



PAS-palveluiden paketointikomponenttikoulutus

26.11.2019

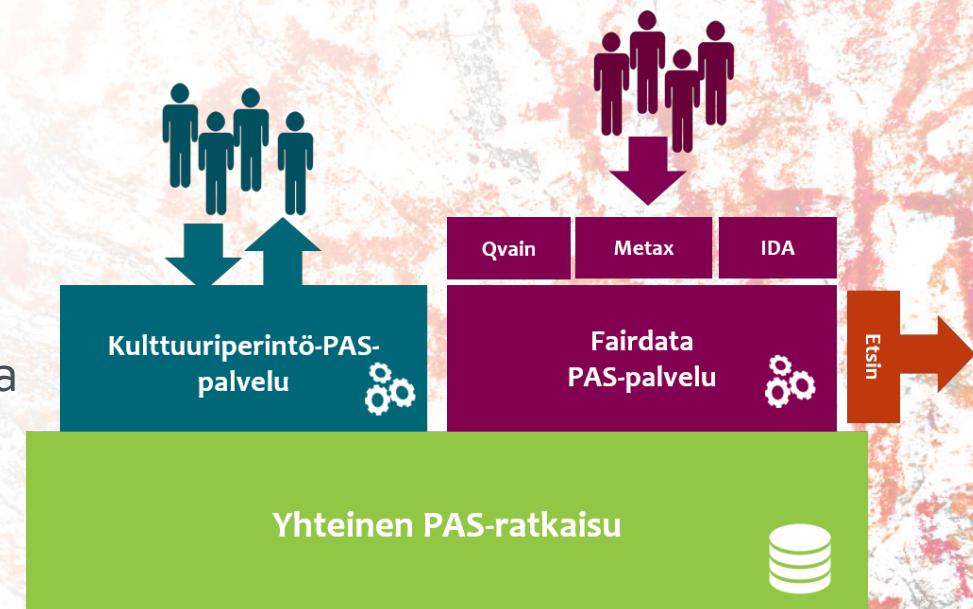


Päivän ohjelma

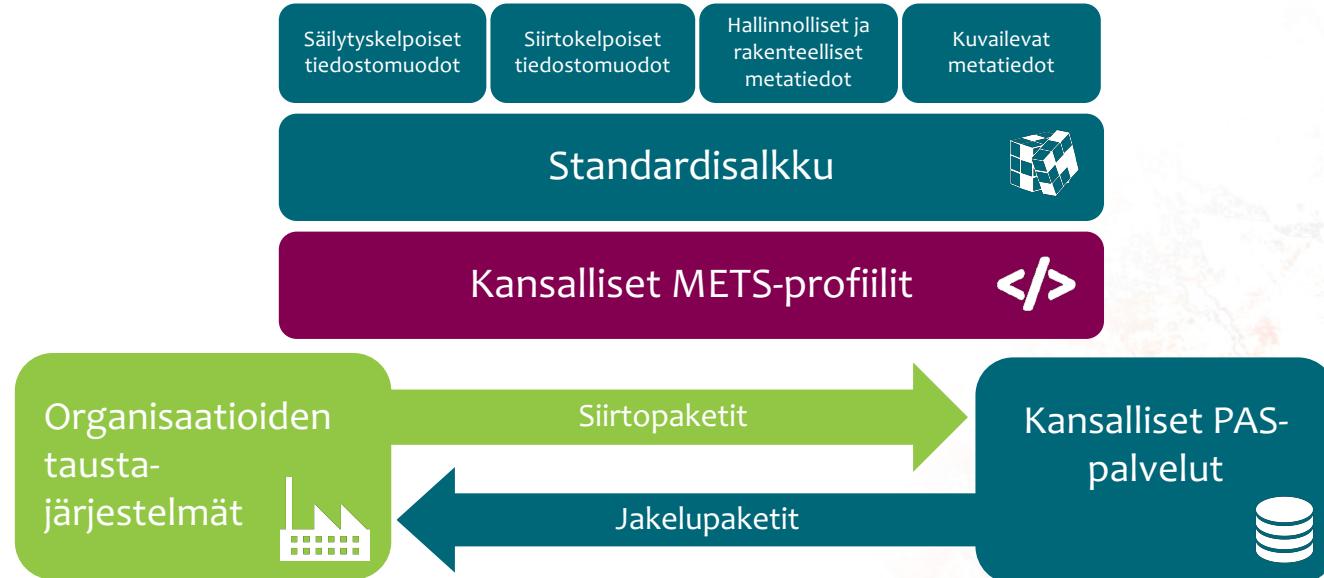
- 10:00 Tervetuloa
- 10:10 Paketoinnin taustaa (Heikki Helin, CSC)
- 10:20 Paketointikomponentin hyödyntäminen (Juha Lehtonen, CSC)
- 11:40 Käyttöliittymä paketointikomponenttiin (Kari Peiponen, Kansallisgalleria)
- 12:00 Lounas (omakustanteinen)
- 13:00 Käytännön harjoituksia (Johan Kylander, CSC)
- 14:00 Kahvitauko
- 14:30 Käytännön harjoituksia (Johan Kylander, CSC)
- 16:00 Koulutus päättyy

