



Archimista 3.1

*Istruzioni comandi Batch
per le funzioni di
amministrazione*

Versione 1.0

25/02/2019

Indice

1 Istruzioni per i Comandi batch	4
1.1 ESTRAZIONE EAD Istruzioni per esportazioni batch formati ICAR-IMPORT	4
1.2 Istruzioni per esportazioni batch formati CAT-SAN e METS-SAN	6
1.3 Indicazioni sul codice di esportazione in formato CAT-SAN e METS-SAN	9
1.3.1 Caso 1: Export del singolo elemento	9
1.3.2 Caso 2: Export degli elementi correlati all'elemento prescelto	9
1.3.3 Caso 3: Export degli oggetti digitali	10
1.4 Installazione versione server	11
1.4.1 Metadata.rb	11
1.4.2 Configurazione_dl.rb	12
2. Caricamento massivo oggetti digitali	14

Storia delle Revisioni

Rev.	Data revisione	Autore	Descrizione
1.0	25/02/2019	Costantino Landino	Inserimento contenuti

1. Istruzioni per i comandi batch

1.1 Istruzioni per esportazioni batch formati ICAR-IMPORT

N.B.: il rake utilizza la virgola come riconoscitore degli argomenti da passare alla procedura. Per tale motivo la query, qualora contenesse delle virgole, va inviata con il simbolo "\" prima delle virgole per consentire alla procedura di riconoscere la virgola come parte dell'argomento stesso.

ESTRAZIONE EAD3 per fondi:

1) RAILS_ENV=production rake ead:build_xml[fonds,"SELECT DISTINCT f.* from rel_project_fonds r\, fonds f\, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=2 and trashed is false and f.ancestry_depth=0"] Consente di estrarre tutti i fondi di un progetto (cambiare id in p.id=2)

2) RAILS_ENV=production rake ead:build_xml[fonds,"SELECT DISTINCT * from fonds where id IN (1\,2\,3) and ancestry_depth=0"] Consente di estrarre tutti i fondi dei quali si conosce l'id e sono di primo livello.

3) RAILS_ENV=production rake ead:build_xml[fonds,"SELECT DISTINCT * from fonds where ancestry_depth=0"] Consente di estrarre tutti i fondi di primo livello presenti nel database.

ESTRAZIONE EAD3 per fonti/strumenti di ricerca:

1) RAILS_ENV=production rake ead:build_xml[sources,"SELECT * from sources where id IN (SELECT DISTINCT source_id FROM rel_source_fonds WHERE fond_id IN (SELECT DISTINCT f.id from rel_project_fonds r\, fonds f\, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=2 and trashed is false and f.ancestry_depth=0))"] Consente di estrarre tutte le fonti associate ai fondi di un progetto (cambiare id in p.id=2)

2) RAILS_ENV=production rake ead:build_xml[sources,"SELECT DISTINCT * from sources where id IN (1\,2\,3)"] Consente di estrarre tutte le fonti dei quali si conosce l'id.

3) RAILS_ENV=production rake ead:build_xml[sources,"SELECT DISTINCT * from sources"] Consente di estrarre tutte le fonti presenti nel database.

ESTRAZIONE EAD3 per unità:

- 1) RAILS_ENV=production rake ead:build_xml[units,"SELECT DISTINCT * from units where id IN (1,2,3)"] Estrazione unità con id specificato.
- 2) RAILS_ENV=production rake ead:build_xml[units,"SELECT DISTINCT * from units"] Estrazione di tutte le unità.
- 3) RAILS_ENV=production rake ead:build_xml[units,"SELECT DISTINCT * from units WHERE root_fond_id = 678 AND ancestry_depth = 0"] Estrae le unità radice relative al fondo con id 678.
- 3) RAILS_ENV=production rake ead:build_xml[units,"SELECT DISTINCT * from units WHERE root_fond_id = 678"] Estrae le unità relative al fondo con id 678.

Nel file di ogni unità sono inserite anche le unità figlie, le figlie delle figlie e così via.

