1.2	Prerequisito	7
1.3	Riferimenti	7
2	Principali scenari d'uso dei servizi	9
2.1	Creazione di una Registrazione di protocollo "in arrivo" ed archiviazione in una specifica struttura aggregativa	9
2.2	Ricerca classificazioni allegate di una classificazione principale	9
2.3	Inserimento di un documento in una precisa struttura aggregativa	10
2.4	Inserimento di un documento "allegato"	10
2.5	Recupero del content-stream di un Document a partire da una classificazione	10
2.6	Creazione di una Registrazione di protocollo "in partenza"	11
2.7	Ricerca protocolli per intervallo date	11
2.8	Creazione di una Registrazione di protocollo "in arrivo"	11
2.9	Gestione della verifica asincrona della firma	11
2.10	Recupero dell'ora della registrazione di protocollo	12
2.11	Invio della segnatura di una registrazione di protocollo e recupero dei dati scambiati	12
2.11.1	Ottenere le informazioni di dettaglio su una registrazione	13
2.11.2	Ottenere lo stato di apposizione del sigillo	14
2.11.3	Ottenere lo stato di invio del messaggio	15
2.11.4	Ottenere i dati dei corrispondenti di una registrazione	16
3	Specifiche tecniche di integrazione	18
3.1	Object Model	18
3.1.1	Modelli per l'utilizzo di "properties"	18
3.2	Definizione interfaccia	19
3.2.1	Approccio "Contract First"	19
3.2.2	Messaggi	19
4	Riferimenti Tecnologici	20
4.1	Specifiche	20
4.2	Framework	20
4.3	Librerie	20
4.4	Libreria di implementazione dell'Object Model	20
4.5	Interfaccia Wrapper Web Services	20
4.6	AcarisException	20
4.7	Eccezioni e gestione Fault in ACARIS	21
4.8	Gestione allegati	21
4.9	DTO	21
5	Il catalogo dei servizi e delle operazioni	22
5.1	Modalità di esposizione dei servizi ACARIS	22
5.2	Fruizione internet APIStore	22
6	RepositoryService	23
6.1	L'operazione getRepositories	23

7	BackOfficeService	24
7.1	L'operazione getPrincipalExt	24
7.2	L'operazione getQueryableObjects	25
7.3	L'operazione getQueryableObjectMetadata	25
7.4	L'operazione query	26
8	ObjectService	28
8.1	L'operazione query	28
8.2	L'operazione createDocument	28
8.3	L'operazione getProperties	29
8.4	L'operazione updateProperties	29
8.5	L'operazione moveDocument	30
8.6	L'operazione createFolder	31
8.7	L'operazione getContentStream	32
8.8	L'operazione createAssociativeDocument	33
8.9	L'operazione createRelationship	33
9	DocumentService	35
9.1	L'operazione creaDocumento	35
9.2	L'operazione trasformaDocumentoPlaceholderInDocumentoElettronico	36
9.3	L'operazione getRappresentazioneDocumento	37
9.4	Oggetti comuni	38
10	NavigationService	40
10.1	L'operazione getChildren	40
10.2	L'operazione getFolderParent	40
10.3	L'operazione getObjectParent	41
11	RelationshipService	43
11.1	L'operazione getObjectRelationship	43
12	OfficialBookService	44
12.1	L'operazione creaRegistrazione	44
12.2	L'operazione query	50
12.3	L'operazione getProperties	50
12.4	L'operazione inviaSegnaturaRegistrazione	51
12.5	L'operazione ricercaDatiInterscambio	53
12.6	L'operazione annullaRegistrazione	56
13	MultifilingServices	57
13.1	L'operazione addAssociativeObjectToFolder	57
13.2	L'operazione aggiungiClassificazione	58
14	ManagementService	60
14.1	L'operazione getVitalRecordCode	60
14.2	L'operazione addAnnotazioni	60
15	SubjectRegistryService	62
15.1	L'operazione creaSoggetto	62
15.2	L'operazione getProperties	62

15.3	L'operazione query	63
15.4	L'operazione ricercaSoggetto	63
15.5	L'operazione ricercaSoggettoDaFonteEsterna	64
15.6	L'operazione getPropertiesMassive	64
16	SMSService	66
16.1	L'operazione query	66
16.2	L'operazione creaSmistamento	66
17	Oggetti comuni a tutti i servizi ACARIS	68

Indice delle tabelle

Tabella 1: dati di output getRepositories	23
Tabella 2: dati di input getPrincipalExt	24
Tabella 3: dati di output getPrincipalExt	24
Tabella 4: dati di input getQueryableObjects	25
Tabella 5: dati di output getQueryableObjects	25
Tabella 6: dati di input getQueryableObjectMetadata	26
Tabella 7: dati di output getQueryableObjectMetadata	26
Tabella 8: dati di input query	26
Tabella 9: dati di output query	27
Tabella 10: dati di input createDocument	28
Tabella 11: dati di output createDocument	29
Tabella 12: dati di input getProperties	29
Tabella 13: dati di output getProperties	29
Tabella 14: dati di input updateProperties	30
Tabella 15: dati di output updateProperties	30
Tabella 16: dati di input moveDocument	30
Tabella 17: dati di output moveDocument	31
Tabella 18: dati di input createFolder	31
Tabella 19: dati di output createFolder	32
Tabella 20: dati di input getContentStream	32
Tabella 21: dati di output getContentStream	33
Tabella 22: dati di input createAssociativeDocument	33
Tabella 23: dati di output createAssociativeDocument	33
Tabella 24: dati di input createRelationship	34
Tabella 25: dati di output createRelationship	34
Tabella 26: dati di input creaDocumento	35
Tabella 27: dati di output creaDocumento	35
Tabella 28: dati di input trasformaDocumentoPlaceholderInDocumentoElettronico	36
Tabella 29: dati di output trasformaDocumentoPlaceholderInDocumentoElettronico	36
Tabella 30: dati di input getRappresentazioneDocumento	37
Tabella 31: dati di output getRappresentazioneDocumento	37
Tabella 32: DocumentoFisicoIRC	38
Tabella 33: ContenutoFisicoIRC	38
Tabella 34: StepErrorAction	39
Tabella 35: FailedStepInfo	39
Tabella 36: RelationshipPropertiesType	39

Tabella 37: dati di input getChildren	40
Tabella 38: dati di output getChildren.....	40
Tabella 39: dati di input getFolderParent.....	41
Tabella 40: dati di output getFolderParent	41
Tabella 41: dati di input getObjectParent	41
Tabella 42: dati di output getObjectParent.....	41
Tabella 43: dati di input getObjectRelationship	43
Tabella 44: dati di output getObjectRelationship	43
Tabella 45: dati di input creaRegistrazione	44
Tabella 46: dati di output creaRegistrazione.....	44
Tabella 47: Protocollazione	45
Tabella 48: ProtocollazioneDocumentoEsistente.....	45
Tabella 49: ProtocollazioneDaSmistamento	45
Tabella 50: RegistrazioneAPI	45
Tabella 51: RegistrazioneArrivo.....	47
Tabella 52: InfoCreazioneCorrispondente.....	47
Tabella 53: dati di input getProperties.....	50
Tabella 54: dati di output getProperties	51
Tabella 55: dati di input inviaSegnaturaRegistrazione	51
Tabella 56: dati di output inviaSegnaturaRegistrazione.....	53
Tabella 57: dati di input ricercaDatiInterscambio	54
Tabella 58: dati di output ricercaDatiInterscambio.....	54
Tabella 59: dati di input annullaRegistrazione	56
Tabella 60: dati di output annullaRegistrazione.....	56
Tabella 61: dati di input addAssociativeObjectToFolder.....	57
Tabella 62: dati di input aggiungiClassificazione	58
Tabella 63: dati di output aggiungiClassificazione	59
Tabella 64: dati di input getVitalRecordCode.....	60
Tabella 65: dati di output getVitalRecordCode	60
Tabella 66: dati di input addAnnotazioni	61
Tabella 67: dati di output addAnnotazioni.....	61
Tabella 68: dati di input creaSoggetto.....	62
Tabella 69: dati di output creaSoggetto	62
Tabella 70: dati di output getProperties	62
Tabella 71: dati di output getProperties	63
Tabella 72: dati di input ricercaSoggetto.....	63
Tabella 73: dati di output ricercaSoggetto	64
Tabella 74: dati di input ricercaSoggettoDaFonteEsterna.....	64
Tabella 75: dati di output ricercaSoggettoDaFonteEsterna	64
Tabella 76: dati di input getPropertiesMassive	65
Tabella 77: dati di output getPropertiesMassive	65
Tabella 78: dati di input creaSmistamento.....	66
Tabella 79: dati di output creaSmistamento	67
Tabella 80: DocumentoFisicoPropertiesType.....	68
Tabella 81: ContenutoFisicoPropertiesType.....	68
Tabella 82: AcarisContentStreamType	68
Tabella 83: QueryConditionType	68
Tabella 84: NavigationConditionInfoType	69
Tabella 85: PropertyFilterType	69
Tabella 86: PagingResponseType	69
Tabella 87: QueryNameType	69

Tabella 88: ObjectResponseType	70
Tabella 89: PropertyType	70
Tabella 90: SimpleResponseType	70

1 Introduzione

I servizi “ACTA Archive Interoperability Services” (in seguito ACARIS) permettono a tutti gli applicativi verticali che producono e/o gestiscono documenti di effettuare l’integrazione con ACTA, sistema di gestione documentale di archivio e protocollo del CSI Piemonte.

1.1 Scopo del documento

Il documento è indirizzato ai fruitori dei servizi applicativi del sistema documentale ACTA. Lo scopo è quello di fornire indicazioni per una integrazione rapida tra un sistema alimentante e il sistema documentale e di essere d’aiuto in fase di stima di progetto.

1.2 Prerequisito

Prerequisito essenziale e obbligatorio per l’integrazione con ACARIS è aver contattato il responsabile referente del servizio dell’ente di riferimento.

Dovrà quindi essere redatta una analisi archivistica in cui si delinea l’esigenza dell’applicativo integrante e si definisce la logica di integrazione.

Per avviare la fase di sviluppo dell’integrazione, è necessario contattare il supporto tecnico ACARIS tramite invio di un messaggio di posta all’indirizzo e-mail supporto_acaris@csi.it.

1.3 Riferimenti

Si riportano di seguito i documenti relativi alle specifiche di dettaglio, disponibili in aggiunta al presente manuale ottenibili contattando supporto_acaris@csi.it.

Numero	Nome file	Descrizione
[1]	PGED-ACTA-STE011-AcArls-V04-BackOfficeServices	Descrizione dei servizi esposti da BackOffice service
[2]	PGED-ACTA-STE011-AcArls-V09_ObjectServices	Descrizione dei servizi esposti da Object service
[3]	PGED-ACTA-STE016-AcArls-V10-DocumentServices	Descrizione dei servizi esposti da Document service
[4]	PGED-ACTA-STE016-AcArls-V02-RelationshipServices	Descrizione dei servizi esposti da Relationship service
[5]	PGED-ACTA-STE015-AcArls-V06-NavigationServices	Descrizione dei servizi esposti da Navigation service
[6]	ACARISWS-BackOfficeService.wsdl, ACARISWS-DocumentService.wsdl, ACARISWS-ManagementService.wsdl, ACARISWS-MultifilingService.wsdl, ACARISWS-NavigationService.wsdl, ACARISWS-ObjectService.wsdl, ACARISWS- OfficialBookService.wsdl, ACARISWS- RelationshipsService.wsdl, ACARISWS- RepositoryService.wsdl, ACARISWS- SMSService.wsdl, ACARISWS- SubjectRegistryService.wsdl	Operazioni dei servizi ACARIS
	ACARIS-ArchiveMessaging.xsd, ACARIS-BackOfficeMessaging.xsd, ACARIS-DocumentMessaging.xsd, ACARIS-ManagementMessaging.xsd, ACARIS-OfficialBookMessaging.xsd, ACARIS-SMSMessaging.xsd, ACARIS-SubjectRegistryMessaging.xsd	

[7]	ACARIS-Common.xsd ACARIS-Common-PRT.xsd ACARIS-Archive.xsd ACARIS-BackOffice.xsd ACARIS-Document.xsd ACARIS-Management.xsd ACARIS-OfficialBook.xsd ACARIS-SMS.xsd ACARIS-SubjectRegistry.xsd	Descrizione dei tipi di dati utilizzati
[8]	Apache CXF 2.2.3	
[9]	ACARIS-binding-archive.xjb ACARIS-binding-backoffice.xjb ACARIS-binding-document.xjb ACARIS-binding-management.xjb ACARIS-binding-officalbook.xjb ACARIS-binding-sms.xjb ACARIS-binding-subjectregistry.xjb	
[10]	PGED-ACTA-STE017-AcArls-OfficialBookServices_V3.0	Descrizione dei servizi esposti da OfficialBook service.
[11]	ACTA-ACARIS_DescrizioneServiziPacchettoInvioSegnatura_V1.0	Descrizione dei servizi legati all'invio della segnatura di una registrazione di protocollo

2 Principali scenari d'uso dei servizi

Di seguito sono illustrati, in modo schematico, alcuni degli scenari tipici di interazione di una applicazione client attraverso i servizi ACARIS. Questa non vuole essere una illustrazione esaustiva delle modalità in cui possono essere utilizzati ed orchestrati le operazioni esposte ma offrire una indicazione sul flusso logico degli scenari di utilizzo più comuni.

2.1 Creazione di una Registrazione di protocollo “in arrivo” ed archiviazione in una specifica struttura aggregativa

Questo scenario di utilizzo è DEPRECATO!

Il motivo è dovuto al fatto che a partire dal mese di Luglio 2021 Doqui ACTA verrà modificato per aderire alle linee guida AgID: la segnatura verrà fissata al momento della creaRegistrazione ed i dati non potranno più essere modificati.

Quindi non sarà più possibile fare questa sequenza di operazioni:

1. *creaRegistrazione* con una serie di documenti cartacei
2. staccare il numero (di protocollo)
3. aggiungere i file in un secondo momento

In sostituzione fare riferimento allo scenario Creazione di una Registrazione di protocollo “in arrivo” (vd 2.8).

- Invocazione del servizio *creaRegistrazione* (vd 12.1) indicando la volontà di creare un oggetto di tipo “in arrivo” senza l’indicazione di un documento esistente in archivio.
- Il sistema, in base ai parametri con cui è invocato il servizio *creaRegistrazione*, crea una registrazione di protocollo, una classificazione principale ed eventualmente degli allegati “place holder” (a cui non è collegato un contenuto).
- Aggiornare i metadati della classificazione principale creata utilizzando l’operazione *updateProperties* (vd 8.4) di Object service, utilizzando l’identificativo della classificazione restituito da *creaRegistrazione*. Per modificare i metadati del documento archivistico, recuperare l’identificativo (objectId) del documento utilizzando l’operazione *getChildren* (vd 10.1) a partire dall’identificatore della classificazione.
- Se si vuole associare alla classificazione principale un contenuto, invocazione del servizio *trasformaDocumentoPlaceholderInDocumentoElettronico* (vd 9.2) utilizzando come parametro di ingresso l’identificativo della classificazione principale ed il contenuto da associare.
- Se sono stati create delle classificazioni allegate, ricerca degli identificativi delle classificazioni allegate (vd 2.2) e, per ognuno di questi, effettuare i passi 3 e 4.
- Per ottenere l’identificativo della struttura aggregativa in cui archiviare il documento, effettuare la ricerca tramite il servizio *query* (vd 8.1) di Object service, per oggetti dello specifico tipo di Folder (per esempio, Serie tipologica di documenti, Fascicolo Reale Annuale)
- Con le informazioni del Folder (aggregazione) target, utilizzo dell’identificativo univoco (Object ID) del Folder come parametro di ingresso del servizio *moveDocument* (vd 8.5) di Object service, in modo da spostare la classificazione principale nell’aggregazione ricercata.

2.2 Ricerca classificazioni allegate di una classificazione principale

1. Invocazione del servizio *getChildren* (vd 10.1) di Navigation service utilizzando come parametro di ingresso l’identificativo della classificazione principale: si otterrà l’identificativo di un oggetto di tipo Gruppo Allegati.
2. Invocazione del servizio *getChildren* utilizzando come parametro di ingresso l’identificativo del gruppo allegati per ottenere la lista delle classificazioni allegate.