PAS-palvelut & päivän ohjelmasta

- CSC tuottaa OKM:lle keskitettyjä PAS-palveluita
 - Kulttuuriperinnön säilyttämiseen (tuotannossa vuodesta 2015)
 - Tutkimusaineistojen säilyttämiseen (tavoitteena aloittaa säilytys vuoden 2019 aikana)
- CSC on ottanut säilytysvastuun noin 550 teratavusta ja noin 1,4 miljoonasta tietopaketista
- PAS-palveluiden kokonaisuus sisältää paljon erilaisia ohjelmistoja/komponentteja
 - Erityisesti Fairdata kokonaisuudessa: Metax, Qvain, IDA, ...
 - Näitä ei tänään käsitellä; ainoastaan paketointikomponenttia
- *Paketointikomponentti != Paketointipalvelu*



Hyödyntäviä organisaatioita ohjaavat PAS-määritykset



- Näkyvin osa PAS-palvelusta hyödyntäville organisaatioille
- PAS-määritykset on tehty tiiviissä yhteistyössä hyödyntävien organisaatoiden kanssa
 - Ja näin myös jatketaan tulevaisuudessa

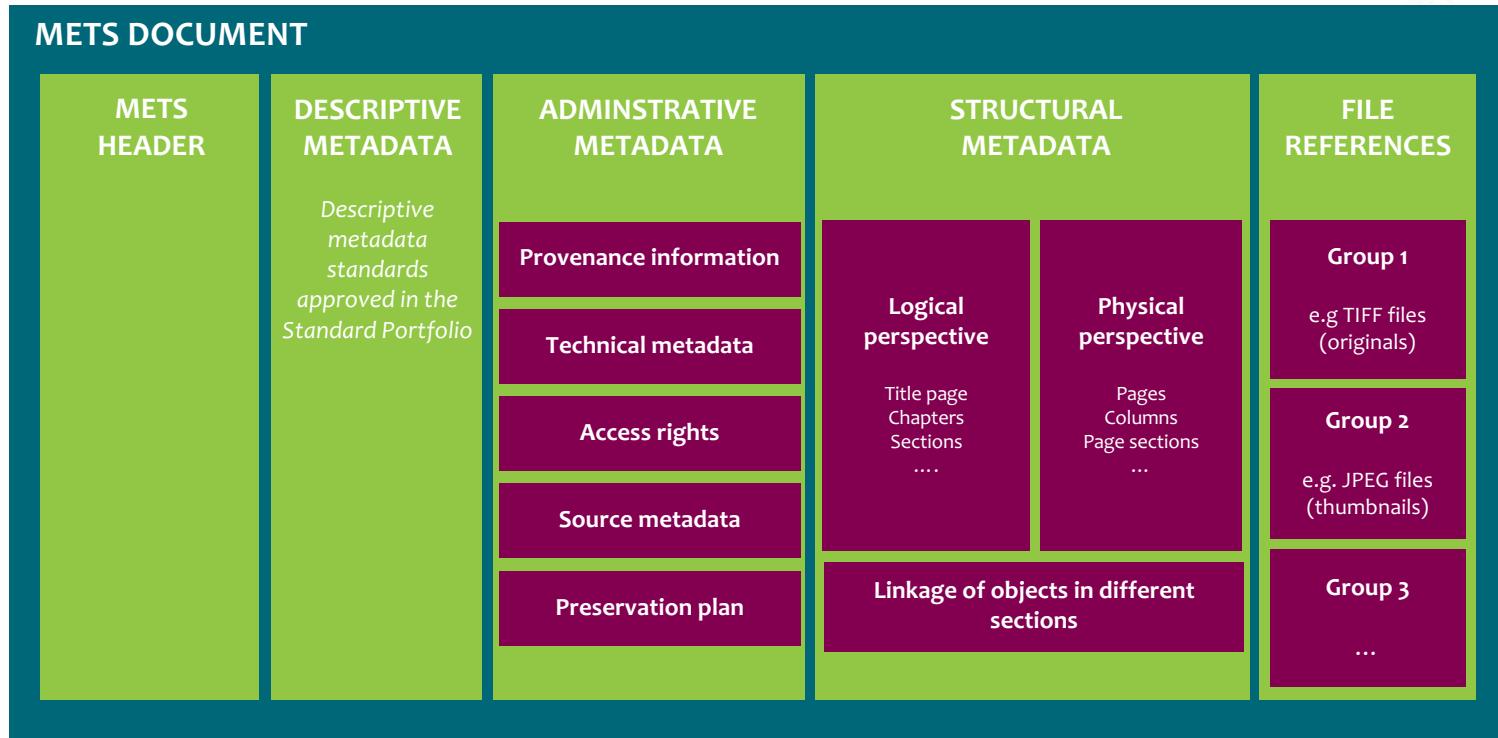
Määritykset

Aineistojen ja niiden metatietojen paketointi pitkäaikaissäilytykseen

- Päivän teeman kannalta merkittävin määritys
- Määrittelee
 - Pakolliset ja ehdollisesti pakolliset metatiedot
 - Kuvalevat, hallinnolliset ja rakenteelliset metatiedot
 - Siirtopaketin (SIP) ja jakelupaketin (DIP) teknisen rakenteen
- Perustuu METS ja PREMIS standardeihin
- Mahdollistaa täysin automaattisen työvuon PAS-palvelun vastaanotossa