ESTRAZIONE EAC-CPF per soggetti produttori:

- 1) RAILS_ENV=production rake ead:build_xml[creators,"SELECT * from creators where id IN (SELECT DISTINCT creator_id FROM rel_creator_fonds WHERE fond_id IN (SELECT DISTINCT f.id from rel_project_fonds r, fonds f, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=2 and trashed is false and f.ancestry_depth=0))"] Consente di estrarre tutti i soggetti produttori associati ai fondi di un progetto (cambiare id in p.id=2)
- 2) RAILS_ENV=production rake ead:build_xml[creators,"SELECT DISTINCT * from creators where id IN (1,2,3)"] Consente di estrarre tutti i soggetti produttori dei quali si conosce l'id.
- 3) RAILS_ENV=production rake ead:build_xml[creators,"SELECT DISTINCT * from creators"] Consente di estrarre tutti i soggetti produttori presenti nel database.

ESTRAZIONE EAC-CPF per profili istituzionali:

- 1) RAILS_ENV=production rake ead:build_xml[institutions,"SELECT DISTINCT * from institutions where id IN (1,2,3)"] Consente di estrarre tutti i profili istituzionali dei quali si conosce l'id.
- 2) RAILS_ENV=production rake ead:build_xml[institutions,"SELECT DISTINCT * from institutions"] Consente di estrarre tutti i profili istituzionali presenti nel database.

ESTRAZIONE EAC-CPF per schede anagrafiche:

- 1) `RAILS_ENV=production rake ead:build_xml[anagraphics,"SELECT * FROM rel_unit_anagraphics WHERE unit_id IN (1,2,3);"]` Consente di estrarre tutte le schede anagrafiche associate alle unità con id 1, 2 o 3.
- 2) `RAILS_ENV=production rake ead:build_xml[anagraphics,"SELECT * FROM rel_unit_anagraphics;"]` Consente di estrarre tutte le schede anagrafiche associate alle unità.

ESTRAZIONE SCONS2 per soggetti conservatori:

- 1) `RAILS_ENV=production rake ead:build_xml[custodians,"SELECT * from custodians where id IN (SELECT DISTINCT custodian_id FROM rel_custodian_fonds WHERE fond_id IN (SELECT DISTINCT f.id from rel_project_fonds r, fonds f, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=2 and trashed is false and f.ancestry_depth=0))"]` Consente di estrarre tutti i soggetti conservatori associati ai fondi di un progetto (cambiare id in p.id=2)
- 2) `RAILS_ENV=production rake ead:build_xml[custodians,"SELECT DISTINCT * from custodians where id IN (1,2,3)"]` Consente di estrarre tutti i soggetti conservatori dei quali si conosce l'id.
- 3) `RAILS_ENV=production rake ead:build_xml[custodians,"SELECT DISTINCT * from custodians"]` Consente di estrarre tutti i soggetti conservatori presenti nel database.

1.1. Istruzioni per esportazioni batch formati CAT-SAN e METS-SAN

Indicazioni generali

Eseguire il comando `RAILS_ENV=production rake san:build_xml[type]` (type può essere: fonds, creators, custodians, sources). La procedura guidata si avvia a schermo e verrà indicata alla fine la posizione dell'archivio zip generato contenente i files in formato CAT-SAN

eseguire il comando `RAILS_ENV=production rake san:build_mets`. La procedura guidata si avvia a schermo e verrà indicata alla fine la posizione dell'archivio zip generato contenente i files in formato METS-SAN

Se l'archivio estratto è di grosse dimensioni, vengono generati archivi parziali di minori dimensioni

`RAILS_ENV=production rake san:build_xml[fonds] "SELECT DISTINCT f.* from rel_project_fonds r, fonds f, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=1 and trashed is false and f.ancestry_depth=0"`

N.B.: il rake utilizza la virgola come riconoscitore degli argomenti da passare alla procedura. Per tale motivo la query, qualora contenesse delle virgole, va inviata con il simbolo "\" prima delle virgole per consentire alla

procedura di riconoscere la virgola come parte dell'argomento stesso.