	<p style="text-align: center;">DoQui – ACTA Servizi ACARIS Manuale Integrazione Fruitore</p>	<p style="text-align: right;">ACTA-Acaris-MUT-002</p> <p style="text-align: right;">Pag. 10 di 70</p>
--	---	---

2.3 Inserimento di un documento in una precisa struttura aggregativa

1. Ricerca tramite il servizio *query* (vd 8.1) di Object Services, per oggetti dello specifico tipo di Folder (per esempio, Serie tipologica di documenti, Fascicolo Reale Annuale)
 - a. In caso di esito negativo, invocazione del servizio *createFolder* (vd 8.6) di Object Services per l'inserimento nel repository di un Folder del tipo e ottenimento delle sue informazioni.
2. Con le informazioni del Folder target, utilizzo dell'identificativo univoco (Object ID) del Folder come parametro di ingresso del servizio *creaDocumento* (vd 9.1) di Document Services.

Il servizio *creaDocumento*, in modo “atomico”, orchestra lo stesso flusso logico delle invocazioni di *createAssociativeDocument* (vd 8.8) e *createRelationships* (vd 8.9) di Object Service: si crea così un oggetto Folder (della sottoclasse Classificazione) ed i necessari oggetti sottoclasse di Document (Documento archivistico, Contenuto Fisico) per caricare un content stream. Il servizio mette in relazione la classificazione con la struttura aggregativa ricercata.


2.4 Inserimento di un documento “allegato”

1. Ricerca tramite il servizio *query* (vd 8.1) di Object Services del Folder a cui è collegato il documento principale a cui si vuole allegare un documento, la ricerca è effettuata per specifico tipo di Folder (per esempio Serie tipologica di documenti, Fascicolo Reale Annuale).
2. Ricerca tramite il servizio *query* di Object Service, per oggetto della specifica sottoclasse di Folder (per esempio, Documento Semplice) che rappresenta il documento principale, al fine di ottenere il suo identificativo.
 - a. Per limitare la ricerca del documento, indicare come parametro di ricerca l'identificativo del Folder (struttura aggregativa) con cui il documento principale è in relazione padre-figlio
3. Utilizzando l'identificativo (Object ID) del documento semplice risalire all'oggetto classificazione a cui è collegato tramite il servizio *getObjectRelationships* (vd 11.1) di Relationship Services. La classificazione è un oggetto derivato da Folder che indica la relazione tra un documento archivistico (sottoclasse di Document) ed una struttura aggregativa.
4. A partire dall'oggetto classificazione utilizzare il suo identificativo come parametro di ingresso del servizio *creaDocumento* (vd 9.1) di Document Services.

Il servizio *creaDocumento*, in questo scenario, crea ed inserisce un oggetto Folder (della sottoclasse Classificazione) ed i necessari oggetti sottoclasse di Document (Documento archivistico, Contenuto Fisico) e crea una relazione con la classificazione principale. Nel caso specifico, l'oggetto di tipo Folder, che modella la relazione padre-figlio tra una classificazione principale ed una classificazione allegata, appartiene alla sottoclasse Gruppo Allegati ed è specifico per modellare la relazione.

2.5 Recupero del content-stream di un Document a partire da una classificazione

1. A partire dall'identificativo di un oggetto Folder (sottoclasse Classificazione), invocazione del servizio *getChildren* (vd 10.1) di Navigation Services per ottenere l'identificativo del Document (sottoclasse Documento archivistico)
2. Invocazione del servizio *getObjectRelationships* (vd 11.1) di Relationships Services per ottenere l'identificativo del Folder (sottoclasse Documento Fisico)
3. Per ottenere il content stream ricercato, utilizzare il servizio *getContentStream* (vd 8.7) di Object Service utilizzando come parametro di ingresso l'identificativo di tipo Documento Fisico.

	<p style="text-align: center;">DoQui – ACTA Servizi ACARIS Manuale Integrazione Fruitore</p>	<p style="text-align: right;">ACTA-Acaris-MUT-002</p> <p style="text-align: right;">Pag. 11 di 70</p>
--	---	---

2.6 Creazione di una Registrazione di protocollo “in partenza”

1. Nel caso in cui il documento a cui si vuole associare una registrazione di protocollo non sia presente nel repository, effettuare l’inserimento del documento nella struttura aggregativa desiderata (vd 2.3)
 - Eventualmente, inserimento dei suoi documenti allegati (vd 2.4)
2. Invocazione del servizio *creaRegistrazione* (vd 12.1) di Official Book Services indicando la volontà di creare un oggetto di tipo “in partenza” a partire da documenti già esistenti in repository (utilizzando la modalità “Protocollazione da documenti esistenti”) e passando, tra i parametri di ingresso, l’identificativo univoco della classificazione principale (ottenuto come risultato dell’invocazione del servizio *creaDocumento* o tramite ricerca).

2.7 Ricerca protocolli per intervallo date

Per leggere le registrazioni di protocollo presenti nel sistema ed i dati collegati è necessario orchestrare diversi servizi.

1. Nel caso della ricerca per date di protocollo si utilizza il servizio *query* (vd 12.2) di Official Book Services.
2. Successivamente per risalire ai dati delle classificazioni legate ad una registrazione di protocollo e navigare le relazioni tra classificazione principale ed eventuali allegati, si devono orchestrare invocazioni dei servizi *query* di Official Book Services e di Object Service, *getChildren* (vd 10.1) di Navigation Service, *getObjectRelationships* (vd 11.1) di Relationship Service.
3. Per scaricare il file collegato ad un documento, utilizzare *getContentStream* (vd 8.7) di Object Service.

2.8 Creazione di una Registrazione di protocollo “in arrivo”

1. Nel caso in cui il documento a cui si vuole associare una registrazione di protocollo non sia presente nel repository, effettuare l’inserimento del documento nella struttura aggregativa desiderata (vd 2.3)
 - Eventualmente, inserimento dei suoi documenti allegati (vd 2.4)
2. Invocazione del servizio *creaRegistrazione* (vd 12.1) di Official Book Services indicando la volontà di creare un oggetto di tipo “in arrivo” a partire da documenti già esistenti in repository (utilizzando la modalità “Protocollazione da documenti esistenti”) e passando, tra i parametri di ingresso, l’identificativo univoco della classificazione principale (ottenuto come risultato dell’invocazione del servizio *creaDocumento* o tramite ricerca).


2.9 Gestione della verifica asincrona della firma

La verifica della firma digitale viene effettuata in fase di inserimento di un documento in modalità sincrona oppure asincrona a seconda di alcune variabili:

- la verifica è **asincrona** se la dimensione del documento è maggiore di una soglia configurata per la linea cliente.
 Lo stesso avviene se i tempi di esecuzione della verifica sono superiori ai 15 secondi (a causa della natura del documento o del numero di firme e relativi certificati da elaborare);
- altrimenti, la verifica è **sincrona**.

È, quindi, necessario gestire la situazione nella quale la verifica della firma non è ancora terminata.

Provando a manipolare un documento sottoposto a verifica asincrona invocando i servizi:

	<p style="text-align: center;">DoQui – ACTA Servizi ACARIS Manuale Integrazione Fruitore</p>	<p style="text-align: right;">ACTA-Acaris-MUT-002</p> <p style="text-align: right;">Pag. 12 di 70</p>
--	---	---

- *creaSmistamento* (vd. 16.2) di *smsService*,
- *moveDocument* (vd. 8.5) di *objectService*,
- *addAssociativeObjectToFolder* (vd. 13.1) di *multifiling*,
- *aggiungiClassificazione* (vd. 13.2) di *multifiling*,
- *creaDocumento* (vd. 9.1) di *documentService*,
 (ma esclusivamente nel caso di aggiunta documento fisico a classificazione esistente o aggiunta contenuto fisico a documento fisico esistente)

si potrebbe ottenere in risposta una di queste eccezioni:

- **SER-E208**
 ACARIS - Il documento target del servizio o uno dei suoi allegati è in attesa di verifica certificati. Impossibile completare la richiesta.
- **SER-E209**
 ACARIS - Il documento target del servizio o uno dei suoi allegati è in attesa di conferma o rifiuto da parte dell'utente. Impossibile completare la richiesta.

Nota:

Il servizio che effettua l'inserimento del documento restituisce lo stesso output sia in caso di verifica sincrona che asincrona.

2.10 Recupero dell'ora della registrazione di protocollo

Per ottenere la data di protocollazione nel formato

yyyy-MM-ddTHH:mm:ss.SSSZ (es: 2022-01-20T11:56:12.000+01:00)

è necessario richiamare il servizio *query* (vd 8.1) di Object Services su entità **EnumObjectType.PROTOCOLLO_PROPERTIES_TYPE**, con il seguente criterio di ricerca:

property name = "**uuidRegistrazione**"

operator = EQUALS

value = `AcarisUtils.decodeClassIdFromObjIdDeChiper(objectIdRegistrazioneProtocollo)`

dove **objectIdRegistrazioneProtocollo** è l'objectId della registrazione di protocollo da ricercare, il cui valore è restituito dal servizio *creaRegistrazione* (vd 12.1) di Official Book Services in **identificazioneRegistrazione.registrazioneId**.

Il valore della data è presente come valore della property "dataProtocollo".

2.11 Invio della segnatura di una registrazione di protocollo e recupero dei dati scambiati

- Partire da una registrazione in partenza presente nel sistema e quindi conoscendo il suo ObjectID, ottenuto come risultato dell'operazione *creaRegistrazione* (vd 12.1) o tramite ricerca dei dati di una registrazione esistente (vd 2.11.1);
- *query* (vd 12.2) di Official Book Services su *SegnaturaProtocolloView* (vd 2.11.2) (integrata con le informazioni necessarie) o su un'altra entità specifica, per sapere se: l'ente richiede il sigillo - il sigillo è già stato apposto. Tali informazioni servono per capire se è già possibile tentare l'invio segnatuta.
- *query* (vd 12.2) di Official Book Services su *CorrispondentiRegistrazioneView* (vd 2.11.4): passaggio opzionale. Nel caso si intenda inviare la segnatuta di protocollo solo ad un sottoinsieme dei destinatari esterni dichiarati per la registrazione di protocollo, oppure se si vuole modificare l'indirizzo e-mail di un destinatario, è necessario conoscere l'identificativo (objectIdCorrispondente) del corrispondente relativo. Per ottenere questo dato, invocare l'operazione *query* sull'entità target

“CorrispondentiRegistrazioneView” per recuperare i dati dei destinatari (corrispondenti esterni di tipo destinatario), tra cui l’identificativo di ognuno (da utilizzare come dato di ingresso dell’operazione “inviaSegnaturaRegistrazione”)

- *inviaSegnaturaRegistrazione* (vd 12.4) di Official Book Services: effettua l’invio della segnatura di una registrazione di protocollo.
- *query* (vd 12.2) di Official Book Services su *RichiestaInvioSegnaturaView* (vd 2.11.3). Si tratta di una nuova entità di ricerca che, utilizzando l’id restituito dall’operazione *inviaSegnaturaRegistrazione* oppure l’objectID della registrazione restituito da *creaRegistrazione*, restituisce lo stato “generale” delle richieste d’invio presenti nel sistema. Sono previsti questi stati:
 - pianificata (l’elaborazione non è ancora iniziata)
 - in elaborazione (è in corso l’elaborazione asincrona della richiesta, che può essere costuita da più parti nel caso in cui è stato richiesto l’invio separato a destinatari diversi)
 - completata (elaborazione di tutte le parti della richiesta è terminata)
- *ricercaDatiInterscambio* (vd 12.5) di Official Book Services: dopo aver appurato che la segnatura è stata inviata tramite la ricerca su *RichiestaInvioSegnaturaView*, possiamo invocare questa operazione per ottenere l’elenco dei metadati relativi ai messaggi scambiati.
- *getContentStream* (vd 8.7) di Object Service: operazione puntuale che permette di fare il download dei dati relativi a un messaggio parte di una comunicazione interscambio (il cui elenco è ottenibile tramite operazione *ricercaDatiInterscambio*).

Si rimanda al documento [11] per una descrizione più approfondita di questi servizi legati all’invio di una segnatura di una registrazione di protocollo.

2.11.1 Ottenere le informazioni di dettaglio su una registrazione

Le informazioni di dettaglio su una registrazione (e in particolare il suo objectID se non è stato memorizzato al momento del *creaRegistrazione*) si possono ottenere conoscendo il numero e la data di protocollo (oltre agli identificativi di AOO/Struttura/Nodo protocollante, nel caso siano presenti più AOO per l’ente di riferimento), interrogando il target **RegistrazioneView** e impostando le seguenti properties di tipo “*queryable*”/“*filtrabile*”:

property	caratteristiche
codice	selezionabile = S filtrabile = S tipoDato = String lista_operatori = EQUALS, LIKE note: numero della registrazione di protocollo (codice), nel formato [0-9]{7}
dataProtocollo	selezionabile = S filtrabile = S tipoDato = Timestamp lista_operatori = EQUALS, GREATER_THAN, LESS_THAN, GREATER_THAN_OR_EQUAL_TO, LESS_THAN_OR_EQUAL_TO note: data della registrazione di protocollo in formato yyyy-MM-dd
idAOProtocollante	selezionabile = S filtrabile = S tipoDato = BigInteger lista_operatori = EQUALS, GREATER_THAN, LESS_THAN note: identificativo della AOO protocollante
idStrutturaProtocol- lante	selezionabile = S filtrabile = S tipoDato = BigInteger lista_operatori = EQUALS, GREATER_THAN, LESS_THAN

	note: identificativo della struttura protocollante
idNodoProtocollante	selezionabile = S filtrabile = S tipoDato = BigInteger lista_operator = EQUALS, GREATER_THAN, LESS_THAN note: identificativo del nodo protocollante

2.11.2 Ottenere lo stato di apposizione del sigillo

Conoscere lo stato di apposizione del sigillo è indispensabile per decidere se proseguire o meno con le operazioni di invio della segnatura.

Sapendo l'objectId della registrazione di protocollo da inviare, o in alternativa il numero e la data di protocollo, è possibile interrogare il target **SegnaturaProtocolloView**. Nel primo caso, usare la property "objectId", nel secondo usando "codice" e "dataProtocollo"; di seguito la descrizione delle properties citate:

property	caratteristiche
codice	selezionabile = S filtrabile = S tipoDato = String lista_operator = EQUALS, LIKE note: numero della registrazione di protocollo (codice), nel formato [0-9]{7}
dataProtocollo	selezionabile = S filtrabile = S tipoDato = Timestamp lista_operator = EQUALS, GREATER_THAN, LESS_THAN, GREATER_THAN_OR_EQUAL_TO, LESS_THAN_OR_EQUAL_TO note: data della registrazione di protocollo in formato yyyy-MM-dd
objectId	selezionabile = S filtrabile = S tipoDato = String lista_operator = EQUALS, LIKE note: identificativo (object id) della registrazione di protocollo legata alla segnatura.

Nella response si evidenziano le seguenti properties:

property	caratteristiche
dtApposizioneSigillo	selezionabile = S filtrabile = N tipoDato = Timestamp lista_operator = EQUALS, GREATER_THAN, LESS_THAN note: data di apposizione del sigillo alla segnatura di protocollo
flagSigilloObbligatorio	selezionabile = S filtrabile = N tipoDato = Boolean lista_operator = EQUALS note: valore booleano (valori possibili 'S' / 'N') che indica se l'apposizione del sigillo alla segnatura è obbligatoria per la AOO protocollante
segnaturaInterna	selezionabile = S filtrabile = N tipoDato = String lista_operator = EQUALS, GREATER_THAN, LESS_THAN note: segnatura di protocollo interna (generata da ACTA al momento della creazione della registrazione di protocollo)

2.11.3 Ottenere lo stato di invio del messaggio

Per conoscere lo stato generale dell'invio della segnatrice è disponibile l'entità **RichiestaInvioSegnatrice-View** che può essere interrogata usando una di queste 2 properties:

property	caratteristiche
objectIdRichiestaInvio	selezionabile = S filtrabile = S tipoDato = String lista_operator = EQUALS, LIKE note: identificativo (object id) della richiesta di invio segnatrice. Output dell'operazione invioSegnatriceRegistrazione esposta da officialbookService
objectIdRegistrazione	selezionabile = S filtrabile = S tipoDato = String lista_operator = EQUALS, LIKE note: identificativo (object id) della registrazione di protocollo legata ad una richiesta di invio segnatrice.