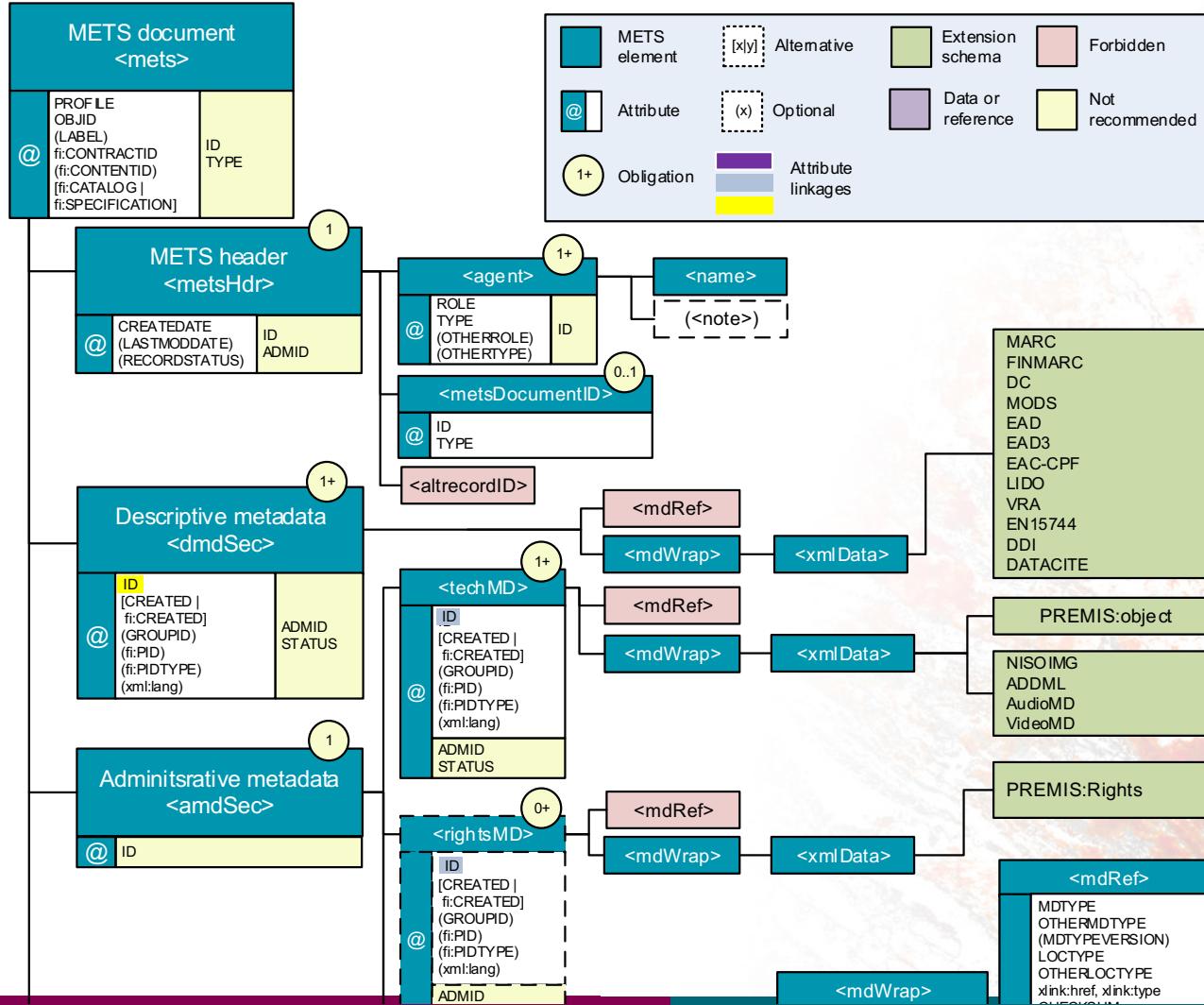
Määritykset

Aineistojen ja niiden metatietojen paketointi pitkäaikaissäilytykseen



Määritykset

Aineistojen ja niiden metatietojen paketointi pitkäaikaissäilytykseen



Paketoinnin vaihtoehdot

ORGANISAATIO PAKETOI ITSE

A. Paketointikomponentin hyödyntäminen

- ✓ Paljon aineistoa
 - + Melko joustava
 - + Ylläpidettävyys määritysten muuttuessa
 - + Vähentää tarvetta METS- ja PREMIS-tuntemukselle
 - Jonkin verran omaa koodausosaamista
 - Määritysten ymmärtäminen

B. Organisaation räätälöimä paketointi

- ✓ Paljon aineistoa; monimutkaisia tarpeita
 - + Joustava ja skaalautuva
 - (Runsaasti) omaa koodausosaamista
 - Määritysten syvälinen ymmärtäminen
 - Ylläpidettävyys määritysten muuttuessa

PAKETOINTIPALVELUN HYÖDYNTÄMINEN

C. Väliaikainen tiedostojen säilytyspalvelu

- ✓ Aineistoa vähän; yksinkertainen ratkaisu
 - + Ei omaa koodausosaamista
 - + Ei vaadi esim. METS- ja PREMIS-tuntemusta
 - Ylimääräinen byrokratia
 - Joustamatona

D. IDA säilytysratkaisu

- ✓ Aineisto+
 - + Vähän omaa koodausosaamista
 - + Ei vaadi esim. METS- ja PREMIS-tuntemusta
 - Joustamatona