ESTRAZIONE SAN per fondi:

1) RAILS_ENV=production rake san:build_xml[fonds,"SELECT DISTINCT f.* from rel_project_fonds r\, fonds f\, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=2 and trashed is false and f.ancestry_depth=0"] Consente di estrarre tutti i fondi di un progetto (cambiare id in p.id=2)

2) RAILS_ENV=production rake san:build_xml[fonds,"SELECT DISTINCT * from fonds where id IN (1\,2\,3) and ancestry_depth=0"] Consente di estrarre tutti i fondi dei quali si conosce l'id e sono di primo livello.

3) RAILS_ENV=production rake san:build_xml[fonds,"SELECT DISTINCT * from fonds where ancestry_depth=0"] Consente di estrarre tutti i fondi di primo livello presenti nel database.

ESTRAZIONE SAN per soggetti produttori:

1) RAILS_ENV=production rake san:build_xml[creators,"SELECT * from creators where id IN (SELECT DISTINCT creator_id FROM rel_creator_fonds WHERE fond_id IN (SELECT DISTINCT f.id from rel_project_fonds r\, fonds f\, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=2 and trashed is false and f.ancestry_depth=0))"] Consente di estrarre tutti i soggetti produttori associati ai fondi di un progetto (cambiare id in p.id=2)

2) RAILS_ENV=production rake san:build_xml[creators,"SELECT DISTINCT * from creators where id IN (1\,2\,3)"] Consente di estrarre tutti i soggetti produttori dei quali si conosce l'id.

3) RAILS_ENV=production rake san:build_xml[creators,"SELECT DISTINCT * from creators"] Consente di estrarre tutti i soggetti produttori presenti nel database.

ESTRAZIONE SAN per soggetti conservatori:

1) RAILS_ENV=production rake san:build_xml[custodians,"SELECT * from custodians where id IN (SELECT DISTINCT custodian_id FROM rel_custodian_fonds WHERE fond_id IN (SELECT DISTINCT f.id from rel_project_fonds r\, fonds f\, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=2 and trashed is false and f.ancestry_depth=0))"] Consente di estrarre tutti i soggetti conservatori associati ai fondi di un progetto (cambiare id in p.id=2)

2) RAILS_ENV=production rake san:build_xml[custodians,"SELECT DISTINCT * from custodians where id IN (1\,2\,3)"] Consente di estrarre tutti i soggetti conservatori dei quali si conosce l'id.

3) RAILS_ENV=production rake san:build_xml[custodians,"SELECT DISTINCT * from custodians"] Consente di estrarre tutti i soggetti conservatori presenti nel database.

ESTRAZIONE SAN per fonti/strumenti di ricerca:

1) RAILS_ENV=production rake san:build_xml[sources,"SELECT * from sources where id IN (SELECT DISTINCT source_id FROM rel_source_fonds WHERE fond_id IN (SELECT DISTINCT f.id from rel_project_fonds r\, fonds f\, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=2 and trashed is false and f.ancestry_depth=0))"] Consente di estrarre tutte le fonti associate ai fondi di un progetto (cambiare id in p.id=2)

2) RAILS_ENV=production rake san:build_xml[sources,"SELECT DISTINCT * from sources where id IN (1\,2\,3)"] Consente di estrarre tutte le fonti dei quali si conosce l'id.

3) RAILS_ENV=production rake san:build_xml[sources,"SELECT DISTINCT * from sources"] Consente di estrarre tutte le fonti presenti nel database.