Nella response si evidenziano le seguenti properties:

property	caratteristiche
destinatari	selezionabile = S filtrabile = S tipoDato = String lista_operator = EQUALS, GREATER_THAN, LESSER_THAN note: elenco degli indirizzi mail (PEC o PEO) per cui è stata effettuata la richiesta di invio segnatrice. Le mail in elenco sono separate utilizzando il carattere ';' (es: 'mail1@pec.doqui.fake.it;mail2@peo.doqui.fake.it')
objectIdMessaggioInterscambio	selezionabile = S filtrabile = S tipoDato = String lista_operator = EQUALS, LIKE note: identificativo (object id) del messaggio interscambio creato in seguito alla richiesta di invio. Può essere utilizzato come input dell'operazione ricerca-DatiInterscambio esposta da officialbookService
descStato	selezionabile = S filtrabile = N tipoDato = String lista_operator = EQUALS, LIKE note: descrizione dello stato della richiesta di invio segnatrice ed indica lo stato complessivo di avanzamento dei job (vedi nota) legati alla richiesta. Possibili valori: <ul style="list-style-type: none"> - pianificata: l'elaborazione non è ancora iniziata - in elaborazione: l'elaborazione della richiesta è in corso (l'elaborazione di almeno un job è in corso) - completata: elaborazione terminata, quindi tutti i job hanno completato la loro parte NOTA: Per ogni richiesta di invio, in base ai parametri di input dell'operazione invio-SegnatriceRegistrazione (invioMultiplo), possono essere creati dal sistema da 1 a più job che hanno il compito di gestire l'invio della segnatrice di interscambio tramite il canale indicato (nella fase attuale dell'implementazione, unico canale è PEC).
flagInvioEffettuato	selezionabile = S filtrabile = N tipoDato = String lista_operator = EQUALS, LIKE

	note: valore (possibili valori: S/N) che indica se l'invio della segnatura è stato effettuato o meno da un dato job legato alla richiesta di invio segnatura (vd nota 1). Un job può completare la sua elaborazione senza effettuare l'invio nel caso limite in cui <ul style="list-style-type: none"> 1- è stato richiesto l'invio non prioritario 2- la dimensione dei documenti da allegare alla pec in invio è superiore al limite previsto per l'ente (attualmente configurato a 100MB) 3- la AOO legata al mittente (quella protocollante) non prevede la presenza di collocazioni telematiche all'interno della segnatura
flagJobCompletato	selezionabile = S filtrabile = N tipoDato = String lista_operator = EQUALS, LIKE note: valore (possibili valori: S/N) che indica se il job (vd nota 1) legato alla richiesta di invio segnatura è completato.

NOTE:

(1) Per ogni richiesta di invio, in base ai parametri di input dell'operazione invioSegnaturaRegistrazione (invioMulti-
plo), possono essere creati dal sistema da 1 a più job che hanno il compito di gestire l'invio della segnatura di inter-
scambio tramite il canale indicato (nella fase attuale dell'implementazione, unico canale è PEC).

2.11.4 Ottenere i dati dei corrispondenti di una registrazione


Per conoscere i dati dettagliati dei corrispondenti di una registrazione è disponibile l'entità **Corrispondenti-RegistrazioneView** che può essere interrogata usando queste properties:

property	caratteristiche
objectIdRegistrazione	selezionabile = S filtrabile = S tipoDato = String lista_operator = EQUALS, LIKE note: identificativo (object id) della registrazione di protocollo legata ad una richiesta di invio segnatura.
tipoMD	selezionabile = S filtrabile = S tipoDato = String lista_operator = EQUALS, LIKE note: tipo corrispondente, M -> mittente, D -> destinatario
flagInterno	selezionabile = S filtrabile = S tipoDato = String lista_operator = EQUALS, LIKE note: valore booleano (valori possibili 'S' / 'N') che indica se il corrispondente è interno o meno

Nella response si evidenziano le seguenti properties:

property	caratteristiche
codTipoCorrispondente	selezionabile = S filtrabile = N tipoDato = String lista_operator = EQUALS, LIKE note: codice che identifica il tipo di corrispondente (valori: PG / PF / PA / AE)
objectIdCorrispondente	selezionabile = S filtrabile = N tipoDato = String lista_operator = EQUALS, LIKE

	note: identificativo (object id) del corrispondente Può essere utilizzato nell'invioSegnaturaRegistrazione per modificare l'indirizzo di un destinatario o per escluderlo del tutto.
flagRichConfermaRicezione	selezionabile = S filtrabile = N tipoDato = String lista_operatori = EQUALS, LIKE note: valore booleano (valori possibili 'S' / 'N') che indica se è richiesto l'invio del messaggio interscambio di conferma ricezione per questo corrispondente in seguito all'invio segnatura
indirizzoTelematico-Smtp	selezionabile = S filtrabile = N tipoDato = String lista_operatori = EQUALS, LIKE note: indirizzo telematico SMTP (posta elettronica) per un corrispondente di tipo AE / PG / PF
indirizzoTelematico-SmtpPa	selezionabile = S filtrabile = N tipoDato = String lista_operatori = EQUALS, LIKE note: indirizzo telematico SMTP (posta elettronica) per un corrispondente di tipo PA
indirizzoTelematico-SmtpAoo	selezionabile = S filtrabile = N tipoDato = String lista_operatori = EQUALS, LIKE note: indirizzo telematico SMTP (posta elettronica) per la AOO del corrispondente di tipo PA
indirizzoTelematico-SmtpUo	selezionabile = S filtrabile = N tipoDato = String lista_operatori = EQUALS, LIKE note: indirizzo telematico SMTP (posta elettronica) per la unità organizzativa (UO) del corrispondente di tipo PA

	<p style="text-align: center;">DoQui – ACTA Servizi ACARIS Manuale Integrazione Fruitore</p>	<p style="text-align: right;">ACTA-Acaris-MUT-002</p> <p style="text-align: right;">Pag. 18 di 70</p>
--	---	---

3 Specifiche tecniche di integrazione

Le funzioni di business di ACTA sono fruibili da applicazioni esterne tramite l’invocazione di servizi applicativi. I servizi applicativi di ACTA sono stateless e ogni chiamata agli stessi rappresenta un’unità atomica di lavoro. Nel caso in cui sia necessario mantenere uno stato fra una chiamata e l’altra, tale operazione è a carico dell’applicazione fruitrice.

I servizi vengono definiti avendo come riferimento lo standard CMIS.

I riferimenti sono il data model e le interfacce previste dallo standard: entrambi sono stati personalizzati allo scopo di adattarli alle esigenze specifiche di ACTA.

Il data model ha ispirato il corrispondente “object model” dei servizi applicativi ACTA.

Le interfacce, opportunamente adattate, costituiscono la base per la definizione dell’interfaccia di esposizione dei servizi. I servizi del sottomodulo di “Management”, però, essendo specifici del business di ACTA non trovano riscontro nello standard CMIS.

3.1 Object Model

Definisce la rappresentazione di tutti gli oggetti (ed oggetti da questi referenziati) che compaiono nelle firme dei servizi esposti tramite l’interfaccia pubblica degli stessi.

L’Object Model viene definito attraverso schemi espressi in W3C XML Schema syntax.

La previsione di esposizione dei servizi applicativi, anche, come Web Services, condiziona in tal senso e guida la definizione dell’object model e della sua struttura.

Tale scelta, obbligata a causa del riferimento allo standard CMIS, consente di riferire il modello all’utilizzo di standard riconosciuti con il vantaggio di poter utilizzare tools che consentono, a partire dal modello, di realizzare in automatico il binding con il linguaggio scelto per l’implementazione: le classi Java di implementazione del modello vengono generate a partire dalle definizioni contenute negli schemi XML. Analoga procedura viene seguita per le interfacce dei servizi che vengono generate a partire da appositi WSDL.

Opportune annotazioni consentono di utilizzare compilatori per eseguire adattamenti durante il processo di binding; questo avviene in modo trasparente: cioè essi non sono resi pubblici attraverso il contratto (XSD e WSDL) rilasciato ai fruitori.

3.1.1 Modelli per l’utilizzo di “properties”

Alcune operazioni previste dallo standard CMIS (ad esempio quelle di creazione di oggetti) prevedono un parametro denominato properties.

Lo standard prevede che il parametro possa essere modellato in modo funzionale alle esigenze dello specifico dominio di applicazione.


La modellazione può basarsi su diverse soluzioni delle quali, però, le più probabili sono:

- Applicazione di un modello generico
- Applicazione di un modello tipizzato

Per modello generico s’intende la previsione di un parametro nel quale sono appiattiti i riferimenti al tipo di dato, al valore ed alla classe di appartenenza.

Per modello **tipizzato** s’intende la previsione dell’utilizzo di strutture dati specifiche per la rappresentazione dei corrispondenti oggetti di dominio: es. folder, voce, aggregazione, fascicolo...

L’object model ACARIS è basato su un modello tipizzato.

	<p style="text-align: center;">DoQui – ACTA Servizi ACARIS Manuale Integrazione Fruitore</p>	<p style="text-align: right;">ACTA-Acaris-MUT-002</p> <p style="text-align: right;">Pag. 19 di 70</p>
--	---	---

3.2 Definizione interfaccia

Nel disegno delle interfacce esposte da servizi possono essere adottati due modelli di sviluppo che in letteratura sono così definiti:

- **“Start from Java”** oppure “Bottom Up” nel caso di esposizione come servizi di logiche implementate in classi già esistenti
- **“Start from WSDL”** oppure “Top Down” oppure “Contract First” nel caso di esposizione come servizi di interfacce generate tramite un processo che utilizza WSDL preesistenti

3.2.1 Approccio “Contract First”

Il riferimento alla specifica CMIS ha reso obbligatoria la scelta di tale modello di sviluppo.

La soluzione, coerentemente con l’approccio CMIS di definizione delle interfacce, privilegia il contratto con i fruitori esposto tramite Web Services

Le interfacce CMIS sono già definite nella specifica: ACARIS le ha adattate al contesto ACTA ed ha modellato tramite XML Schema l’*Object Model* che rappresenta l’estensione del *data model* CMIS ai fini della rappresentazione del particolare “domino” di applicazione.

Il contratto (interfacce esposte ai fruitori dei servizi, soprattutto Web Services) è costituito tramite WSDL, scaricabile da apposito url secondo quanto stabilito dalle specifiche in tema di esposizione Web Services.

Sono previsti i WSDL di definizione dei seguenti servizi:

Management, Multifiling, Navigation, Object, Relationships, Repository, Document, OfficialBook, BackOffice, SMS, SubjectRegistry.

Attraverso il WSDL del singolo servizio il fruitore, effettuando un binding con il proprio linguaggio di implementazione, genera una libreria di classi locale che utilizza per invocare le operazioni esposte dal singolo servizio. Per i fruitori che implementano l’integrazione utilizzando il linguaggio Java è resa comunque disponibile una libreria (actasrv-client.jar) che contiene le classi generate nella modalità appena descritta a partire dal WSDL utilizzando il tool **wsdl2java**.

Per non appesantire il contratto con dettagli riguardanti il *binding* con un particolare linguaggio, ACARIS segue l’approccio *external binding declaration*.

Il *SOAP message* in input e output contiene nel body di *envelope* la rappresentazione XML (*marshalling*) del tipo definito nell’XML schema. Il framework sul quale viene basata l’implementazione del servizio effettua un’operazione di *unmarshalling* del messaggio SOAP e costruisce delle classi Java contenute nella libreria costituente l’object model ACARIS.

Per la documentazione relativa fare riferimento al paragrafo 1.3 e al paragrafo Riferimenti Tecnologici.

3.2.2 Messaggi

Per lo schema di definizione fare riferimento ai documenti al paragrafo 1.3.

Questo schema elenca:

- *Complex types* costituenti le sezioni “Request” e “Response” di ogni messaggio
- Definizioni di tipi che compaiono nelle segnature dei messaggi ma non fanno direttamente parte dell’object model
- La definizione di *AcarisException* e *AcarisFaultType*

I diversi WSDL utilizzano i messaggi per la costruzione delle operazioni che costituiscono l’interfaccia di ogni servizio.

4 Riferimenti Tecnologici

4.1 Specifiche

- JAX-WS 2.x
si vedano i riferimenti di letteratura per quanto riguarda le specifiche SUN in tema di realizzazione di Web Services e la documentazione specifica di [\[8\]](#)
- JAXB 2.x
rappresenta la specifica riferita come standard da JAX-WS per effettuare il *binding* Java/XML in modo da rappresentare *Schema components* (elementi, tipi, attributi) come *Java contents* (classi, bean properties)
- C.S.I.

4.2 Framework

CXF 2.2.3: rappresenta il framework adottato da CSI per l'introduzione delle specifiche JAX-WS 2.x nell'attuale stack tecnologico per l'implementazione della SOA.

4.3 Librerie

Per quanto riguarda il Wrapper WS (Web Services) si faccia riferimento alle librerie utilizzate in [\[8\]](#).

Poiché il processo di generazione automatica dell'object model, a partire dai WSDL dei servizi, è attività che prescinde dalla modalità di esposizione, in questo allegato viene fatto un espresso riferimento alla libreria **jaxb-api-2.1.jar**

4.4 Libreria di implementazione dell'Object Model

Il tool **wsdl2java** previsto in CXF [\[8\]](#) può effettuare la generazione sia dei value object corrispondenti ai tipi definiti nei WSDL/XSD passati in input (object model), che delle interfacce esportate dai singoli servizi sulla base delle definizioni contenute nei WSDL: *Service Endpoint Interface*.

Attraverso apposite istruzioni contenute in [\[9\]](#) è possibile personalizzare l'alberatura di package delle classi generate in modo da renderla compatibile con le linee guida CSI in merito.

La libreria di classi così generata deve essere disponibile all'implementazione dei servizi ACARIS e disponibile nel relativo classpath.

4.5 Interfaccia Wrapper Web Services

Utilizza la *Service Endpoint Interface* (SEI) di ogni servizio così come generata dall'apposito tool a partire dal WSDL di ogni servizio.

La SEI viene implementata dal relativo *Service Implementation Bean* (SIB) il cui compito, esauriti eventuali compiti di adattamento, è quello di invocare il vero e proprio *ServiceImpl* per l'esecuzione delle logiche di servizio.

4.6 AcarisException

Viene definita in [\[6\]](#) insieme alle definizioni dei "messaggi" costituenti rispettivamente la parte "*Request*" e "*Response*" di ogni operazione.

AcarisException viene definita come un element di tipo *AcarisFaultType*:

```
<xs:complexType name="acarisFaultType">
  <xs:sequence>
    <xs:element name="exceptionType" type="tns:enumServiceException" />
    <xs:element name="errorCode" type="tns:enumErrorCodeType" />
    <xs:element name="technicalInfo" type="tns:string" />
    <xs:element name="objectId" type="tns:ObjectIdType" />
    <xs:element name="className" type="tns:enumObjectType" />
  </xs:sequence>
</xs:complexType>
```

```

        <xs:element name="propertyName" type="tns:string" />
      </xs:sequence>
</xs:complexType>
<xs:element name="acarisException" type="tns:acarisFaultType" />

```

4.7 Eccezioni e gestione Fault in ACARIS

AcarisException viene utilizzata in funzione di “*fault*” nei WSDL di definizione dei singoli servizi. Si rimanda alla letteratura su SOAP sulla definizione del contenuto di *envelope* per un approfondimento sullo scopo ed il funzionamento di *SOAP fault message*.

In ACARIS, il tool *wsdl2java*, attraverso le definizioni contenute nei singoli WSDL dei servizi, genera in automatico la segnatura della singola operazione utilizzando quanto definito come “*fault*” in funzione di eccezione sollevata dalla stessa operazione.

Vanno inserite nell'alberatura standard: AcarisException, nella versione generata in automatico da *wsdl2java*, quindi, viene inserita in `it.doqui.acta.actasrv.exception`. Questa versione viene utilizzata dal ServiceImpl e dal wrapper SOAP.

4.8 Gestione allegati

In ACARIS la gestione degli allegati è effettuata tramite la classe *AcarisContentStreamType* definita come segue in [\[6\]](#).

```

<!-- Content Stream -->
<xs:complexType name="acarisContentStreamType">
  <xs:sequence>
    <xs:element name="length" type="tns:contentStreamLengthType" />
    <xs:element name="mimeType" type="tns:enumMimeTypeType" minOccurs="0" />
    <xs:element name="filename" type="tns:contentStreamFilenameType" minOccurs="0" />
    <xs:element name="streamMTOM" type="tns:base64Binary"
      xmime:expectedContentTypes="*/*" />
    <xs:element name="stream" type="tns:base64Binary" />
  </xs:sequence>
</xs:complexType>

```

La classe è modellata in modo da gestire il passaggio dei parametri compatibilmente con la doppia modalità di esposizione.