VAIN FAIRDATA

CSC auttaa organisaatioita säilyttämisen kaikissa vaiheissa

- Tämä tuki on merkittävä osa organisaatioille tarjottavaa palvelua
 - Sopimusvelvoite CSC:lle
- Kokemus osoittanut, että organisaatiot tarvitsevat usein tukea erityisesti käyttöönnotoissa
- Emme jätä organisaatioita yksin missään PAS-palveluihin liittyvässä asiassa

pas-support@csc.fi



Määritykset saatavilla

Myös englanniksi

- <http://digitalpreservation.fi/specifications>
 - Aineistojen ja niiden metatietojen paketointi pitkääikaissäilytykseen (v.1.7.1)
 - Säilytys- ja siirtokelpoiset tiedostomuodot (v.1.7.0)
 - PAS-palveluiden rajapinnat (v.2.1.0)
- Ohjelmistokoodia, skeemat & schematron säädöt saatavilla @GitHub
 - <https://github.com/Digital-Preservation-Finland/>



Paketointikomponentti





Paketointikomponentin hyödyntäminen



Sisällys

- Mikä on paketointikomponentti?
- Paketointikomponentin asennus
- Käyttö
 - Aineiston tuominen digitaalisiksi objekteiksi
 - Tiedostomuotokohtaisten teknisten metatietojen keruu
 - Tapahtumahistorian luonti
 - Kuvailevan metatiedon tuonti
 - Keruu tiedostoluetteloksi ja rakennekartaksi
 - Luodun ja kerätyn metatiedon koostaminen siirtopaketiksi

Mikä on paketointikomponentti?



Paketointikomponentti (Pre-Ingest Tool)

- Työkalu kansallisten PAS-määritysten mukaisten siirtopakettien muodostamiseksi
- Tavoitteena on helpottaa hyödyntävien organisaatioiden tuotantojärjestelmien integraatiota PAS-palveluun
- Haaste: Hyödyntävien organisaatioiden järjestelmät ovat kuitenkin hyvin erilaisia
- Mahdollistetaan myös suora hyödyntäminen, jos tuotantojärjestelmään integroiminen on tarpeeseen nähdyn kohtuutonta

Paketointikomponentti (Pre-Ingest Tool)

- Joustava, modulaarinen ja muokattava
 - Jaetaan paketointi selkeisiin osakokonaisuuksiin, skripteihin
 - Pidetään skriptit mahdollisimman itsenäisinä ja annetaan tiedot argumentteina
 - Mahdolistaa esim. sen, että hyödynnetään esim. vain tiettyjä osakokonaisuuksia paketointikomponentista, ja toiset osat toteutetaan itse
 - Haluamiaan osakokonaisuuksia voi halutessaan täydentää tekemällä omia toteutuksia
- Saatavilla GitHubista:
 - <https://github.com/Digital-Preservation-Finland>
 - Lisenssi: GPLv3

Paketoinnin vaiheet



Paketoinnin vaiheet



Tekninen metatieto
Perusmetatiedot, myös tiedostomuotovalidointi



Tiedostomuotokohtainen tekninen metatieto
Kuva, ääni, video, rakenteellinen teksti



Kuvaileva metatieto
Valmiin XML-muotoisen kuvailumetatiedon tuonti



Tapahtumahistoria
Aineiston syntyhistoria ja muut aineistoon kohdistuneet tapahtumat

Paketoinnin vaiheet



Paketointikomponentin hyödyntäminen

- Paketointikomponenttia on hyödynnetty useissa PAS-käyttöönnotoissa
- Esim. Fairdata PAS-palvelu hyödyntää paketointikomponenttia
- Paketointikomponentti ei ole “valmis”:
 - Kaikkia määritysten mukaisia tiedostomuotoja ja nk. natiivimuotoja ei vielä tueta
 - Määritykset muuttuvat vuosikellon mukaisella sykkillä, ja paketointikomponentti mukana
 - Kehitystyötä tehdään PAS-käyttöönnottojen edetessä hyödyntävien organisaatioiden tarpeiden mukaisesti
- Määritysten mukaisia paketteja ei kannata alkaa tehdä, jos PAS-käyttöönotto ei ole ajankohtainen
 - Suositeltavaa kuitenkin on, että metatiedot ovat saatavilla (mielellään koneellisesti helppolukuisessa muodossa), ja että tiedostot ovat PAS-määritysten mukaisissa muodoissa.
 - Haasteet voidaan PAS-käyttöönnoton käynnistyessä ratkaista yhdessä.

Paketointikomponentin asennus



Paketointikomponentin asennus

- Paketointikomponenttia hyödynnetään aktiivisesti CentOS-ympäristössä.
 - Paketointikomponentin saanee samaan tapaan toimimaan myös muissa Linux-käyttöjärjestelmissä, esim. Ubuntu.
 - Paketointikomponentti ei sovellu käytettäväksi Microsoft Windows -ympäristössä, vaan sen päälle tarvitaan vähintään Linux-virtuaaliympäristö.
- Asennusohjeissa CentOS-ympäristöön asennetaan virtuaaliympäristö virtualenv-työkalulla, jotta ympäristö saadaan vakioiduksi ympäristöksi.
 - Tällöin muut käyttöjärjestelmään asennetut paketit eivät vaikuta haitallisesti toimintaan.
 - Asennusta voi toki yrittää ilman virtualenv-ympäristöäkin.