ESTRAZIONE METS-SAN

1) RAILS_ENV=production rake san:build_mets["SELECT * FROM units WHERE id IN (SELECT distinct attachable_id FROM digital_objects WHERE attachable_type = 'Unit' AND asset_content_type LIKE 'image%') AND root_fond_id IN (SELECT DISTINCT f.id from rel_project_fonds r\, fonds f\, projects p where p.id=r.project_id and f.id=r.fond_id and f.published is true and p.id=2 and trashed is false and f.ancestry_depth=0) ORDER BY sequence_number"] Consente di estrarre tutti gli oggetti digitali associati alle unità inserite in fondi (sia di primo livello che successivi) di un progetto (cambiare id in p.id=2)

Quanto detto in precedenza per l'estrazione SAN vale anche per l'estrazione EAD ed EAC-CPF, con l'unica differenza nella chiamata del rake come da esempio:

RAILS_ENV=production rake ead:build_xml[fonds,"SELECT DISTINCT * from fonds where ancestry_depth=0"]

1.2. **Indicazioni sul codice di esportazione in formato CAT-SAN e METS-SAN**

Le possibili combinazioni implementate all'interno della pagina di esportazione di Archimista riguardo il formato CAT-SAN e METS-SAN sono le seguenti:

- 1) Export del singolo elemento (solo complesso, solo produttore, solo conservatore, sola fonte). Viene prodotto un singolo file xml in un archivio zip relativo all'entità scelta per l'export
- 2) Export dell'elemento e delle entità correlate. Vengono prodotti i file xml dell'elemento selezionato e delle entità correlate all'interno di un archivio zip.
- 3) Export comprensivo di oggetti digitali. Viene aggiunto il file xml degli oggetti digitali, se esistenti, in formato METS-SAN relativo all'elemento scelto.

1.3.1 Caso 1: Export del singolo elemento

- Se selezionato l'export di un fondo, viene creato il file xml "data-fond.xml" attraverso il codice contenuto in lib/views/fonds/_fonds.xml.builder. La variabile "records" contiene il fondo prescelto.
- Se selezionato l'export di un soggetto conservatore, viene creato il file xml "data-custodian.xml" attraverso il codice contenuto in lib/views/custodians/_custodians.xml.builder. La variabile "records" contiene il soggetto conservatore prescelto.
- Se selezionato l'export di un soggetto produttore, viene creato il file xml "data-creator.xml" attraverso il codice contenuto in lib/views/creators/_creators.xml.builder. La variabile "records" contiene il soggetto produttore prescelto, "fond_ids" è un array con gli id di tutti i fondi "root" presenti in Archimista. I fondi in relazione vengono estratti all'interno del file lib/views/creators/_creator.xml.builder.
- Se selezionato l'export di una fonte, viene creato il file xml "data-source.xml" attraverso il codice contenuto in lib/views/sources/_sources.xml.builder. La variabile "records" contiene la fonte prescelta.

1.3.2 Caso 2: Export degli elementi correlati all'elemento prescelto

- Se selezionato l'export di un fondo, viene creato il file xml "data-fond.xml" attraverso il codice contenuto in lib/views/fonds/_fonds.xml.builder. La variabile "records" contiene il fondo prescelto. In aggiunta vengono creati gli elementi relazionati al fondo prescelto, ovvero tutti i conservatori di tale fondo. Questi verranno inseriti in un file xml "data-custodian.xml" attraverso un "ciclo for" che analizza gli id di tutti i conservatori correlati. Oltre ai conservatori, verranno inseriti anche i produttori correlati al fondo. I soggetti produttori inseriti nel file xml "data-creator.xml" sono quelli relazionati al fondo e, in aggiunta, quei produttori che risultano a loro volta produttori dei soggetti relazionati al fondo. Infine vengono inserite tutte le fonti relazionate al fondo all'interno del file xml "data-source.xml".
- Se selezionato l'export di un soggetto conservatore, viene creato il file xml "data-custodian.xml" attraverso il codice contenuto in lib/views/custodians/_custodians.xml.builder. La variabile "records" contiene il soggetto conservatore prescelto. In aggiunta viene creato il file xml "data-fond.xml" contenente tutti i fondi

“root” correlati al soggetto conservatore. Inoltre viene creato il file xml “data-creator.xml” contenente tutti i soggetti produttori e i relativi produttori collegati a tutti i fondi a loro volta collegati al soggetto conservatore prescelto. Le fonti non sono incluse nell’export.