4.9 DTO

Le classi che rappresentano l'*object model* di ACARIS devono essere assegnate al layer apposito all'interno della componente utilizzata.

Esempio: `it.doqui.acta.actasrv.dto.acaris.common`

In sede di generazione automatica delle classi Java a partire dai *wsdl*, tramite una *binding declaration* in [\[9\]](#), è possibile decidere conformemente l'alberatura di package dei DTO costituenti l'*object model*.

I DTO, specifici di una particolare modalità di esposizione, possono essere inseriti in un package **dto** apposito dentro il wrapper di riferimento: ad esempio i DTO usati per il mapping tra modello tipizzato e generico usati nel “*generic wrapper*”.

Esempio: `it.doqui.acta.actasrv.util.acaris.wrapper.dto.common.PropertyType`

5 Il catalogo dei servizi e delle operazioni

I servizi sono fruibili sia da esposizione su rete Internet che su rete interna CSI.

Le operazioni messe a disposizione sono suddivise in diversi servizi che le raggruppano logicamente e che sono riconducibili ai moduli di ACTA.

- RepositoryService
- BackOfficeService
- ObjectService
- DocumentService
- RelationshipService
- NavigationService
- MultifilingService
- ManagementService
- OfficialBookService
- SubjectRegistryService
- SMSService

Nei successivi capitoli saranno illustrati le operazioni esposte per i diversi servizi e, per ognuna di queste, sono riportate le tabelle che descrivono i dati dal punto di vista logico/funzionale, per il dettaglio completo delle strutture e una spiegazione dettagliata delle operazioni esposte dai servizi, vedi i documenti in elenco al paragrafo 1.3.

5.1 Modalità di esposizione dei servizi ACARIS

I servizi applicativi ACARIS prevedono la modalità Web Service come principale modalità di esposizione.

Il singolo servizio viene esposto tramite un Wrapper costruito davanti alla componente software che effettua l'implementazione vera e propria del servizio.

I Web Services implementati sono standard e conformi al **WS-I Basic Profile**.

Il framework utilizzato è il **CXF**.

5.2 Fruizione internet APIStore

Le API pubblicate sull' API Manager sono disponibili sullo Store al seguente indirizzo:

<https://api-piemonte-store.csi.it/>

Per poter accedere alle API è necessario che il fruitore completi i seguenti passi operativi:

- Accreditamento allo Store
- Creazione dell'applicazione contenitore delle API
- Sottoscrizione dell'API di interesse
- Utilizzo delle chiavi OAuth2 per accedere all'API

6 RepositoryService

I servizi di questa classe permettono ai client di ACTA di recuperare informazioni quali:

- i repository disponibili su di un ACTA service endpoint
- i metadati (Repository Name e Repository Identity) necessari a individuare l'Ente attestato su quel repository ed a riferire univocamente al repository nelle operazioni CRUD degli ACTA Services
- le configurazioni applicative del repository

6.1 L'operazione getRepositories

Si tratta dell'unico servizio non specifico per un singolo repository (Ente). Permette ai client di ACTA di individuare il repository su cui operare tra quelli disponibili ad un service endpoint.

La localizzazione del repository avviene per mezzo dei dati caratteristici dell'ente: codice, denominazione, etc.

Questa operazione non richiede parametri di ingresso.

Tabella 1: dati di output getRepositories

Liv	Dato	Annotazione
1	AcarisRepositoryEntryType[]	Elenco di oggetti che identificano un repository configurato nel service endpoint
-2	acarRepositoryEntry	Singolo oggetto
--3	repositoryId	Identificativo ACARIS del repository
--3	repositoryName	Nome completo identificativo del repository

7 BackOfficeService

Insieme di servizi che consentono di ottenere informazioni sulle entità utilizzate nel modulo back office di ACTA che gestisce la struttura organizzativa degli utenti che sono abilitati all'utilizzo del repository. Offre anche dei servizi specifici del modello degli ACTA Services necessari per ottenere metadati per la navigazione e la ricerca degli oggetti attraverso l'interfacciamento a servizi.

7.1 L'operazione getPrincipalExt

Questa operazione consente di recuperare l'elenco degli eventuali "principal" che possono operare sul repository (Ente) di interesse per conto di una determinata applicazione client. Un principal è un token applicativo rilasciato da ACTA, ha durata temporale limitata ed è richiesto durante l'invocazione delle altre operazioni. Attraverso questo token, il sistema identifica l'applicativo fruitore e l'utente per conto del quale è richiesta l'operazione, in modo da effettuare i controlli di accesso alle risorse e sulle operazioni a cui un utente è abilitato, in base a quanto configurato nel back office di ACTA.

Questa operazione rimpiazza la getPrincipal, tuttora presente per retrocompatibilità ma di fatto deprecata.

Tabella 2: dati di input getPrincipalExt

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	idUtente	Codice fiscale di un utente censito in organigramma dell'Ente
1	idAOO	Identificativo dell'AOO in cui è collocato l'utente
1	idStruttura	Identificativo della struttura organizzativa in cui è collocato l'utente
1	idNodo	Identificativo del nodo organizzativo in cui è collocato l'utente
1	clientApplicationInfo	Dati che identificano l'applicativo client
-2	appKey	Stringa alfanumerica che identifica univocamente l'applicativo fruitore

Tabella 3: dati di output getPrincipalExt

Liv	Dato	Annotazione
1	PrincipalExtResponses[]	Elenco di oggetti contenenti dati relativi ad un utente collocato in organigramma ACTA
-2	principalExtResponse	Singolo oggetto
--3	principalId	Token applicativo, con valenza temporale, legato all'applicativo client e l'utente utilizzato per richiamare questa operazione
--3	collocazioneUtente	Identificativo della struttura organizzativa in cui è collocato l'utente
---4	Aoo	Dato identificativo dell'AOO di collocazione

---4	Struttura	Dato identificativo della struttura di collocazione
---4	Nodo	Dato identificativo del nodo di collocazione
--3	Profili	Elenco di descrizioni dei profili abilitati per questa collocazione
----4	identificatore	Identificatore del profilo
----4	Codice	Codice del profilo

7.2 L'operazione getQueryableObjects

Questa operazione permette di ottenere l'elenco degli oggetti di tipo Queryable che possono essere utilizzati nelle operazioni di query come target. Può trattarsi di entità fisiche oppure di viste create dinamicamente dal sistema.

L'elenco è specifico per l'ente di interesse.

Tabella 4: dati di input getQueryableObjects

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
2	service	Nome del service di cui si vogliono conoscere gli oggetti ricercabili <ul style="list-style-type: none"> • Archive • BackOffice • Management • OfficialBook • Sms • SubjectRegistry Archive corrisponde al service "Object"

Tabella 5: dati di output getQueryableObjects

Liv	Dato	Annotazione
1	QueryableObject[]	Elenco di QueryableObject
-2	queryableObject	Singolo oggetto
--3	object	Nome dell'entità che può essere utilizzata nelle operazioni di ricerca e recupero metadati

7.3 L'operazione getQueryableObjectMetadata

Questa operazione permette di ottenere l'elenco dei metadati associati ad un oggetto di tipo Queryable utilizzabile in un'operazione di ricerca (query, getProperties, getDescendants, getChildren, getFolderParent, getObjectParents). L'elenco ottenuto è specifico per l'ente di interesse.

Tabella 6: dati di input getQueryableObjectMetadata

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	queryableObject	Identificatore di una entità ricercabile (oggetto Queryable)
1	operation	Identificatore dell'operazione di ricerca (servizio ACARIS)

Tabella 7: dati di output getQueryableObjectMetadata

Liv	Dato	Annotazione
1	QueryableObjectMetadataList []	Lista di metadati di queryableObject
-2	queryableObjectMetadataList	Singolo Oggetto
--3	queryName	Identificatore di un metadato (property) dell'entità queryable
---4	className	Classe a cui appartiene la property
---4	propertyName	Nome della property
--3	selectable	Indica se il metadato è selezionabile
--3	queryable	Indica se il metadato è ricercabile
--3	updatable	Indica se il metadato è aggiornabile
--3	dataType	Indica il tipo: Boolean, String, BigInteger, ...
--3	propertyFilterConfigurationInfo	Indicazioni, in base all'operation in input, dei filtri di ricerca utilizzabili per ottenere questo metadato

7.4 L'operazione query

Consente di effettuare delle ricerche sugli oggetti di competenza del modulo funzionale Back office: appartengono a questo modulo gli oggetti che modellano le strutture organizzative (AOO, Struttura, Nodo), gli Enti e gli utenti configurati nel sistema

Tabella 8: dati di input query

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)

1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	target	Indicazione della tipologia di oggetto di cui effettuare la ricerca
1	filter	Filtro da applicare alla ricerca (oggetto di tipo PropertyFilterType, vd <i>Tabella 85</i>)
1	criteria	Elenco di criteri di ricerca
-2	criterio	Singolo criterio (oggetto di tipo QueryConditionType, vd <i>Tabella 83</i>)
1	navigationLimits	Opzionale, dati per limitare la ricerca ad un sottoalbero dei dati presenti per l'ente
1	maxItems	Indica il numero massimo di oggetti attesi in output. Utilizzato insieme a skipCount per effettuare una paginazione dei risultati.
	skipCount	Indica l'indice dell'elemento da cui continuare nella restituzione degli oggetti trovati. Utilizzato insieme a maxItems per effettuare una paginazione dei risultati.

Tabella 9: dati di output query

Liv	Dato	Annotazione
1	pagingResponse	Risultato della ricerca (oggetto di tipo PagingResponseType, vd <i>Tabella 86</i>)

8 ObjectService

Per gli oggetti che possono essere ricondotti alle tipologie base dello standard CMIS, gli ACTA Services offrono operazioni CRUD per l'oggetto coinvolto identificato dal proprio ID (object ID), per ognuna delle operazioni elementari si distinguono:

- servizi di creazione di oggetti come createDocument, createFolder, createRelationship e restituzione dell'identificativo univoco;
- servizi di lettura delle proprietà degli oggetti e, opzionalmente, delle operazioni permesse su di essi;
- servizi di aggiornamento delle proprietà degli oggetti, per quelle a valore multiplo occorre sostituire l'intera lista di valori, le proprietà read-only non sono aggiornabili dai client;
- supporto all' "optimistic locking" per mezzo della proprietà ChangeToken: si tratta di un attributo generato dal repository, e da questo dipendente, che permette a quest'ultimo di verificare in fase di update se l'oggetto è stato aggiornato rispetto alla precedente lettura;
- servizi di creazione e lettura di content stream per mezzo dell'identificativo dell'oggetto Document di appartenenza, se un contenuto è già stato associato ad un Document al momento della creazione dello stream il contenuto è sovrascritto.

Non tutte le operazioni elencate sono sempre possibili per ogni istanza degli oggetti del modello, ciò non solo per le limitazioni imposte dai tipi di oggetti definiti nel modello ma anche per le policy di gestione dell'archivio che modificano nel tempo le operazioni possibili sulle istanze (trasferimenti da corrente a deposito, attività collaborative sui documenti ed altri processi propri del ciclo di vita dei contenuti).

Sebbene un content stream possa apparire una proprietà di un oggetto Document in realtà i servizi che restituiscono le proprietà dell'oggetto, o permettono di ricercare per mezzo di queste, non restituiscono un content stream.

8.1 L'operazione query

Consente di effettuare delle ricerche sugli oggetti di competenza del modulo funzionale Archive: appartengono a questo modulo gli oggetti che modellano le strutture aggregative, le classificazioni, i documenti.

Questa operazione è esposta su più service con la medesima firma: fare riferimento a Tabella 8: dati di input query e Tabella 9: dati di output query per l'elenco dei dati di input e output dell'operazione.

8.2 L'operazione createDocument

Crea un oggetto documento del tipo specificato e aggiunge il documento in un folder.

Se è specificato un Folder ed il tipo di Documento non è fra quelli permessi per questo Folder viene sollevata una AcarisException con messaggio SER-E902. Se l'objectType non accetta un contentStream viene sollevata una AcarisException con messaggio SER-E903.

Tabella 10: dati di input createDocument

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	typeId	Identificativo del tipo di documento. Ammesso solo ContenutoFisicoPropertiesType
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi

		ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	properties	Elenco delle properties che caratterizzano il documento
1	folderId	Identificativo ACARID del folder padre per il documento che si vuole creare
1	contentStream	Oggetto che contiene i dati su un file fisico, nel caso di aggiunta di un documento elettronico. Necessario per typeId = ContenutoFisicoPropertiesType

Tabella 11: dati di output createDocument

Liv	Dato	Annotazione
1	objectId	Identificativo ACARIS del documento creato dall'operazione

8.3 L'operazione getProperties

Consente di ottenere una lista di proprietà dell'oggetto specificato come parametro, scelte fra quelle selezionate, in base al filtro sulle proprietà richieste come valore di ritorno.

Tabella 12: dati di input getProperties

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	objectId	Identificativo ACARIS dell'oggetto di cui si richiedono i metadati
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	filter	Filtro utilizzato per indicare le proprietà richieste in output

Tabella 13: dati di output getProperties

Liv	Dato	Annotazione
1	objectResponse	Contiene i metadati richiesti (oggetto di tipo ObjectResponseType, vd Tabella 88)

8.4 L'operazione updateProperties

L'operazione consente l'aggiornamento delle property di un oggetto specifico.

La tipologia dell'oggetto da modificare viene stabilita facendo riferimento a quanto specificato nel parametro del tipo PropertyType. Vengono modificate solo le properties espressamente elencate nel parametro properties di tipo PropertyType[].

La politica adottata per l'aggiornamento è quella della completa sovrascrittura dei valori esistenti per gli oggetti presenti in ACTA, qualora il nuovo valore rispetti le regole per l'oggetto in modifica.

Tabella 14: dati di input updateProperties

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	objectId	Identificativo ACARIS dell'oggetto di cui vuole aggiornare le property
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	changeToken	Informazione temporale necessaria per verificare l'accesso in concorrenza.
1	properties	Lista di properties da modificare e relativo valore (oggetti di tipo PropertyType, vd Tabella 89)

Tabella 15: dati di output updateProperties

Liv	Dato	Annotazione
1	simpleResponse	Dati relativi all'oggetto su cui è stata effettuata la modifica (oggetto di tipo SimpleResponseType, vd Tabella 90)

8.5 L'operazione moveDocument

Consente di effettuare lo spostamento di un documento (con gli eventuali allegati), da una struttura aggregativa di partenza ad un'altra di destinazione.

Tabella 16: dati di input moveDocument

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	associativeObjectId	Identificativo ACARIS dell'oggetto da spostare
1	sourceFolderId	Identificativo ACARIS della struttura aggregativa di partenza
1	targetFolderId	Identificativo ACARIS della struttura aggregativa di destinazione
1	associativeProperties	Proprietà da indicare per pilotare l'esecuzione dell'operazione
-2	offlineMoveRequest	Indica se si vuole chiedere l'esecuzione asincrona dell'operazione qualora il documento da spostare abbia un numero di allegati superiore a quelli previsti

Tabella 17: dati di output moveDocument

Liv	Dato	Annotazione
1	objectId	Questo oggetto può assumere più valori, in alternativa tra loro <ul style="list-style-type: none"> a) Identificativo ACARIS della richiesta asincrona b) Identificativo ACARIS dell'oggetto dopo lo spostamento, nel caso di operazione non asincrona

8.6 L'operazione createFolder

Questa operazione consente di creare una struttura aggregativa, che è un oggetto di un tipo derivato dalla classe base Folder, ed eventualmente creare una gerarchia padre-figlio con una voce o un'altra struttura aggregativa.

Il tipo dell'oggetto da creare deve essere compatibile con il tipo dell'oggetto Folder padre.

I tipi di oggetti creabili sono

1. Fascicolo Reale
 - a. Annuale
 - b. Continuo
 - c. Legislatura
 - d. Libero
 - e. Ereditato
2. Sottofascicolo
3. Dossier
4. Serie
 - a. di Fascicoli
 - b. di Dossier
 - c. tipologica di documenti
5. Volume
 - a. di Fascicolo
 - b. di Serie di fascicoli
 - c. di Serie Tipologica di documenti
 - d. di Sottofascicolo

Tabella 18: dati di input createFolder

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	typeId	Identificativo della tipologia di folder da creare
1	principal	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS
1	properties	Insieme di property che caratterizzano il folder. Esiste un oggetto (estensione di FolderPropertiesType) specifico per ogni typeId utilizzabile
1	folderId	Identificativo ACARIS del folder padre per il folder che si vuole creare

Tabella 19: dati di output createFolder

Liv	Dato	Annotazione
1	objectId	Identificativo ACARIS del folder creato

8.7 L'operazione getContentStream

Permette di recuperare il 'content-stream' (un file in formato binario) associato all'oggetto specificato come parametro di tipo Documento Fisico (cioè i file associati ad esso) oppure di tipo Messaggio interscambio.