Paketointikomponentin asennus

- Ennen asennusta kannattaa varmistua, että Linux-ympäristöstä löytyy gcc-, openssl-devel- ja swig-kirjastot, muussa tapauksessa digitaalisen allekirjoituksen luonti ei toimi.
- Asennus suoritetaan seuraavilla komennolla:
 - *sudo yum install python-virtualenv* (asennetaan virtuaaliympäristö-työkalu)
 - *virtualenv venv* (luodaan uusi ympäristö)
 - *source venv/bin/activate* (aktivoidaan luotu ympäristö)
 - *pip install --upgrade pip* (päivitetään asennussovellus)
 - *pip install -r requirements_github.txt* (asennetaan luotuun ympäristöön)
 - *pip install .*
- Luodun virtuaaliympäristön voi sulkea komennolla “*deactivate*” ja aktivoida taas uudelleen edellä olevalla *source*-komennolla.

Paketointikomponentin asennus

- Lisäksi tarvitaan seuraavat, jos EI validoida tiedostomuotojen oikeellisuutta:
 - Kuvatiedostoille: ImageMagick
 - Ääni- ja videotiedostoille: MedialInfo
 - MXF sisältäen video/jpeg2000-muotoista videota: FFMpeg
 - Office-tiedostoille: Linuxin file-komennosta vähintään versio 5.30
 - PDF-tiedostoille: veraPDF
- Tiedostomuodon oikeellisuuden validointi tarvitsee kaikkien edellisten lisäksi:
 - Teksti- ja XML-muotoisille tiedostoille: JHove, v.Nu
 - Ääni- ja videotiedostoille (muu kuin WAV): FFMpeg
 - WAV-tiedostoille: JHove
 - Kuvatiedostoille: dpx-validator (**PAS**), JHove, pngcheck
 - PDF-tiedostoille: GhostScript, JHove
 - Office-tiedostoille: LibreOffice
 - SPSS-tiedostoille: pspp
 - WARC-tiedostoille: warc-tools

Muita työkaluja Githubissa

- *dpres-ipt*: Työkalu tietopakettien validointiin
 - Tämän työkalun README:sta löytyy luettelo tarvittavista lisäkomponenteista
- *file-scrapers*: Työkalu teknisten metatietojen keräämiseen ja tiedostomuodon validointiin
 - Paketointikomponentti ja dpres-ipt asentavat tämän
 - Tämän työkalun README:sta löytyy luettelo tarvittavista komponenteista
- *dpres-signature*: Työkalu pakettien allekirjoitukseen
 - Paketointikomponentti osaa myös, ja hyödyntää tätä
- *dpres-specification-migrator*: Päivitä vanhat paketit uudempien määritysten mukaisiksi
- *dpres-xml-schemas*: XML skeema- ja schematron-katalogi
- Python-kirjastot mm. METS, PREMIS, MIX, yms. XML-muodoille.
- ...jne...

Siirtopaketin muodostaminen paketointikomponentilla



Ohjeet

- Ohjelmistossa on README.rst, jossa on yleisiä käyttöohjeita.
- Jokaisessa skriptissä on --help-argumentti, jota käytämällä saa lisätietoa kyseisestä skriptistä.

```
[(venv) [vagrant@dpres dpres-siptools]$ create-mix --help
Usage: create-mix [OPTIONS] FILENAME

    Write MIX metadata for an image file.

FILENAME: Relative path to the file from current directory or from
--base_path.

Options:
  --workspace <WORKSPACE PATH>  Workspace directory for the metadata files.
                                Defaults to ./workspace/
  --base_path <BASE PATH>      Source base path of digital objects. If used,
                                give path to the file in relation to this base
                                path.
  --help                      Show this message and exit.
```

Aineiston tuominen digitaalisiksi objekteiksi



Aineiston tuominen digitaalisiksi objekteiksi

- Tuotavalle aineistolle luodaan perusmetatiedot. Jokaiselle tiedostolle joko annetaan parametrina tai muussa tapauksessa luodaan automaattisesti seuraavat metatiedot:
 - Tiedoston tunniste
 - Tiedostomuoto ja sen versio
 - Tarkistussumma (tämä on syytä tallentaa mahdollisimman aikaisessa vaiheessa)
 - Syntyaika
 - Tekstitiedostoille merkistökoodaus tiedostomuodon yhteyteen
- Parametrina voi antaa edellisten lisäksi soveltuvasti:
 - Jonkin formaattirekisterin (esim. PRONOM) koodin
 - Tiedoston järjestysnumeron.
- Tässä vaiheessa myös validoidaan tiedostomuodon oikeellisuus, ellei sitä ohiteta

Tiedostojen tuominen digitaalisiksi objekteiksi

import-object [OPTIONS] [FILEPATHS] ...