- Se selezionato l’export di un soggetto produttore, viene creato il file xml “data-creator.xml” attraverso il codice contenuto in lib/views/creators/_creators.xml.builder. La variabile “records” contiene il soggetto produttore prescelto, “fond_ids” è un array con gli id di tutti i fondi “root” presenti in Archimista. I fondi in relazione vengono estratti all’interno del file lib/views/creators/_creator.xml.builder. In aggiunta viene creato il file xml “data-fond.xml” contenente tutti i fondi “root” correlati al soggetto produttore. I soggetti conservatori e le fonti non sono inclusi nell’export.
- Se selezionato l’export di una fonte, viene creato il file xml “data-source.xml” attraverso il codice contenuto in lib/views/sources/_sources.xml.builder. La variabile “records” contiene la fonte prescelta. In aggiunta viene creato il file xml “data-fond.xml” contenente tutti i fondi “root” correlati al soggetto produttore. I soggetti conservatori e i soggetti produttori non sono inclusi nell’export.

1.3.3 Caso 3: Export degli oggetti digitali

- In associazione ai casi precedenti, è possibile esportare in formato METS-SAN gli oggetti digitali correlati agli elementi prescelti per l’esportazione. In particolare, quando viene esportato un fondo o viene selezionato l’export completo di un’altra entità e di conseguenza i fondi ad esso collegati, la procedura verifica l’esistenza ed estrae tutte le unità relazionate ai fondi “root” da esportare. Viene creato il file xml “data-digital-object.xml” attraverso il codice contenuto in lib/views/digital_objects/_digital_objects.xml.builder. La variabile “records” contiene tutte le unità precedentemente estratte.

1.3. *File di configurazione*

1.3.1. *Metadata.rb*

Esempio di file di configurazione per i tracciati CAT-SAN

Application metadata

APP_NAME = "Archimista"

APP_VERSION = "3.0.2"

APP_STATUS = ""

AEF_COMPATIBLE_VERSIONS = [110, 120, 121, 200, 210, 211, 220, 300, 301, 302]

APP_CREATOR = "INGLOBA360 srl"

APP_EDITION = "server" # server | standalone | hub

IM_ENABLED = DigitalObject.is_enabled?

DEST_DIR = "."

Sistema aderente SAN.

PROVIDER = "ARC-ICAR"

Url base dei complessi archivistici.

FONDS_URL = "http://archivista.ingloba360.it/fonds"

Url base dei soggetti produttori.

CREATORS_URL = "http://archivista.ingloba360.it/creators"

Url base dei soggetti conservatori.

CUSTODIANS_URL = "http://archivista.ingloba360.it/custodians"

Url base degli oggetti digitali.

DIGITAL_OBJECTS_URL = "http://archivista.ingloba360.it/digital_objects"

Parametri forniti dalla redazione SAN per l'esportazione di oggetti digitali.

DL_SISTEMA_ADERENTE = "ARC-ICAR"

DL_PROGETTO_DIGITALIZZAZIONE = "ARC-ICAR:san.cat.prjDgt.000"

DL_CONSERVATORE = "san.cat.sogC.0000"

DL_COMPLESSO = "san.cat.complArch.00000"

1.3.2. Configurazione_dl.rb

Esempio di file di configurazione per l'esportazione in formato METS-SAN

#Identificazione della serie o sottoserie dalla quale estrapolare le unità documentarie che saranno utilizzate

#per la creazione dei metadati METS-SAN

Sistema aderente Sezione RDF del mets-SAN <san-dl:haSistemaAderente>

#url su archivista <http://archivista.ingloba360.it/fonds/448>

DL_FOND_ID="448"

#Gestione del contesto <mets:mdWrap MDTYPE="EAD" MIMETYPE="text">

Sistema aderente Sezione RDF del mets-SAN <san-dl:haSistemaAderente>

PROVIDER = "DATINI"

#I dati relativi al Soggetto Conservatore sono validi per l'intero sistema

#sezioni METS-SAN da compilare con identificativi SAN

#san-dl:haConservatore

#ead-context:repository id

#ead-context:abbr

#ead-context:corpname

DL_HACONSERVATORE="san.cat.sogC.4980"