Nel caso di documento fisico, ad un singolo documento, possono essere collegati più di un contenuto ma l'applicativo client può richiedere uno solo di questi contenuti per singola invocazione di questa operazione, valorizzando opportunamente il dato di input streamId.

Nel secondo caso, si possono recuperare dati di un messaggio interscambio (segnatura, conferma ricezione, annullamento, aggiornamento, notifica eccezione) o di una ricevuta PEC. Se si vuole consultare un messaggio di interscambio della comunicazione presente nel sistema, l'operazione restituirà solo l'xml dell'oggetto. Se si vuole consultare un messaggio ricevuta PEC (messaggio scambiato dai server di posta certificati per certificare, appunto, le fasi di invio, ricezione e consegna di un messaggio di posta elettronica tra caselle certificate), l'operazione restituirà un file in formato eml (che rappresenta il vero e proprio messaggio di posta della ricevuta) contenente i dati identificativi della fase della comunicazione che rappresenta, oltre a indicazione temporale che indica quando è stata effettuata la fase in oggetto.

Tabella 20: dati di input getContentStream

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	documentId	Identificativo ACARIS del documento fisico, ottenibile tramite altre operazioni (ad es. query o getObjectRelationships)
1	streamId	Indicazione sulla tipologia dello stream richiesto <ul style="list-style-type: none"> • PRIMARY • SIGNATURE • TIMESTAMP • RENDITION_ENGINE • RENDITION_DOCUMENT

Tabella 21: dati di output getContentStream

Liv	Dato	Annotazione
1	AcarisContentStreamType[]	Lista di AcarisContentStreamType
-2	acarisContentStreamType	Singolo oggetto (vd Tabella 82)

8.8 L'operazione createAssociativeDocument

Crea un oggetto documento del tipo specificato e aggiunge il documento in un folder creando anche l'oggetto che permette l'associazione tra il documento ed il folder specificato.

Se vengono specificati un Folder, un tipo di Documento ed un tipo di Associazione non permessi il sistema genera una AcarisException con messaggio SER-E905. Se il folder padre (specificato da folderId) non accetta un contentStream, viene sollevata una AcarisException con messaggio SER-E903.

Tabella 22: dati di input createAssociativeDocument

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	typeId	Identificativo della tipologia di documento associativo da creare
1	associationTypeId	Identificativo della tipologia di associazione da creare con il folder parent identificato da folderId
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	propertiesDocumento	Elenco delle properties che caratterizzano il documento
1	propertiesAssociazione	Elenco delle properties che caratterizzano l'oggetto che modella l'associazione con il folder parent
1	folderId	Identificativo ACARIS del folder padre
1	contentStream	Oggetto che contiene i dati su un file fisico. Necessario per typeId = ContenutoFisicoPropertiesType

Tabella 23: dati di output createAssociativeDocument

Liv	Dato	Annotazione
1	objectId	Identificativo ACARIS del document creato

8.9 L'operazione createRelationship

Crea un oggetto Relationship del tipo specificato.

Se il source object o il target object non sono presenti nel sistema, viene sollevata una

AcarisException SER-E903.

Se il typeld del source o dell'oggetto target non è uno degli ObjectTypes permessi nel Tipo di Relationship, viene sollevata una AcarisException SER-E904.

Tabella 24: dati di input createRelationship

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	typeld	Identificativo del tipo di Relationship. Ammesso solo il valore DocumentCompositionPropertiesType
1	properties	Elenco delle properties che caratterizzano la relazione.
1	sourceObjectId	Identificativo ACARIS dell'oggetto sorgente della relazione
1	targetObjectId	Identificativo ACARIS dell'oggetto destinazione della relazione

Tabella 25: dati di output createRelationship

Liv	Dato	Annotazione
1	objectId	Identificativo ACARIS dell'oggetto creato

9 DocumentService

A fianco degli ACTA Services raccolti negli Object Services, sono messi a disposizione delle ECM application dei servizi appositamente creati per riunificare, in ottica di ottimizzazione, alcune delle operazioni di navigazione e creazione previste per la gestione del documento. Document Services mette a disposizione una versione alternativa delle operazioni contenute nel servizio Object Services: operazioni a grana “grossa” che effettuano, al loro interno, quelle operazioni di “orchestrazione” altrimenti demandate alle applicazioni client. Si tratta di “orchestrazione di funzionalità” e non di servizi: non vengono utilizzati direttamente i servizi applicativi ora disponibili ma le logiche di più basso livello da essi utilizzate.

9.1 L’operazione creaDocumento

Permette di inserire in archivio un documento all’interno di una struttura aggregativa esistente o in relazione padre-figlio con un altro documento (creazione di classificazione di un allegato).

Si può richiedere la creazione di un documento archivistico solo metadati oppure con la presenza di un file elettronico associato.

Questa operazione permette anche l’aggiunta di un ulteriore documento fisico legato ad una classificazione già presente in archivio e l’aggiunta di un ulteriore contenuto fisico (oggetto che contiene un file elettronico) ad un documento fisico esistente.

Tabella 26: dati di input creaDocumento

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l’operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l’operazione descritta in 7.1)
1	tipoOperazione	Indicazione della tipologia di creazione che si vuole effettuare
1	datiCreazione	Struttura dati contenente le informazioni necessarie alla creazione del documento

Tabella 27: dati di output creaDocumento

Liv	Dato	Annotazione
1	IdentificatoreDocumento	Struttura contenente l’objectId del documento creato, quello della classificazione (negli scenari di creazione della stessa) e l’indicazione della tipologia creata scelta nel seguente elenco: <ul style="list-style-type: none"> • documentoArchivistico.semplice • documentoArchivistico.db • documentoArchivistico.registro • documentoFisico • contenutoFisico

9.2 L'operazione trasformaDocumentoPlaceholderInDocumentoElettronico

Consente di trasformare un documento nato come placeholder in documento elettronico: sostanzialmente siamo in presenza di un documento ACTA creato senza un documento elettronico associato.

L'operazione permette di inserire uno o più contenuti associati ad un documento, con o senza firma, in base alle composizioni permesse (file singolo, con firma o marca detached, XML e foglio di stile associato), dando indicazioni sull'eventuale operazione di sbustamento del file firmato.

Tabella 28: dati di input trasformaDocumentoPlaceholderInDocumentoElettronico

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	classificazioneId	Identificativo ACARIS della classificazione "placeholder" di cui si richiede la trasformazione
1	registrazioneId	Nel caso in cui la classificazione sia protocollata, rappresenta l'identificativo ACARIS della registrazione di protocollo
1	infoRichiesta	Dati inerenti la richiesta
-2	tipoDocFisicoId	Identificativo del tipo di documento fisico
-2	composizioneId	Identificazione della composizione
-2	multiplo	Indica se il documento fisico è multiplo
-2	diventaElettronico	Valorizzare a TRUE se si vuole che il documento diventi puramente elettronico
-2	statoDiEfficaciaId	Identificativo dello stato di efficacia del documento elettronico
-2	rimandareOperazioneSbustamento	Nel caso in put sia fornito un file contenente delle informazioni di firma elettronica o marca temporale, indica se effettuare lo sbustamento o rimandare l'operazione
1	documentiFisici	Dati del documento elettronico da creare
-2	documentoFisico	Singolo documento fisico (oggetto di tipo DocumentoFisicoIRC, vd Tabella 32)

Tabella 29: dati di output trasformaDocumentoPlaceholderInDocumentoElettronico

Liv	Dato	Annotazione
1	IdentificazioniTrasformazione	Dati relativi ai documenti inseriti (uno per documento)
-2	identificazione	Singolo documento

--3	relazione	Dati che indicano la relazione tra il documento ed i contenuti fisici inseriti (oggetto di tipo RelationshipPropertiesType, vd Tabella 36)
--3	dataUltimoAggiornamento	Data di ultimo aggiornamento del documento inserito
--3	failedSteps	Nel caso in cui si siano verificati errori nella verifica della firma di uno dei file inseriti, indicazione del singolo passo che è andato in errore
---4	failedStep	Passo (oggetto di tipo FailedStepInfo, vd Tabella 35)

9.3 L'operazione getRappresentazioneDocumento

Il servizio utilizza un oggetto di tipo IdentificatoreDocumento per recuperare gli identificatori degli oggetti che concorrono a comporre un oggetto “documento” nel sistema.

Le informazioni restituite dipendono dal tipo di identificatore passato al servizio come “punto di partenza”.

Sono previsti i seguenti casi:

1. punto di partenza coincidente con documento archivistico (di qualsiasi tipo)
2. punto di partenza coincidente con documento fisico
3. punto di partenza coincidente con contenuto fisico

Poiché esiste una relazione di contenimento tra le tre tipologie di documento sopra elencate, il sistema restituisce, a seconda dei casi, i seguenti oggetti:

- DocumentoArchivisticoldMap
- DocumentoFisicoldMap
- ContenutoFisicoldMap

Tabella 30: dati di input getRappresentazioneDocumento

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	puntoDiPartenza	IdentificatoreDocumento a cominciare dal quale bisogna recuperare gli identificatori associati
1	recuperaAscendenza	Consente di indicare se restituire gli identificatori dei parent rintracciabili risalendo fino al documento archivistico.

Tabella 31: dati di output getRappresentazioneDocumento

Liv	Dato	Annotazione
1	MappaIdentificazioneDocumento	Oggetto che contiene le informazioni richieste. Restituita una delle sottoclassi previste a seconda della valorizzazione del parametro “puntoDiPartenza”. Le tipologie previste sono:

		<ul style="list-style-type: none"> • DocumentoArchivisticoldMap • DocumentoFisicoldMap • ContenutoFisicoldMap
--	--	--

9.4 Oggetti comuni

Di seguito sono riportate le indicazioni degli oggetti comuni a più operazioni dei Document services.

Tabella 32: DocumentoFisicoIRC

Liv	Dato	Annotazione
1	documentoArchivistico	Identificativo ACARIS del documento presente in ACTA
1	propertiesDocumentoFisico	Insieme di dati del documento elettronico (oggetto di tipo DocumentoFisicoPropertiesType, vd Tabella 80)
1	contenutiFisici	Elenco di oggetti che descrivono un contenuto fisico
-2	contenutoFisico	Singolo contenuto fisico (oggetto di tipo ContenutoFisicoIRC, vd Tabella 81)
1	azioniVerificaFirma	Elenco di oggetti per indicare come comportarsi se si presenta un errore in fase di verifica della firma del file in input
-2	azione	Singola azione per una delle fasi della verifica della firma (oggetto di tipo StepErrorAction, vd Tabella 34)

Tabella 33: ContenutoFisicoIRC

Liv	Dato	Annotazione
1	documentoFisico	Identificativo ACARIS del documento presente in ACTA
1	dataUltimoAggiornamentoDocumento	Data di ultimo aggiornamento del documento padre ACTA associato
1	propertiesContenutoFisico	Proprietà che descrivono un contenuto fisico (oggetto di tipo ContenutoFisicoPropertiesType, vd Tabella 81)
1	tipo	Tipo del contenuto fisico (oggetto tipo EnumStreamId)
1	stream	Dati relativi al file da associare al contenuto fisico, compreso il contenuto binario (oggetto di tipo AcarisContentStreamType, vd Tabella 82)
1	azioniVerificaFirma	Elenco di oggetti per indicare come comportarsi se si presenta un errore in fase di verifica della firma del file in input
-2	azione	Singola azione per una delle fasi della verifica della firma (oggetto di tipo StepErrorAction, vd Tabella 34)

Tabella 34: StepErrorAction

Liv	Dato	Annotazione
1	step	Indicazione del passo a cui si riferisce l'oggetto
1	action	Indicazione dell'azione da eseguire (oggetto di tipo EnumStepErrorAction)

Tabella 35: FailedStepInfo

Liv	Dato	Annotazione
1	fileName	Nome file per cui la verifica firma ha sollevato degli errori
1	failedSteps	Indicazione numerica del passo in errore

Tabella 36: RelationshipPropertiesType

Liv	Dato	Annotazione
1	relationType	Tipo di relazione presente tra gli oggetti di questa relazione
1	sourceId	Identificativo ACARIS dell'oggetto padre della relazione
1	targetId	Identificativo ACARIS dell'oggetto figlio della relazione
1	sourceType	Tipo di oggetto padre
1	targetType	Tipo di oggetto figlio

10 NavigationService

Per gli oggetti della classe base Folder, oltre agli object services di cui sopra, sono definiti servizi di navigazione della gerarchia di oggetti del repository.

Sono disponibili funzionalità di paging, utili per scorrere folder di dimensioni considerevoli, e di attraversamento di un sotto albero sino al livello di profondità richiesto, lo schema di attraversamento non è definibile dal client.

Anche per gli oggetti della classe base Relationship sono definiti servizi di navigazione, questi permettono di individuare tutte le relazioni, o loro sottoinsieme, aventi un determinato oggetto come source o target.

10.1 L'operazione getChildren

Ottiene la lista degli oggetti 'figli' contenuti in uno specifico folder. Sono restituite solo le proprietà selezionate per ogni oggetto restituito. Questa operazione non restituisce l'eventuale contenuto binario associato al documento.

Tabella 37: dati di input getChildren

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	folderId	Identificativo ACARIS dell'oggetto di cui si richiede il figlio
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	filter	Filtro da applicare alla ricerca (oggetto di tipo PropertyFilterType, vd Tabella 85)
1	maxItems	Valori da utilizzare per effettuare una paginazione dei risultati. <ul style="list-style-type: none"> maxItems indica il numero massimo di oggetti skipCount indica l'elemento da cui continuare nella restituzione degli oggetti trovati
	skipCount	

Tabella 38: dati di output getChildren

Liv	Dato	Annotazione
1	pagingResponse	Risultato della ricerca (oggetto di tipo PagingResponseType, vd Tabella 86)

10.2 L'operazione getFolderParent

Il servizio restituisce una lista di oggetti in relazione figlio-padre (antenato) con l'oggetto della classe base Folder (Classificazione, Gruppo Allegato, Documento Fisico) specificato come parametro.

Per ogni oggetto devono essere restituite le proprietà di identità dell'oggetto e del proprio padre per permettere un eventuale riordinamento da parte del client.

Tabella 39: dati di input getFolderParent

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	folderId	Identifica l'oggetto da cui si inizia la navigazione può solo essere un oggetto di tipo Folder. Se viene specificato un oggetto che non è di tipo Folder il sistema genera una AcarisException con il messaggio SERNAV-E001
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	filter	Vengono determinate le properties da restituire per ogni oggetto trovato durante la navigazione secondo quanto indicato in nell' algoritmo ACARIS_A01_Property_Filters

Tabella 40: dati di output getFolderParent

Liv	Dato	Annotazione
1	objectResponse	Risultato dell'operazione di ricerca, contenente le informazioni sul folder parent (oggetto di tipo ObjectResponseType, vd Tabella 88)

10.3 L'operazione getObjectParent

Il servizio restituisce una lista di oggetti in relazione figlio-padre (antenato) con l'oggetto di una classe base diversa da Folder (Documento, Contenuto fisico) specificato come parametro.


Per ogni oggetto devono essere restituite le proprietà di identità dell'oggetto e del proprio padre per permettere un eventuale riordinamento da parte dell'applicativo client.

Tabella 41: dati di input getObjectParent

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	objectId	Identificativo di un oggetto non di tipo Folder
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)

Tabella 42: dati di output getObjectParent

Liv	Dato	Annotazione
1	objectResponses[]	Elenco di oggetti di tipo objectResponse che contengono l'oggetto di tipo document specificato.

	<p align="center">DoQui – ACTA Servizi ACARIS Manuale Integrazione Fruitore</p>	<p align="right">ACTA-Acaris-MUT-002</p> <p align="right">Pag. 42 di 70</p>
--	--	---

-2	objectResponse	Singolo oggetto (oggetto di tipo ObjectResponseType, vd Tabella 88)
----	----------------	---

11 RelationshipService

Oltre i servizi raccolti nei Navigation Services, è messo a disposizione un servizio tramite che permette la navigazione dei legami tra un oggetto della classe DocumentoFisico ed uno della classe Document, necessario per ottenere i riferimenti necessari alla consultazione tramite ACTA Services dei content stream archiviati nel repository

11.1 L'operazione getObjectRelationship

Il servizio restituisce una lista di relazioni associate all'oggetto, della tipologia specificata con riferimento alla direzione specificata.