FILEPATHS: Tiedosto/hakemisto, optioista riippuen voi antaa yhden tai useita

--workspace <polku>: Työtilan hakemisto

--base_path <polku>: Aineiston juurihakemisto

--skip_wellformed_check: Tiedostomuotojen oikeellisuuden validoinnin ohitus

--file_format <mimetyyppi> <versio>: Tiedoston MIME-tyyppi ja muodon versio

--format_registry <nimi> <koodi>: Formaattirekisterin nimi ja formaatin koodi

Tiedostojen tuominen digitaalisiksi objekteiksi

--*identifier* <*tyyppi*> <*tunniste*>: Tunnisten tyyppi (esim. urn) ja arvo

--*checksum* <*algoritmi*> <*summa*>: Tarkistussumman algoritmi (esim. md5) ja arvo

--*date_created* <*aika*>: Tarkka tai epätarkka tiedoston syntymäaika

--*charset* <*merkistö*>: Tekstitiedostojen merkistökoodaus

--*order* <*numero*>: Tiedoston järjestysnumero

Tiedostojen tuominen digitaalisiksi objekteiksi

import-object tests/data/test_import.pdf --workspace ./workspace/

```
(venv) vagrant@dpres:~/scratch/dpres-siptools$ ls -l workspace/
total 12
-rw-rw-r-- 1 vagrant vagrant 1842 Nov 26 01:37 dcdecc38a52034d453cb3e7d41e85db5-PREMIS%3AOBJECT-amd.xml
-rw-rw-r-- 1 vagrant vagrant 120 Nov 26 01:37 dcdecc38a52034d453cb3e7d41e85db5-scraped.pkl
-rw-rw-r-- 1 vagrant vagrant 182 Nov 26 01:37 md-references.xml
```

```
(venv) vagrant@dpres:~/scratch/dpres-siptools$ cat workspace/md-references.xml
<?xml version='1.0' encoding='UTF-8'?>
<mdReferences>
  <mdReference ref_type="amd" file="tests/data/test_import.pdf">_dcdecc38a52034d453cb3e7d41e85db5</mdReference>
</mdReferences>
```

Varsinainen tulos

```
(venv) vagrant@dpres:~/scratch/dpres-siptools$ cat workspace/dcdecc38a52034d453cb3e7d41e85db5-PREMIS%3AOBJECT-amd.xml
<?xml version='1.0' encoding='UTF-8'?>
<mets:mets xmlns:mets="http://www.loc.gov/METS/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.loc.gov/METS/ http://www.loc.gov/standards/mets/mets.xsd" PROFILE="local" OBJID="bb90f615-aec6-4611-b124-39d2fde20fd5">
  <mets:amdSec>
    <mets:techMD ID="_dcdecc38a52034d453cb3e7d41e85db5" CREATED="2019-11-25T23:37:27.271297">
      <mets:mdWrap MDTYPE="PREMIS:OBJECT" MDTYPEVERSION="2.3">
        <mets:xmlData>
          <premis:object xmlns:premis="info:lc/xmlns/premis-v2" xsi:type="premis:file">
            <premis:objectIdentifier>
              <premis:objectIdentifierType>UUID</premis:objectIdentifierType>
              <premis:objectIdentifierValue>66babcac-2c25-4161-b16e-f6460ea274d0</premis:objectIdentifierValue>
            </premis:objectIdentifier>
            <premis:objectCharacteristics>
              <premis:compositionLevel>0</premis:compositionLevel>
              <premis:fixity>
                <premis:messageDigestAlgorithm>MD5</premis:messageDigestAlgorithm>
                <premis:messageDigest>87bbc5ea526a46f7db43adbdfee71e3e</premis:messageDigest>
              </premis:fixity>
              <premis:format>
                <premis:formatDesignation>
                  <premis:formatName>application/pdf</premis:formatName>
                  <premis:formatVersion>1.4</premis:formatVersion>
                </premis:formatDesignation>
              </premis:format>
              <premis:creatingApplication>
                <premis:dateCreatedByApplication>2019-11-19T17:15:32.920289</premis:dateCreatedByApplication>
              </premis:creatingApplication>
            </premis:objectCharacteristics>
          </premis:object>
        </mets:xmlData>
      </mets:mdWrap>
    </mets:techMD>
  </mets:amdSec>
```

Videosäiliöiden tuominen digitaalisiksi objekteiksi

- Videosäiliöt voivat sisältää useita video- ja audioraitoja
- Tällainen tiedosto annetaan *import-object*-skriptille normaalisti, jolloin se luo koko säiliölle ja sen sisältämille raidoille tekniset perusmetatiedot tarvittavine keskinäisine linkityksineen kerralla.
- Tällä hetkellä tuetaan mm.
 - MPEG-perhe, versiot 1, 2, ja 4 sekä näiden kombinaatiot
 - Quicktime-tiedostot, joissa DV-video- ja PCM-audioraitoja
 - Matroska-tiedostot, jossa FFV1-video- ja PCM-audioraitoja
 - MXF-tiedostot, jossa JPEG 2000 –videoraitoja
 - MOV-tiedostot, jossa AVC- / DV-video- ja AAC- / PCM-audioraitoja
- Tukea eri yhdistelmille lisätään tarpeen mukaan PAS-käyttöönottojen edetessä

Tiedostomuotokohtaisten teknisten metatietojen keruu



Tiedostomuotokohtaisten teknisten metatietojen keruu

- Metatietoihin tarvitaan erilaisia tiedostomuotokohtaisia teknisiä metatietoja
 - Ks. tiedostomuotomääritys: <https://digitalpreservation.fi/specifications>
 - Esimerkiksi kuvatiedostoista tarvitaan kuvan leveys, korkeus, värikoordinaatisto jne.
- Paketointikomponentti osaa analysoida tiedostot ja kerätä pakolliset metatiedot
- Eri tyyppisille aineistoille on omat metatietoformaattinsa:
 - Kuvatiedostoille (MIX-metatieto)
 - Äänitiedostoille (AudioMD-metatieto)
 - Videotiedostoille (VideoMD-metatieto)
 - CSV-tiedostoille (ADDML-metatieto)