DL_REPOSITORYID="GGASI:IC0300014497"

DL_ABBR="GGASI:IC0300014497"

DL_CORPNAME="Archivio di Stato di Prato"

#I valori seguenti servono a relazionare un progetto di digitalizzazione SAN con i relativi metadati METS

#sezioni METS-SAN da compilare con identificativi SAN

#san-dl:haProgettoDigitalizzazione

#san-dl:haComplesso

#ead-context:unitid

#ead-context:unittitle

DL_HAPROGETTO ="san.cat.prjDgt.447"

DL_HACOMPLESSO ="san.cat.complArch.113436"

DL_UNITID ="DATINI:san.cat.complArch.113436"

DL_UNITTITLE ="Carteggio, Archivio di Stato di Prato, Datini, Fondaco di Avignone, Carteggio diretto al fondaco da Aigues Mortes a Valenza"

2. Caricamento massivo di oggetti digitali:

1) Configurare la procedura di caricamento attraverso il file di parametri dobj.json, nel quale vanno indicati:

- a) "type": "Fond" la tipologia di entità a cui collegare gli oggetti digitali
- b) "id": "1230" l'id dell'entità
- c) "img_dir": "/public/test/" la directory dove sono contenuti i files digitali da copiare
- d) "file_names" : [
 "file1.jpeg",
 "file2.jpeg"
] è l'array contenente tutti i nomi dei files
- e) "user_id": "1" l'id dell'utente a cui assegnare la procedura
- f) "group_id": "1" l'id del gruppo dell'utente a cui assegnare la procedura

La struttura consente il caricamento multiplo per diverse tipologie di entità. Vanno sempre inserite in una struttura ad array:

[{...}] array con singolo oggetto

[{...},{...},...,{...}] array con n oggetti

2) eseguire la procedura RAILS_ENV=production rake dobj:upload

Esempio di esecuzione 1:

```
root@ICAR02 prove]# ls -l
```

```
totale 8092
```

```
-rw-r--r-- 1 root root 1795955 set 20 18:03 ACT3.24_1899_005_0001.pdf
-rw-r--r-- 1 root root 1644875 set 20 18:03 ACT3.24_1899_005_0002.pdf
-rw-r--r-- 1 root root 1510530 set 20 18:03 ACT3.24_1899_005_0003.pdf
-rw-r--r-- 1 root root 1721895 set 20 18:03 ACT3.24_1899_005_0004.pdf
-rw-r--r-- 1 root root 1601464 set 20 18:03 ACT3.24_1899_005_0005.pdf
-rw-r--r-- 1 root root 1442 set 20 17:34 dobj.json
```

```
[root@ICAR02 prove]# vi dobj.json
```

```
[root@ICAR02 prove]#
```

```
Cd /var/www/html/archimista/lib/tasks
```

```
RAILS_ENV=production rake dobj:upload
```

Esempio di esecuzione 2

```
[root@ICAR02 tasks]# RAILS_ENV=production rake dobj:upload
```

```
(in /var/www/html/archimista)
```

Avvio procedura di caricamento multiplo di oggetti digitali su entità archivistica

Gli oggetti digitali verranno caricati su entità di tipologia Unit con id 6087

Caricamento del file: /var/www/html/archimista/prove/ACT3.24_1899_005_0001.pdf

Caricamento del file: /var/www/html/archimista/prove/ACT3.24_1899_005_0002.pdf

Caricamento del file: /var/www/html/archimista/prove/ACT3.24_1899_005_0003.pdf

La procedura è stata completata, tutti i file digitali sono stati associati all' entità prescelta

Gli oggetti digitali verranno caricati su entità di tipologia Unit con id 6087

Caricamento del file: /var/www/html/archimista/prove/ACT3.24_1899_005_0004.pdf

Caricamento del file: /var/www/html/archimista/prove/ACT3.24_1899_005_0005.pdf

La procedura è stata completata, tutti i file digitali sono stati associati all' entità prescelta

[root@ICAR02 tasks]#