Tabella 43: dati di input getObjectRelationship

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	objectId	Identificativo dell'oggetto con cui la relazione è associata
1	typeId	Tipo Relazione, viene selezionata dal seguente elenco: <ul style="list-style-type: none"> HistoryModificheTecnichePropertiesType (non implementato) DocumentCompositionPropertiesType HistoryVecchieVersioniPropertiesType (non implementato) DocumentAssociationPropertiesType (non implementato)
1	direction	Tipo di direzione source o target. I valori possibili sono: <ul style="list-style-type: none"> Source Target Either
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	filter	Elenco delle properties

Tabella 44: dati di output getObjectRelationship

Liv	Dato	Annotazione
1	response	Rappresenta la lista delle relazioni trovate con le relative properties. Viene veicolata con un oggetto di tipo RelationshipPropertiesType.

12 OfficialBookService

Gli ACTA Services mettono a disposizione delle ECM application, un insieme di servizi per interagire con oggetti propri del modulo di gestione della protocollazione: creazione delle registrazioni di protocollo, navigazione dei legami tra gli oggetti del modulo e di recupero del dettaglio delle singole entità. Sono stati aggiunti dei servizi per la gestione della segnatura di una registrazione di protocollo.

12.1 L'operazione creaRegistrazione

L'operazione consente la creazione di una registrazione di protocollo.

Le tipologie di registrazione a disposizione sono:

- registrazione in arrivo
- registrazione in partenza
- registrazione interna

Tabella 45: dati di input creaRegistrazione

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	tipologiaCreazione	Indicazione del metodo di creazione da utilizzare <ul style="list-style-type: none"> • Protocollazione • ProtocollazioneDocumentoEsistente • ProtocollazioneDaSmistamento
1	infoRichiestaCreazione	Dati per la creazione della registrazione di protocollo
-2	registroid	Indicazione del contenitore in cui creare la registrazione: <ul style="list-style-type: none"> • registroid per indicazione puntuale del registro • aooProtocollanteld per utilizzare il registro legato ad una specifica AOO
	aoaProtocollanteld	
-2	senzaCreazioneSoggettiEsterni	Indica la volontà di non creare i soggetti ACTA per i corrispondenti esterni

Tabella 46: dati di output creaRegistrazione

Liv	Dato	Annotazione
1	IdentificazioneRegistrazione	
-2	registrazioneld	Identificativo ACARIS della registrazione creata

-2	numero	Numero della registrazione creata
-2	folderId	Identificativo ACARIS e tipologia del folder in cui è presente la classificazione legata alla registrazione
	folderType	
-2	dataUltimoAggiornam ento	Data di creazione della registrazione
-2	classificazioneld	Identificativo ACARIS della classificazione legata alla registrazione

Il dato di input infoRichiestaCreazione di tipo RegistrazioneRequest è utilizzato per veicolare i dati necessari per la creazione della registrazione. Questo è specializzato da sottoclassi specifiche per il metodo di creazione da utilizzare: Protocollazione, ProtocollazioneDocumentoEsistente, ProtocollazioneDaSmistamento.

Tabella 47: Protocollazione

Liv	Dato	Annotazione
1	folderId	Opzionale. Identificativo ACARIS oppure tipologia del folder in cui inserire la classificazione creata per la registrazione
	folderType	
1	registrazioneAPI	Metadati della registrazione da creare

Tabella 48: ProtocollazioneDocumentoEsistente

Liv	Dato	Annotazione
1	classificazioneld	Identificativo ACARIS della classificazione già presente in ACTA
1	registrazioneAPI	Metadati della registrazione da creare

Tabella 49: ProtocollazioneDaSmistamento

Liv	Dato	Annotazione
1	smistamentold	Identificativo ACARIS dello smistamento da collegare alla registrazione
1	registrazioneAPI	Metadati della registrazione da creare

Il dato di input di tipo RegistrazioneAPI indica il tipo di registrazione di protocollo di cui è richiesta la creazione (tra Arrivo, Partenza e Interna). Questo è specializzato dalle sottoclassi: RegistrazioneArrivo, RegistrazionePartenza, RegistrazioneInterna.

Tabella 50: RegistrazioneAPI

Liv	Dato	Annotazione
-----	------	-------------

1	tipoRegistrazione	Tipo di registrazione da creare <ul style="list-style-type: none"> • arrivo • partenza • interna
1	infoCreazione	Metadati
-2	protocollante	Dati per identificare il protocollante
--3	strutturald	Identificativo del nodo e della struttura in organigramma ACTA a cui sarà legata la registrazione
	nodold	
-2	numeroRegistrazione Precedente	In caso la registrazione sia legata ad una registrazione precedente, indicare numero ed anno
	annoRegistrazionePrecedente	
-2	registrazioneRiservata	Indica se la registrazione è riservata
-2	oggetto	Oggetto della registrazione. Utilizzato anche come oggetto della classificazione creata
-2	collocazioneCartacea	Collocazione cartacea della registrazione
-2	mezzoTrasmissivold	Identificativo del mezzo trasmissivo
-2	documentoRiservato	Indica se il documento è riservato
-2	classificazioneProposta	Stringa che indica l'indice archivistico della voce in cui si richiede la classificazione.
-2	dataDocumento	Data del documento della registrazione
-2	descrizioneAllegato	Opzionale. Elenco di descrizioni degli allegati, nel caso di creazione della classificazione (metodo: Protocollazione)
--3	descrizione	Descrizione dell'allegato usata come oggetto dell'allegato
-2	mittentiInterni	Elenco dei corrispondenti della registrazione di protocollo.
	destinatariInterni	
	destinatariEsterni	
--3	mittenteInterno	Singolo mittente/destinatario
	destinatarioInterno	
	destinatarioEsterno	

---4	corrispondente	Dati del corrispondente (oggetto di tipo InfoCreazioneCorrispondente)
-2	annotazione	Annotazione da inserire sulla registrazione
-2	forzareSeRegistrazioneSimile	Indica se effettuare la creazione in caso di presenza di una registrazione simile
-2	forzareSeRegistrazionePropostaInvalida	Indica se effettuare la creazione in caso di errore sulla indicazione della classificazione proposta
-2	forzareSePresenzaInviti	Indica se effettuare la creazione di inviti automatici in lettura sulla registrazione
-2	forzareSePresenzaDallInoltro	Indica se effettuare la creazione in caso di presenza di inoltri verso altra AOO

La classe RegistrazioneArrivo, diversamente da RegistrazionePartenza e RegistrazioneInterna , aggiunge alcuni dati supplementare rispetto il tipo base.

Tabella 51: RegistrazioneArrivo

Liv	Dato	Annotazione
1	infoDateArrivo	Informazioni sulla data di arrivo della registrazione
-2	dataRicezione	Data di ricezione della registrazione
-2	dataTimbroPostale	Data del timbro postale
1	infoProtocolloMittente	Se la registrazione è legata ad una registrazione del mittente, indica i dati di quella registrazione
-2	numero	Numero della registrazione mittente
-2	anno	Anno della registrazione mittente
-2	data	Data della registrazione mittente
1	mittentiEsterni	Elenco di mittenti esterni
-2	mittenteEsterno	Mittente
--3	corrispondente	Dati del corrispondente (oggetto di tipo InfoCreazioneCorrispondente)

Tabella 52: InfoCreazioneCorrispondente

Il dato di input di tipo InfoCreazioneCorrispondente contiene i dati del corrispondente (destinatario o mittente) da creare.

A partire dal mese di Luglio 2021 DoquiACTA è stato modificato per aderire alle linee guida AgID e questo dato di input ha subito delle variazioni che sono, però, retrocompatibili:

sono stati aggiunti dei dati (tutti opzionali) per distinguere le varie tipologie di corrispondenti (Pubblica Amministrazione, Persona Fisica e Giuridica) e dati di maggior dettaglio per la Pubblica Amministrazione.

Liv	Dato	Annotazione
1	denominazione	Denominazione del corrispondente (in alternativa a Nome e Cognome)
1	nome	Nome e Cognome del corrispondente. Devono essere valorizzati in alternativa alla denominazione. La concatenazione di cognome + " " + nome viene usata come Denominazione del corrispondente
1	cognome	
1	carica	Carica all'interno dell'organizzazione / AOO. Se valorizzata, viene concatenata alla Denominazione del corrispondente.
1	persona	Descrizione di una persona in riferimento a un corrispondente di tipo Persona Giuridica o Pubblica Amministrazione. Se valorizzata, viene concatenata alla Denominazione del corrispondente.
1	testoFoglioTrasmissione	
1	infoSoggettoAssociato	Indica un riferimento ad un soggetto tramite due sottoclassi: soggetto presente in anagrafica interna (RiferimentoSoggettoEsistente) oppure su fonte esterna (SoggettoEsterno)
1	tipologiaCorrispondente	Opzionale. Potrà contenere: - PA (Amministrazione), - AE (Amministrazione Estera), - PF (Persona Fisica), - PG (Persona Giuridica). Se non valorizzato verrà utilizzato il seguente default: PF se valorizzati nome e cognome, PG se valorizzata la denominazione
1	codiceFiscale	Opzionale. * Nel caso di PF conterrà il suo CF (pattern: "[A-Z]{6}[0-9LMNPQRSTUVWXYZ]{2}[ABCDEHLMRST][0-9LMNPQRSTUVWXYZ]{2}[A-Z][0-9LMNPQRSTUVWXYZ]{3}[A-Z]"). * Nel caso della PA conterrà il CF dell'amministrazione nel formato PIVA (pattern: "[0-9]{11}") * Nel caso di PG: - se "ditta individuale" -> conterrà il CF della persona titolare (pattern: "[A-Z]{6}[0-9LMNPQRSTUVWXYZ]{2}[ABCDEHLMRST][0-9LMNPQRSTUVWXYZ]{2}[A-Z][0-9LMNPQRSTUVWXYZ]{3}[A-Z]"). - altrimenti -> conterrà il CF della ditta nel formato PIVA (pattern: "[0-9]{11}") * Nel caso di AE il campo sarà ignorato.
1	PIVA	Opzionale.

		<p>* Nel caso di PG conterrà la PIVA della ditta nel formato PIVA (pattern: "[0-9]{11}")</p> <p>* Nel caso di PF, PA ed AE il campo sarà ignorato.</p> <p>NOTA RIEPILOGATIVA sui campi codiceFiscale e PIVA nel caso di PG:</p> <ul style="list-style-type: none"> - nel caso di "ditta individuale" avremo sempre sia il codiceFiscale (di 16 caratteri alfanumerici) che la PIVA (di 11 cifre) - negli altri casi avremo sempre sia il codiceFiscale che la PIVA di 11 cifre. <p>Normalmente saranno uguali, ma potranno essere diversi nel caso di trasferimento del domicilio fiscale in altra provincia (in questi casi viene attribuita una nuova PIVA)</p>
1	codiceIPAPA	<p>Opzionale.</p> <p>Ha senso solo per la tipologiaCorrispondente PA e in esso DEVE ESSERE indicato il codice dell'ente PA preso da IPA.</p> <p>Quindi se tipologiaCorrispondente="PA" il campo è OBBLIGATORIO.</p> <p>Se la tipologiaCorrispondente è diversa da PA il valore eventualmente inserito verrà ignorato.</p>
1	codiceIPAAOO	<p>Opzionale.</p> <p>Ha senso solo per la tipologiaCorrispondente PA e in esso verrà indicato il codice della Area Organizzativa Omogenea (AOO) preso da IPA.</p> <p>Quindi se tipologiaCorrispondente="PA" il campo è opzionale.</p> <p>Se la tipologiaCorrispondente è diversa da PA il valore eventualmente inserito verrà ignorato.</p>
1	codiceIPAUE	<p>Opzionale.</p> <p>Ha senso solo per la tipologiaCorrispondente PA e in esso verrà, indicato il codice dell'Unità Organizzativa (UO) preso da IPA.</p> <p>Quindi se tipologiaCorrispondente="PA" il campo è opzionale.</p> <p>Se la tipologiaCorrispondente è diversa da PA il valore eventualmente inserito verrà ignorato.</p>
1	indirizzoTelematicoPA	<p>Opzionali.</p> <p>I 3 campi di contatto possono essere utilizzati nel caso di tipologiaCorrispondente PA.</p> <p>I campi valorizzati verranno utilizzati per alimentare i contatti nella Segnatura e FORZARE l'invio via PEC (o ws) a quei precisi destinatari.</p> <p>Se la tipologiaCorrispondente è diversa da PA il valore eventualmente inserito in questi campi verrà ignorato.</p>
1	indirizzoTelematicoAOO	
1	indirizzoTelematicoUO	
1	denominazioneAmministrazione	<p>Opzionale.</p> <p>Ha senso solo per la tipologiaCorrispondente PA e AE:</p> <ul style="list-style-type: none"> - nel caso di PA, dovrà essere indicata la denominazione della pubblica amministrazione presa da IPA; - nel caso di AE, sarà la denominazione dell'amministrazione estera inserita "liberamente".

		Se non valorizzato, il sistema lo valorizzerà di default con la "denominazione" Se la tipologiaCorrispondente è diversa da PA e AE il valore eventualmente inserito verrà ignorato.
1	denominazioneUfficio	Opzionale. Ha senso solo per la tipologiaCorrispondente AE: in esso verrà indicato la denominazione dell'ufficio dell'Amministrazione Estera. Se la tipologiaCorrispondente è diversa da AE il valore eventualmente inserito verrà ignorato.
1	indirizzoTelematico	Opzionale. Da utilizzare solo nel caso di tipologiaCorrispondente = PF, PG, AE. Se valorizzato verrà utilizzato per alimentare i contatti nella Segnatura ed effettuare l'invio via PEC (ma anche via ws) a quel preciso destinatario. Se la tipologiaCorrispondente è uguale a PA il valore eventualmente inserito verrà ignorato.
1	richiestaConferma	Opzionale. Indica la volontà di ricevere un messaggio interscambio che conferma l'avvenuta protocollazione da parte del corrispondente. Se non valorizzato, il sistema lo valorizzerà di default con 'S'.

12.2 L'operazione query

Consente di effettuare delle ricerche sugli oggetti di competenza del modulo funzionale Official book: registri di protocollo e relative registrazioni, classificazioni e documenti legati a registrazioni, anagrafica interna dei corrispondenti, mezzi trasmissivi configurati, stato di apposizione del sigillo e stato di invio del messaggio. Questa operazione è esposta su più service con la medesima firma: fare riferimento a Tabella 8: dati di input query e Tabella 9: dati di output query per l'elenco dei dati di input e output dell'operazione.

12.3 L'operazione getProperties

Operazione di interrogazione puntuale degli oggetti del modulo OfficialBook di tipo *Properties*.

Tabella 53: dati di input getProperties

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	objectId	Identificatore ACARIS dell'oggetto di cui restituire i metadati
1	filter	Filtro utilizzato per indicare le proprietà richieste in output

Tabella 54: dati di output getProperties

Liv	Dato	Annotazione
1	objectResponse	Contiene i metadati richiesti (oggetto di tipo ObjectResponseType, vd Tabella 88)

12.4 L'operazione inviaSegnaturaRegistrazione

L'operazione tramite cui il fruitore richiede l'invio della segnatura informatica di una registrazione di protocollo. L'invocazione è puntuale e permette di richiedere l'invio per una registrazione di protocollo in partenza già presente in uno dei registri aperti legati alle AOO abilitate nel sistema. Un fruitore può invocare questa operazione solo per registrazioni di protocollo inserito dallo stesso fruitore tramite integrazione a servizi.

Data una registrazione di protocollo in partenza presente nel sistema, la richiesta di invio può essere effettuata se la segnatura di protocollo è stata creata e, se l'ente lo richiede, questa è stata sigillata elettronicamente. Il fruitore può richiedere che l'invio sia effettuato anche in assenza di sigillo.

È possibile indicare che il sistema prenda in carico la richiesta anche in assenza della segnatura di protocollo: in questo caso, sarà il sistema ad attendere che questa sia prodotta prima di proseguire con l'invio.

L'invio può essere effettuato solo tramite una delle caselle mail che sono state configurate in Doqui ACTA e per cui è abilitato l'utente utilizzato per richiamare l'operazione.