Kuvatiedostojen teknisten metatietojen keruu

create-mix [OPTIONS] FILENAME

FILENAME: Kuvatiedosto

--workspace <polku>: Työtilan hakemisto

--base_path <polku>: Aineiston juurihakemisto

Esimerkki:

create-mix images/image.tif --workspace ./workspace

Äänitiedostojen teknisten metatietojen keruu

create-audiomd [OPTIONS] FILENAME

FILENAME: Äänitiedosto

--workspace <polku>: Työtilan hakemisto

--base_path <polku>: Aineiston juurihakemisto

Esimerkki:

create-audiomd audio/sound.wav --workspace ./workspace

Videotiedostojen teknisten metatietojen keruu

create-videomd [OPTIONS] FILENAME

FILENAME: Videotiedosto

--workspace <polku>: Työtilan hakemisto

--base_path <polku>: Aineiston juurihakemisto

Esimerkki:

create-videomd video/video.m1v --workspace ./workspace

Videosäiliöiden teknisten metatietojen keruu

Videosäiliölle video- ja audioraidat tulee kerätä erikseen:

create-videomd video.mp4 --workspace ./workspace

- Kerää kaikkien videoraitojen metatiedot video.mp4-tiedostosta
- Jokaisesta raidasta tulee oma VideoMD-metatietolohkonsa

create-audiomd video.mp4 --workspace ./workspace

- Kerää kaikkien audioraitojen metatiedot video.mp4-tiedostosta
- Jokaisesta raidasta tulee oma AudioMD-metatietolohkonsa

Rakenteellisen tekstin (CSV) teknisten metatietojen keruu

create-addml [OPTIONS] FILENAME

FILENAME: CSV-tiedosto

--workspace <polku>: Työtilan hakemisto

--base_path <polku>: Aineiston juurihakemisto

--header: Käytetään, jos CSV-tiedostossa on otsikkorivi

--charset <merkistö>: CSV-tiedoston merkistökoodaus (pakollinen)

--delim <merkki>: Kenttien eroinmerkki CSV-tiedostossa (pakollinen)

--sep <merkki>: Tietueiden eroinmerkki CSV-tiedostossa (pakollinen)

--quot <merkki>: Lainausmerkit CSV-tiedostossa (pakollinen)

Tapahtumahistorian luonti



Tapahtumahistorian luonti

- Metatiedoissa on pakollista kuvata aineiston tapahtumahistoria.
 - Vähintään aineiston syntyhistoria on kuvattava.
- Tapahtumahistoria lisää todistusvoimaisuutta, kun kirjattujen tapahtumien avulla voidaan esimerkiksi selvittää, minkälaisia aineistoa muuttavia toimenpiteitä aineistolle on tehty.
- Digitoinnissa on tärkeää tietää tekijän ja ajan lisäksi esimerkiksi skannerin tai kameran merkki, malli, asetukset sekä kaikki ne vaiheet, jotka digitoinnissa on tehty.
 - Tällaiset tapahtumat on kannattavaa tallentaa mahdollisimman aikaisessa vaiheessa, koska niiden selvittäminen voi myöhemmin olla varsin vaikeaa.

Tapahtumahistorian luonti

premis-event [OPTIONS] EVENT_TYPE EVENT_DATETIME

EVENT_TYPE: Tapahtumatyyppi

EVENT_DATETIME: Tapahtuman aikaleima

--workspace <polku>: Työtilan hakemisto

--base_path <polku>: Aineiston juurihakemisto

--event_target <polku>: Aineiston hakemisto/tiedosto, johon tapahtuma kohdistuu

--event_detail <kuvaus>: Lyhyt kuvaus tapahtumasta (pakollinen)

Tapahtumahistorian luonti

--event_outcome <tulos>: Tapahtuman lopputulos (pakollinen)

--event_outcome_detail <tuloksen kuvaus>: Tarkempi tuloksen kuvaus

--agent_name <nimi>: Agentin nimi

--agent_type <tyyppi>: Agentin tyyppi

Esimerkki:

```
premis-event creation '2016-10-13T12:30:55' --workspace ./workspace \
--event_target './data' --event_detail 'Creation of data' \
--event_outcome 'success' --event_outcome_detail 'Outcome detail' \
--agent_name 'Demo Application' --agent_type 'software'
```

Kuvailevan metatiedon tuonti



Kuvailevan metatiedon tuonti

- Kuvaileva metatieto kuvaa mm. mitä aineisto sisältää ja siten asettaa sen kontekstiin.
- Kuvailevaa metatietoa ei voida yleisesti tuottaa automaattisesti, ja siten paketointikomponentti ei osaa sitä luoda.
- Siirtopakettiin tarvitaan vähintään yksi kuvaileva metatietokokonaisuus jossakin standardisalkussa esitetyssä muodossa.
- Paketointikomponentilla olemassa olevan XML-muotoisen metatiedon voi liittää paketoitavan aineiston osaksi.

Kuvailevan metatiedon tuonti

import-description [OPTIONS] DMDLOCATION

DMDLOCATION: Polku XML-muotoiseen kuvailevaan metatietoon

--workspace <polku>: Työtilan hakemisto

--base_path <polku>: Aineiston juurihakemisto

--dmdsec_target <polku>: aineistohakemisto, johon kuvaileva metatieto kohdistuu

--remove_root: Poistaa juurielementin tuodusta kuvailevasta metatiedosta

- Tämä voi olla tarpeen esim. Dublin Core -muotoisessa metatiedossa, jossa ei varsinaisesti ole juurielementtiä, vaan metatieto on yleensä standardiin kuulumattoman elementin sisällä.