Il flusso logico si basa sui dati della registrazione in oggetto, oltre a quelli forniti in input dell'operazione stessa (come descritto nella *Tabella 55: dati di input inviaSegnaturaRegistrazione*)

- la lista dei destinatari (opzionale), che rappresenta un sottoinsieme dei destinatari della registrazione in oggetto e permette di effettuare l'invio solo ad alcuni di essi;
- la forzatura dell'invio anche senza sigillo, per permettere di gestire urgenze nel caso in cui si richiede l'invio senza attendere l'apposizione del sigillo elettronico (operazione asincrona per natura);
- presa in carico della richiesta anche se al momento la segnatura informatica per la registrazione di protocollo indicata non è stata creata dal sistema (valorizzando info.invioNonPrioritario = true).

Il servizio farà immediatamente dei controlli formali restituendo delle specifiche eccezioni in caso di errore (se ad esempio la registrazione indicata in input non è valida o se l'indirizzo telematico indicato come uno dei destinatari è formalmente errato); il sistema, quindi, restituirà l'objectId per le consultazioni successive sullo stato di invio e agirà in asincrono per portare a termine quanto richiesto.

A partire dalla versione 12.2 di ACTA, l'operazione permette al fruitore di richiedere la generazione e l'invio della 'copia di cortesia' del documento principale della registrazione di protocollo: questa non è altro che una copia del solo documento principale sulla quale è stampigliata una dicitura che riporta gli estremi della registrazione di protocollo a cui il documento afferisce, e una eventuale dicitura aggiuntiva (scelta dall'Ente mittente). La copia di cortesia può essere generata solamente su permesso dell'Ente ed inviata solamente a destinatari che non sono Pubbliche Amministrazioni. I destinatari che dovessero ricevere un messaggio contenente in allegato anche la copia di cortesia rileveranno una non conformità tra la segnatura di protocollo e l'insieme dei documenti allegati al messaggio.

Tabella 55: dati di input inviaSegnaturaRegistrazione

Liv	Dato	Annotazione
-----	------	-------------

1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	identificatoreRegistrazione	Obbligatorio. Identificativo (objectId) della registrazione di protocollo di cui si richiede l'invio della segnatura informatica.
1	mittente	<p>Facoltativo. È colui che invia la registrazione. Se non presente vale quanto indicato nella segnatura di protocollo. Composto da:</p> <ul style="list-style-type: none"> • identificatore del corrispondente (facoltativo) per appaiarlo al dato in segnatura. Se non presente, si suppone che sia presente un solo mittente • casella (e-mail) (facoltativo) Se non presente, si utilizza il relativo indirizzo e-mail del corrispondente mittente della registrazione in oggetto
1	destinatari	<p>Facoltativo. Sono i destinatari della comunicazione.</p> <p>Se non presente vale quanto indicato nella segnatura di protocollo. Altrimenti può corrispondere ad un sottoinsieme dei corrispondenti indicati nella registrazione.</p> <p>Questa lista può essere:</p> <ul style="list-style-type: none"> - vuota: sono utilizzati i dati indicati in segnatura - con tutti i destinatari, per completare e/o modificare i dati presenti in segnatura per l'invio (ad es. l'indirizzo e-mail in "casella") - con un sottoinsieme di destinatari, per escludere da una richiesta di invio alcuni di quelli presenti in segnatura. <p>È composto da:</p> <ul style="list-style-type: none"> • identificatore del corrispondente (obbligatorio) per appaiarlo al dato in segnatura • casella (e-mail) (facoltativo) Se non presente, si utilizza il relativo indirizzo e-mail del corrispondente destinatario della registrazione in oggetto • confermaRicezione per indicare al destinatario la volontà di ricevere un messaggio interscambio di conferma ricezione della segnatura informatica inviata • usaCanalePec: per indicare l'uso della posta certificata come canale di comunicazione per l'interscambio. Al momento è l'unico canale disponibile in Doqui ACTA (per cui il valore di questo dato è ininfluente) ma si prevede in futuro di implementare un canale di interscambio basato su servizi web • copiaCortesia (opzionale): utilizzato per richiedere l'invio della copia di cortesia, se l'ente ha abilitato questa opzione. Nel caso in cui questo parametro non è indicato, il servizio utilizza il valore di default impostato dall'ente. Si noti che la copia di cortesia non viene inviata in nessun caso a destinatari di tipo PA.
1	info	<p>Si tratta di informazioni aggiuntive per il servizio, tutte opzionali e isolate:</p> <ul style="list-style-type: none"> • invioMultiplo: permette di mandare comunicazioni singole ad ogni destinatario (si tratta del caso in cui non si vuole che i destinatari siano a

		<p>conoscenza della comunicazione della stessa segnatrice ad altre persone). I destinatari PA riceveranno la segnatrice originale nella quale si vedono tutti gli indirizzi e-mail, mentre gli altri (PF, PG) riceveranno una segnatrice “ridotta” contenente solo il loro indirizzo.</p> <ul style="list-style-type: none"> • forzaturaAssenzaSigillo: nel caso in cui l’ente sia configurato per sigillare la segnatrice, serve per dire al sistema che quella segnatrice va inviata anche se non è ancora stata sigillata: serve per casi di urgenza e di indisponibilità del servizio di sigillatura. Viene tracciata come forzatura al processo standard. Se l’ente non prevede il sigillo, il valore inserito qui viene ignorato. • invioNonPrioritario: permette al fruitore di inserire una richiesta di invio per una registrazione di protocollo la cui segnatrice è in fase di creazione. Il valore di default causa il rilancio di un errore gestito se la segnatrice non è presente. • copiaLavoro: riservato ad usi futuri
--	--	--

Tabella 56: dati di output inviaSegnatriceRegistrazione

Liv	Dato	Annotazione
1	identificatoreInvioSegnatrice	<p>Identificativo attraverso il quale, una volta che viene innescato l’invio, si può ottenere lo stato dello stesso attraverso la consultazione dell’oggetto query.</p> <p>L’invio da acaris sarà sempre asincrono: lo stato iniziale sarà quindi sempre una pianificazione dell’invio stesso, il sistema si occuperà poi di verificare se ci sono degli invii in attesa e inizierà a processarli fino ad ottenere un invio “avvenuto” o mancato.</p>

12.5 L’operazione ricercaDatiInterscambio

Questa operazione offre una vista sintetica di un flusso di interscambio e relativi messaggi scambiati, a livello di registrazione di protocollo o di singolo invio.

Nel caso in cui si faccia una ricerca a livello di registrazione (indicando come parametro di ingresso l’identificativo objectId della stessa), è possibile che il sistema risponda con più insiemi di dati (ognuno facente riferimento ad un flusso di interscambio) poiché il sistema permette di effettuare un invio segnatrice più volte per la stessa registrazione di protocollo.

Nello scenario più semplice di un interscambio tra AOO, in cui invio una segnatrice ad un unico destinatario ed entrambi utilizzano una casella di posta certificata, il flusso in oggetto avrà:

- la segnatrice inviata,
- la ricevuta PEC di accettazione (il server della posta certificata del mittente accetta di mandare il messaggio)
- la ricevuta PEC di consegna (il server della posta certificata del destinatario comunica di aver ricevuto il messaggio e di averlo consegnato alla casella pec destinataria)
- eventuali messaggi di interscambio: il destinatario può mandare la conferma di ricezione, una notifica di eccezione o altri messaggi definiti da AgID per l’interscambio tra AOO.

Se il messaggio viene mandato ad una casella NON PEC avrò solo la ricevuta di accettazione e non avrò quello che segue.

L’operazione in oggetto restituisce, per ogni messaggio elaborato e registrato nel sistema, un elenco di dati che lo descrivono, tra cui quello che da utilizzare come input dell’operazione *getContentStream* (vd 8.7).

Tabella 57: dati di input ricercaDatiInterscambio

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	identificatoreRegistrazione	ObjectId che identifica una registrazione di protocollo
1	identificatoreSegnatura	ObjectID che si può ottenere dalla query su RichiestaInvioSegnaturaView. Nel caso la registrazione di protocollo in oggetto è legata a più invii, specificando questo si ottengono i dati interscambio del singolo invio.
1	tipoMessaggio	Posso specificarlo per richiedere, come risposta, i dati dei messaggi di un tipo specifico che fanno parte del flusso di interscambio: Segnatura, RicevutaPEC, Conferma, Aggiornamento, Annullamento, NotificaEccezione
1	maxItems	È il numero di elementi per pagina (configurabile dall'utente purché inferiore al massimo consentito a livello di sistema). Se non specificato viene usato il valore di default. Trattandosi di una ricerca che potrebbe restituire molti risultati abbiamo optato per una ricerca paginata.
1	skipCount	Da quale elemento deve partire la ricerca. Se non specificato avremo i primi x risultati (partenza da pagina 0).

Tabella 58: dati di output ricercaDatiInterscambio

Il servizio restituirà come risposta, un elenco di oggetti di tipo datiInterscambio, che contiene tutti gli elementi necessari al fruitore, per ricostruire la situazione complessiva e per mettere in atto eventuali "politiche" di re-invio.

Liv	Dato	Annotazione
1	datiInterscambioRisposte	Elenco di oggetti trovati con la ricerca
-2	datoInterscambio	Singolo oggetto (oggetto di tipo DatoInterscambio, vd Tabella 88) che riepiloga tutte le informazioni principali sui messaggi scambiati tra mittente e destinatario (esplicitamente, con un'operazione dell'utente, o implicitamente, dai server di posta perché nel sistema delle PEC vengono inviate automaticamente delle comunicazioni di accettazione / consegna).

--3	objectIdMessaggio	Identificativo del singolo elemento, di qualunque tipo esso sia. È utilizzabile come input dell'operazione getContentStream (vd 8.7)
--3	objectIdMessaggioInviato	Valido solo per i messaggi di tipo ricevuta pec. Contiene l'objectIdMessaggio della segnatura a cui la ricevuta fa riferimento. Potrei avere più invii e quindi più segnature e avere i dati delle ricevute in pagine diverse. È utilizzabile come input dell'operazione getContentStream (vd 8.7)
--3	tipoMessaggio	Codice che indica se si tratta di Segnatura, Ricevuta PEC, Conferma Ricezione, Aggiornamento, Annullamento, Notifica Eccezione
--3	StatoPecInviata	Valido solo per i messaggi interscambio inviati da Doqui ACTA. Codice che indica lo stato del messaggio: Inviata, Consegnata, Accettata, Errore Consegna, Non Accettata
--3	oggettoMessaggio	Oggetto del messaggio di posta legato al messaggio scambiato in oggetto. Nota: nel caso di ricevuta PEC, l'oggetto è uguale a quello del messaggio PEC inviato a cui questa si riferisce con la preposizione di una dicitura che ne indica la tipologia (ad esempio, nel caso di ricevuta di tipo "accettazione", avrò "RICEVUTA ACCETTAZIONE: Protocollo...", mentre nel caso di ricevuta di tipo "consegna" avrò "RICEVUTA_CONSEGNA: Protocollo..."
--3	mittenteMessaggio	Indirizzo email (PEC) tramite il quale è stato inviato il messaggio in oggetto. Nota: nel caso di invio segnatura il mittente è colui che realmente ha innescato il processo; nel caso delle ricevute il verso è opposto: il mittente dell'invio segnatura diventerà il destinatario della ricevuta.
--3	destinatariMessaggio	Destinatari del messaggio in oggetto. Nel caso della ricevuta PEC si tratta del server di posta certificata che ha accettato il messaggio
--3	perContoDi	Valido solo per le ricevute PEC di consegna ed errore consegna: è l'indirizzo di consegna a cui il server di posta ha consegnato realmente il messaggio
--3	dataInvioMessaggio	Data e ora in cui è stato inviato il messaggio: viene valorizzato nel caso di messaggio interscambio
--3	dataRicezioneMessaggio	Data e ora in cui il mittente ha ricevuto il messaggio: viene valorizzato nel caso di ricevute PEC
--3	identificatoreRegistrazione	Identificativo (objectId) della registrazione di protocollo
-2	hasMoreItems	Nel caso di paginazione, indica la presenza di altri oggetti inerenti alla ricerca effettuata. Se la ricerca ha prodotto più elementi rispetto a quanto indicato da maxItems e hasMoreItems = true, è necessario gestire la paginazione. L'utente agirà su skipCount (il primo valore sarà numero elementi pagina precedente + 1).

12.6 L'operazione annullaRegistrazione

Questa operazione consente di annullare una registrazione. Le modifiche recenti apportate ad ACTA impediscono la modifica di una registrazione di protocollo, per cui l'unica soluzione in caso di errore è l'annullamento della registrazione errata. L'operazione potrebbe essere necessaria per molteplici motivi, ad esempio se ci si rende conto di non poter effettuare l'invio della segnatura informatica di protocollo per errori formali o di aver inserito delle indicazioni errate che riguardano i corrispondenti della registrazione o i documenti a cui questa afferisce.

L'operazione permette al fruitore di annullare esclusivamente le registrazioni di protocollo create dallo stesso tramite servizi Acaris.

Tabella 59: dati di input annullaRegistrazione

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	identificatoreRegistrazione	Identificativo (objectId) della registrazione di protocollo di cui si vuole richiedere l'annullamento
1	motivazione	Motivo dell'annullamento

Tabella 60: dati di output annullaRegistrazione

Liv	Dato	Annotazione
1	dataAnnullamento	Timestamp dell'operazione di annullamento (se vengono superati i controlli formali)

13 MultifilingServices

Per gli oggetti della classe base Folder, oltre agli Object Services di cui sopra, sono definiti servizi di mantenimento della gerarchia di oggetti del repository. Gli ACTA Services permettono di creare associazioni di un oggetto derivato dalla classe base Document ad uno o più oggetti (si supporta il multi-filing) derivati dalla classe base Folder.

Invece gli oggetti derivati dalla classe base Folder non possono essere associati a più di un Folder, causa il vincolo che impone di avere un solo padre, e ad un oggetto della propria discendenza per il vincolo di ciclicità dell'albero, per questo è stato previsto un servizio specifico.

13.1 L'operazione addAssociativeObjectToFolder

L'operazione permette di aggiungere una relazione padre-figlio all'oggetto specificato come parametro (un Document) associandolo ad un folder padre attraverso la creazione di un oggetto 'associativo' di tipo classificazione. Il folder padre è un oggetto che modella una struttura aggregativa (Fascicolo, Sottofascicolo, Dossier, Serie, Volume).

Il servizio permette di sfruttare le possibilità di multi-filing del repository (nodo figlio legato a più di un nodo padre) degli oggetti derivati dalla classe base Document.

Questa operazione non prevede valori di ritorno.

Tabella 61: dati di input addAssociativeObjectToFolder

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	objectId	Documento da aggiungere al folder. I tipi di oggetto ammessi sono: <ul style="list-style-type: none"> • DocumentoDBPropertiesType • DocumentoRegistroPropertiesType • DocumentoSemplicePropertiesType
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	folderId	Identificativo ACARIS del folder che conterrà il folder 'associativo'. I tipi di oggetto ammessi sono: <ul style="list-style-type: none"> • DossierPropertiesType • SottofascicoloPropertiesType • FascicoloRealeEreditatoPropertiesType • FascicoloRealeLiberoPropertiesType • FascicoloRealeContinuoPropertiesType • FascicoloRealeLegislaturaPropertiesType • FascicoloRealeAnnualePropertiesType • VolumeSerieTipDocPropertiesType • SerieTipologicaDocumentiPropertiesType

		<ul style="list-style-type: none"> VolumeSottoFascPropertiesType VolumeFascPropertiesType ClassificazionePropertiesType
1	associationPropertiesType	Amnesso solo il valore ClassificazionePropertiesType
1	associativeObjectProperties	Elenco delle properties che caratterizzano il documento

13.2 L'operazione aggiungiClassificazione

Il servizio costituisce un'evoluzione del servizio addAssociativeObjectToFolder: permette di aggiungere una relazione padre-figlio all'oggetto specificato come parametro.

Questa operazione introduce alcune innovazioni in merito alla gestione degli oggetti di tipo derivato dalla classe Document che sono già in relazione con l'oggetto in input del servizio (le classificazioni che possono trovarsi in relazione padre-figlio con quella da copiare, che sono considerate "allegate" alla prima) e la possibilità di prenotare l'esecuzione offline dell'operazione.

Quando il folder padre è un oggetto di tipo Classificazione, questo significa che si vuole creare un allegato ad altro documento. In questo caso, è necessario che il documento da allegare non abbia a sua volta allegati: in caso contrario il sistema restituisce un'eccezione veicolando il codice di errore SER_E_173.

Tabella 62: dati di input aggiungiClassificazione

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	classificazioneId	Identificativo ACARIS della classificazione di cui si vuole effettuare una copia
1	folderId	Identificativo ACARIS del folder che conterrà il folder 'associativo'. I tipi di oggetto ammessi sono: <ul style="list-style-type: none"> DossierPropertiesType SottofascicoloPropertiesType FascicoloRealeEreditatoPropertiesType FascicoloRealeLiberoPropertiesType FascicoloRealeContinuoPropertiesType FascicoloRealeLegislaturaPropertiesType FascicoloRealeAnnualePropertiesType VolumeSerieTipDocPropertiesType

		<ul style="list-style-type: none"> • SerieTipologicaDocumentiPropertiesType • VolumeSottoFascPropertiesType • VolumeFascPropertiesType • ClassificazionePropertiesType
1	params	Elenco di oggetti di tipo ItemType veicolati tramite un oggetto di tipo VarargsType da utilizzare come indicazioni per specificare il tipo di operazione da effettuare.
		addConAllegati <ul style="list-style-type: none"> • true, viene aggiunta una classificazione con gruppo allegati e se ci sono allegati questi vengono copiati • false, viene aggiunta una classificazione semplice senza allegati
		offlineAddRequest , simile a moveDocument. Se addConAllegati = true e la verifica del sistema individua la necessità di effettuare l'operazione offline <ul style="list-style-type: none"> • true, il sistema salva una prenotazione • false, viene sollevata un'eccezione, richiedendo di richiamare nuovamente l'operazione valorizzando offlineAddRequest = true
		numeroClassificazione la valorizzazione è obbligatoria nel caso di inserimento in un folder di tipo Serie tipologica di documenti (a numerazione libera): il sistema valorizzerà in questo modo il campo codice della classificazione. Per tutti gli altri tipi di Folder padre, il sistema valorizza in automatico il campo codice della classificazione.

Tabella 63: dati di output aggiungiClassificazione

Liv	Dato	Annotazione
1	objectId	Identificativo ACARIS della classificazione creata oppure, in caso di prenotazione di un'operazione di copia offline degli allegati, della prenotazione dell'esecuzione asincrona

Riepilogo del flusso di chiamate:

- se aggiungo una classificazione senza allegati mi viene restituito l'identificatore della classificazione
- se aggiungo una classificazione con allegati e richiedo l'esecuzione offline, se il sistema rileva che può essere effettuata una prenotazione, restituisce l'identificatore della prenotazione
- se si chiede l'esecuzione offline, per capire se l'identificatore restituito è una prenotazione bisogna invocare *query* con target moveDocumentOfflineView: se viene restituita eccezione SERQRY-E007 allora l'identificatore passato in input non è una prenotazione

14 ManagementService

Gli ACTA Services permettono di operare su oggetti del modello concettuale di ACTA ed in aggiunta su entità di business non direttamente sotto il controllo di tale modello ma comunque necessarie per l'ERMS. Si tratta delle entità del modulo BKO di ACTA richieste in fase di creazione di oggetti derivati da Document

14.1 L'operazione `getVitalRecordCode`

Questa operazione permette di ottenere l'elenco degli identificativi dei "gradi di validità" validi per un repository (Ente) che possono essere utilizzati per caratterizzare oggetti di tipo Folder (struttura aggregativa) e Document.

La vitalità è usata come metro di misura dell'importanza che un Folder ha in seno al repository.

Tabella 64: dati di input `getVitalRecordCode`

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)

Tabella 65: dati di output `getVitalRecordCode`

Liv	Dato	Annotazione
1	vitalRecordCodeType[]	<p>Elenco di oggetti contenente le property di ogni vital record code trovato:</p> <ul style="list-style-type: none"> • idVitalrecordCode : identificativo univoco del vital record code per il repository • descrizione: stringa che contiene la descrizione del vital record code • tempoDiVitalità: numero che contiene il tempo di vitalità in giorni del vital record code

14.2 L'operazione `addAnnotazioni`

Questa operazione permette di aggiungere un'annotazione ad un oggetto di tipo Folder (Fascicolo, Serie, Dossier, Volume, Sottofascicolo, Classificazione) o ad un documento archivistico.

L'annotazione inserita può essere formale o meno, in base alla valorizzazione dei parametri di ingresso.

Nel caso di aggiunta di annotazioni ad un documento, si distinguono due casi:

1. Applicazione classificazione corrente: il parametro `objectId` fa riferimento ad un oggetto di tipo Classificazione, per cui l'annotazione è posta solo su questa collocazione del documento.
2. Applicazione intero documento: il parametro `objectId` fa riferimento ad un oggetto di tipo documento archivistico (Documento Semplice, Documento Registro, Documento DB), per cui l'annotazione è posta su tutte le collocazioni (classificazioni) del documento all'interno del titolare ACTA (nel caso in cui un documento può essere copiato in diversi folder)

Tabella 66: dati di input addAnnotazioni

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	objectId	Identificativo ACARIS dell'oggetto a cui si vuole aggiungere un'annotazione
1	annotazione	Oggetto che trasmette le informazioni sull'annotazione <ul style="list-style-type: none"> • descrizione corrisponde al testo dell'annotazione • annotazioneFormale indica se l'annotazione ha valore formale

Tabella 67: dati di output addAnnotazioni

Liv	Dato	Annotazione
1	idAnnotazioneType	Identificativo dell'annotazione inserita

15 SubjectRegistryService

I servizi raccolti nei Subject Registry Services permettono di agire su oggetti delle classi specifiche di ACTA del modulo di gestione del protocollo: creazione di soggetti gestiti nell'anagrafica dei corrispondenti, loro aggiornamento e recupero delle informazioni di questa tipologia di oggetti.

15.1 L'operazione creaSoggetto

L'operazione consente la creazione di un Soggetto applicando, in quanto compatibili con una esposizione a servizi, le logiche già previste per questa funzionalità nel modulo PRT di Acta

Tabella 68: dati di input creaSoggetto

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	tipologiaCreazione	EnumTipologiaCreazioneSoggetto: il fruitore indica la tipologia di soggetto da creare: <i>SoggettoProvvisorio</i> o <i>SoggettoDefinitivo</i> .
1	infoRichiestaCreazione	SoggettoRequest: Individuato il tipo di soggetto da creare il parametro infoRichiestaCreazione consente di specificare, conformemente alle indicazioni contenute in tipologiaCreazione, le informazioni specifiche del soggetto da creare. Il parametro infoRichiestaCreazione è astratto e quindi ci si aspetta che venga valorizzato con una specifica sottoclasse dello stesso. In caso contrario il sistema solleva apposita eccezione.

Tabella 69: dati di output creaSoggetto

Liv	Dato	Annotazione
1	identificazioneSoggetto	Oggetto di tipo IdentificazioneSoggetto che contiene l'identificatore del soggetto creato, se si tratta di soggetto definitivo o provvisorio e un oggetto di tipo ChangeToken che può essere utilizzato in eventuali operazioni di aggiornamento.

15.2 L'operazione getProperties

Operazione di interrogazione puntuale degli oggetti del modulo SubjectRegistry di tipo *Properties*.

Tabella 70: dati di output getProperties

Liv	Dato	Annotazione
-----	------	-------------

1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	objectId	Identificativo ACARIS dell'oggetto di cui si richiedono i metadati
1	filter	Filtro utilizzato per indicare le proprietà richieste in output

Tabella 71: dati di output getProperties

Liv	Dato	Annotazione
1	objectResponse	Contiene i metadati richiesti (oggetto di tipo ObjectResponseType, vd Tabella 88)

15.3 L'operazione query

Consente di effettuare delle query sul modulo funzionale Acaris SubjectRegistry.

Si ricorda che il servizio viene esposto dai singoli moduli Acaris allo scopo di consentire, all'interno di un modulo, l'interrogazione delle sole entità afferenti al modulo considerato; allo stesso tempo, però, le logiche da applicare sono comuni a tutti i moduli e quindi anche le verifiche che il sistema effettua.

Questa operazione è esposta su più moduli con la medesima firma: fare riferimento a Tabella 8: dati di input query e Tabella 9: dati di output query per l'elenco dei dati di input e output dell'operazione.

15.4 L'operazione ricercaSoggetto

Si tratta di una figura particolare di query che, in primo luogo cerca sulla fonte esterna indicata dal fruitore e, in caso di mancanza di risultati, ripete la stessa sulle fonti interne tramite il normale impiego del servizio query.

Tabella 72: dati di input ricercaSoggetto

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	identitaDigitale	Si tratta di un'identità IRIDE ed è obbligatoria in quanto richiesta nell'interazione con il servizio che effettua la ricerca sulla fonte esterna ad ACTA.
1	codiceFiscale	Almeno uno tra codice fiscale e partita IVA dei due deve essere valorizzato
1	partitaIva	
1	fonteEsterna	Consente di specificare la fonte sulla quale effettuare la ricerca.

		Nella versione attuale del servizio si può specificare solo “GMS”.
1	flagPersonaGiuridica	Consente di specificare se il soggetto ricercato debba essere considerato o meno come persona giuridica

Tabella 73: dati di output ricercaSoggetto

Liv	Dato	Annotazione
1	anagraficaGenerica	Oggetto di tipo AnagraficaGenerica che contiene le informazioni sul soggetto richiesto (nome, cognome, cf, partita iva, denominazione)

15.5 L'operazione ricercaSoggettoDaFonteEsterna

Si tratta di un servizio generico di ricerca di un soggetto su fonte esterna ad ACTA.

Tabella 74: dati di input ricercaSoggettoDaFonteEsterna

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	identitaIride	Si tratta di un'identità IRIDE ed è obbligatoria in quanto richiesta nell'interazione con il servizio che effettua la ricerca sulla fonte esterna ad ACTA.
1	criteri	Indicano i criteri necessari per indirizzare la tipologia di ricerca e il servizio esterno da invocare

Tabella 75: dati di output ricercaSoggettoDaFonteEsterna

Liv	Dato	Annotazione
1	anagraficaSoggetto[]	Lista di soggetti trovati nell'anagrafica esterna

15.6 L'operazione getPropertiesMassive

Nell'object model Acaris sono definiti alcuni oggetti con properties a cardinalità multipla. Si tratta essenzialmente delle relazioni con classi esterne con cardinalità 0..*. In questi casi, i servizi di navigazione e ricerca valorizzano properties di questo tipo con gli identificatori degli oggetti relazionati. Si rende necessario, quindi, un'ulteriore operazione di recupero del dettaglio di ogni singola entità considerata.

Allo scopo di facilitare le operazioni di recupero ed ottimizzare le prestazioni complessive del sistema, in ogni modulo funzionale Acaris, viene messo a disposizione un servizio massivo di recupero delle properties.

Tabella 76: dati di input `getPropertiesMassive`

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	identificatori	Lista di identificatori ACARIS di cui restituire il dettaglio
1	filter	Filtro utilizzato per indicare le proprietà richieste in output

Tabella 77: dati di output `getPropertiesMassive`

Liv	Dato	Annotazione
1	objectResponseType[]	
-2	objectResponseType	Singolo oggetto (oggetto di tipo ObjectResponseType, vd Tabella 88)

16 SMSService

16.1 L'operazione query

Servizio analogo a quello presente nei Object Service. Permette di accedere alle proprietà degli oggetti di competenza del modulo di gestione degli smistamenti.

Questa operazione è esposta su più service con la medesima firma: fare riferimento a Tabella 8: dati di input query e Tabella 9: dati di output query per l'elenco dei dati di input e output dell'operazione.

16.2 L'operazione creaSmistamento

Consente di creare un oggetto di tipo Smistamento. Un oggetto di questo tipo permette, per esempio, di poter rendere possibile l'accesso ad un Document ad utenti dell'organigramma che non ne avrebbero accesso in base alla loro collocazione o di "assegnare" la competenza di un Document ad un determinato gruppo di utenti che condividono la stessa collocazione in organigramma.

Tabella 78: dati di input creaSmistamento

Liv	Dato	Annotazione
1	repositoryId	Identificativo del repository (Ente) in ACTA (ottenuto invocando l'operazione descritta in 6.1)
1	principalId	Token applicativo che identifica un utente utilizzato per invocare i servizi ACARIS (ottenuto invocando l'operazione descritta in 7.1)
1	mittente	È un utente fisico.
1	destinatari	Può essere una lista di utenti oppure nodi organizzativi. Può essere anche un destinatario per conoscenza. Se richiesto dalla tipologia di smistamento viene valorizzato l'identificatore del fascicolo temporaneo.
1	oggettiSmistati	Per ogni invocazione dell'operazione è permessa la presenza di una sola classificazione, quindi questa lista può contenere un solo elemento. Nel caso in cui lo smistamento abbia origine da una registrazione di protocollo (creata dal modulo Protocollo), bisogna valorizzare anche l'identificatore della registrazione di protocollo
1	infoCreazione	Oggetto che contiene una serie di informazioni necessarie in fase di creazione di uno smistamento <ul style="list-style-type: none"> tipoSmistamento: indica la tipologia di smistamento da creare. Il valore viene ricavato tramite navigazione o query puntuale sul QueryableObject di riferimento.

Tabella 79: dati di output creaSmistamento

Liv	Dato	Annotazione
1	idSmistamento	Identificatore dello smistamento creato. Corrisponde all'identificatore dello SmistamentoMittente creato

17 Oggetti comuni a tutti i servizi ACARIS

Di seguito sono riportate le indicazioni degli oggetti comuni a più servizi esposti.

Tabella 80: DocumentoFisicoPropertiesType

Liv	Dato	Annotazione
1	descrizione	Descrizione del documento fisico

Tabella 81: ContenutoFisicoPropertiesType

Liv	Dato	Annotazione
1	contentStreamFileName	Nome del file associato al contenuto fisico
1	contentStreamLength	Dimensione (in byte) del file associato al contenuto fisico
1	contentStreamMimeType	Mime type del file associato al contenuto fisico
1	sbustamento	Indica se il contenuto fisico deve essere sbustato
1	modificabile	Indica se il file associato al contenuto fisico è di tipo modificabile

Tabella 82: AcarisContentStreamType

Liv	Dato	Annotazione
1	streamMTOM	Oggetto che veicola il contenuto binario del file in modalità MTOM
1	filename	Nome del file associato
1	length	Lunghezza del file associato
1	mimeType	Mime type del file associato
1	nodeUID	Identificativo del contenuto fisico presente su ACTA

Tabella 83: QueryConditionType

Liv	Dato	Annotazione
1	propertyName	Nome della property
1	operator	Operatore di confronto
1	value	Valore della property

Tabella 84: NavigationConditionInfoType

Liv	Dato	Annotazione
1	parentNodeId	Identificativo ACTA dell'oggetto che rappresenta il nodo radice dell'albero in cui effettuare la ricerca
1	limitToChildren	Indica se la ricerca deve essere effettuata per i figli diretti di parentNode o per tutta la sua discendenza

Tabella 85: PropertyFilterType

Liv	Dato	Annotazione
1	filterType	Tipo di filtro da utilizzare <ul style="list-style-type: none"> • ALL, nessun filtro • NONE, insieme minimo delle proprietà • LIST, restituire solo le property indicate in propertyList
1	propertyList	Elenco di property di cui si richiede la restituzione
-2	property	Singola property (oggetto di tipo QueryNameType, vd Tabella 87)

Tabella 86: PagingResponseType

Liv	Dato	Annotazione
1	objects	Elenco di oggetti trovati con la ricerca
-2	object	Singolo oggetto (oggetto di tipo ObjectResponseType, vd Tabella 88)
--3	objectId	Identificativo ACARIS dell'oggetto
--3	properties	Elenco di property dell'oggetto restituito
---4	property	Singola property (oggetto di tipo PropertyType, vd Tabella 89)
1	hasMoreItems	Nel caso di paginazione, indica la presenza di altri oggetti inerenti alla ricerca effettuata

Tabella 87: QueryNameType

Liv	Dato	Annotazione
1	className	Nome della classe di riferimento della property

1	propertyName	Nome della property
---	--------------	---------------------

Tabella 88: ObjectResponseType

Liv	Dato	Annotazione
1	objectId	Identificativo ACARIS dell'oggetto
1	properties	Elenco di property dell'oggetto restituito (oggetto di tipo PropertyType, vd Tabella 89)

Tabella 89: PropertyType

Liv	Dato	Annotazione
1	queryName	Dati identificativi della property (oggetto di tipo QueryNameType, vd Tabella 87)
1	value	Valore della property

Tabella 90: SimpleResponseType

Liv	Dato	Annotazione
1	objectId	Identificativo ACARIS di un oggetto esistente
1	changeToken	Indicazione temporale dell'ultima modifica effettuata sull'oggetto