- Esimerkki:

```
import-description 'metadata/dc_description.xml' --workspace ./workspace |
--dmdsec_target 'tests/data/' --remove_root
```

Keruu tiedostoluetteloksi ja rakennekartaksi



Keruu tiedostoluetteloksi ja rakennekartaksi

- Määritysissä on pakollisina metatietoina tiedosto-osio ja rakennekartta. Nämä linkittävät tiedostot ja edellä luodut metatieto-osiot yhteen.
- Paketointikomponentissa tämä toteutuu automaattisesti omalla skriptillä
 - Komponentti ei kuitenkaan osaa luoda monimutkaisia rakennekarttoja
 - Tälle osa-alueelle tarvittaisiin käyttötapaukset esim. PAS-käyttöönottojen kautta
- Rakennekarttoja tarjotaan paketointikomponentissa tällä hetkellä kahdenlaisia:
 - Hakemistorakenteen mukainen fyysinen rakenne
 - EAD3-metatiedon mukainen looginen rakenne
- Muunlaisia rakennekarttoja punnitaan ja lisätään PAS-käyttöönottojen edetessä

Keruu tiedostoluetteloksi ja rakennekartaksi

compile-structmap [OPTIONS]

--workspace <polku>: Työtilan hakemisto

--structmap_type <tyyppi>: Rakennekarttan tyyppi

--root_type <tyyppi>: Juurielementin tyyppi

--dmdsec_loc <polku>: Kuvailevan metatiedon sijainti

- Käytetään, jos rakennekartta luodaan kuvailevan metatiedon perusteella.
- Tällä hetkellä tässä toiminnossa tuetaan vain EAD3-metatietoa.

- Esimerkki: *compile-structmap --workspace ./workspace*

Luodun ja kerätyn metatiedon koostaminen siirtopaketiksi



Luodun ja kerätyn metatiedon koostaminen siirtopaketiksi

- Edellä olevilla askelilla on nyt luotu tarvittavat metatiedot ja lueteltu aineiston tiedostot, mutta tieto on vielä hajallaan. Tässä vaiheessa aineisto kootaan yhteen.
- Tämä käsittää seuraavat vaiheet:
 - METS-dokumentin luonti aiemmin luodun ja kerätyn metatiedon perusteella
 - Siirtopaketin sähköinen allekirjoitus
 - Siirtopaketin kääriminen TAR-paketiksi.

Metatietojen keruu METS-dokumentiksi

compile-mets [OPTIONS] METS_PROFILE ORGANIZATION_NAME CONTRACTID

METS_PROFILE: METS-profiili

ORGANIZATION_NAME: Siirtopaketin luoneen organisaation nimi

CONTRACTID: PAS-palvelun antama sopimustunniste

--workspace <polku>: Työtilan hakemisto

--base_path <polku>: Aineiston juurihakemisto

--objid <tunniste>: Siirtopaketin tunniste

Metatietojen keruu METS-dokumentiksi

- label <nimi>: Siirtopaketin lyhyt kuvaus
- contentid <tunniste>: Aineistokokonaisuuden tunniste
- create_date <aikaleima>: Siirtopaketin syntyaika
- last_moddate <aikaleima>: Siirtopaketin viimeisin muokkausaika
- record_status <status>: "submission" (oletus) tai "update"
- clean: Poistetaan väliaikaistiedostot
- copy_files: Kopioidaan paketoitava aineisto työtilaan (workspace)

Siirtopaketin allekirjoitus

sign-mets [OPTIONS] SIGN_KEY

SIGN_KEY: Privaattiavain, jolla aineisto allekirjoitetaan

--workspace <polku>: Työtilan hakemisto

- Avaimien luontiin annetaan erilliset ohjeet PAS-palvelun käyttöönnotossa:
 - <https://digitalpreservation.fi/specifications/ohjeita-kayttoonotosta>
- Lisätietoja allekirjoituksesta mm. paketoinnin määritysdokumentissa:
 - <https://digitalpreservation.fi/specifications>

Siirtopaketin kääriminen TAR-tiedostoksi

compress [OPTIONS] DIR_TO_TAR

DIR_TO_TAR: Hakemisto, joka kääritään.

--tar_filename <tiedosto>: Luotavan TAR-tiedoston nimi.

- TAR-käärimisen voi toki tehdä myös esim. Linuxin tar-komennoilla tämä sijasta, tai sen voi pakata ZIP-tiedostoon.
- TAR-tiedosto (tai ZIP-tiedosto) siirretään siirtopakettina PAS-palveluun.

Koostaminen: Esimerkki

- Koostetaan METS-dokumentti:

```
compile-mets ch 'CSC''47265f3e-f423-4926-9f27-7bab08508732' \|  
--workspace ./workspace --copy_files --clean
```

- Allekirjoitetaan siirtopaketti:

```
sign-mets rsa-key.crt --workspace ./workspace
```

- Kääritään TAR-tiedostoon:

```
compress ./workspace --tar_filename sip.tar
```



Kiitos!

<http://digitalpreservation.fi/>

<https://github.com/Digital-Preservation-Finland/>

Twitter: @dpres_fi

pas-support@csc.fi