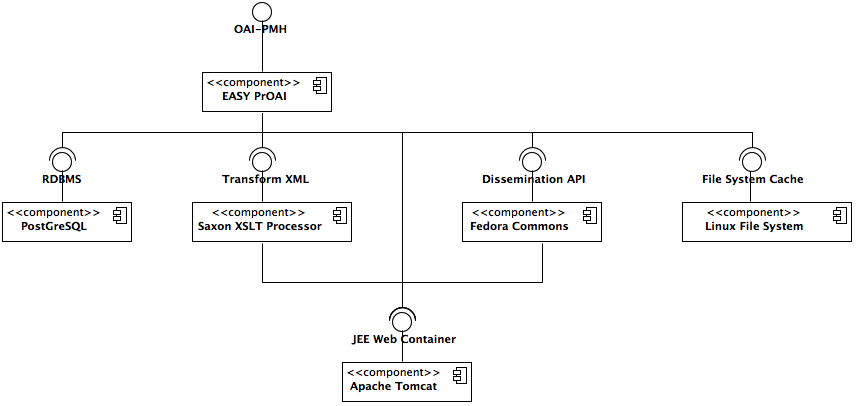
# EASY OAI-PMH handleiding

EASY kan fungeren als een OAI-PMH data provider middels EASY PrOAI, een aangepaste versie van PrOAI. Deze handleiding beschrijft hoe deze service module is opgebouwd.

## Overzicht van de componenten

### De componenten

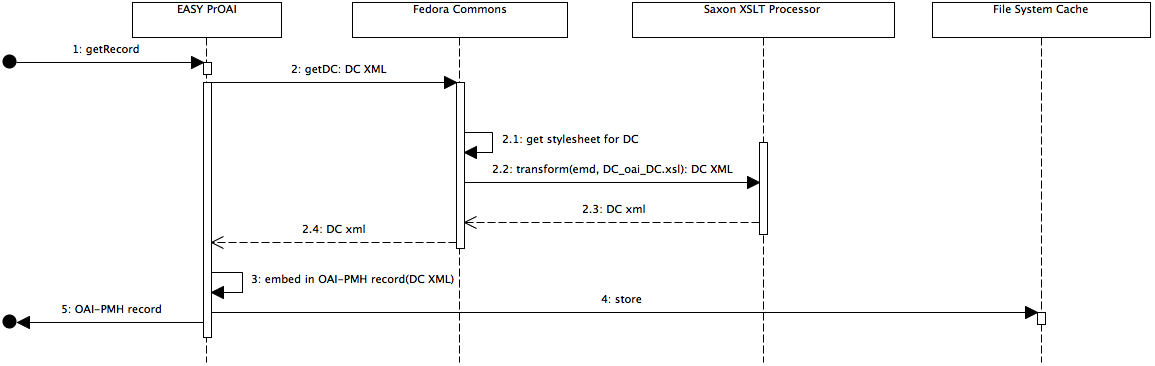
De OAI-PMH-ondersteuning van EASY bestaat uit een aantal samenwerkende onderdelen, die in het onderstaande diagram zijn weergegeven.



De installatie en configuratie van deze componenten is beschreven in de “EASY Installation Guide.” De componenten die de “JEE Web Container” interface consumeren zijn web-applicaties die op Tomcat 6 draaien. (Het gaat om EASY PrOAI, Saxon XSLT Processor en Fedora Commons.) De overige componenten draaien direct op het besturingssysteem.

### Een interactie

Het OAI-PMH-protocol biedt een aantal methodes aan om metadata te harvesten. Hieronder volgt een vereenvoudigd sequence diagram van een interactie waarbij een client één metadatarecord opvraagt van EASY PrOAI. In dit voorbeeld is het record nog niet aanwezig in de file system cache.



1. De EASY PrOAI service module krijgt via http een aanvraag binnen voor een metadata record. Gespecificeerd zijn het Fedora ID van een EASY dataset en een gewenst metadataformat.
2. EASY PrOAI kijkt in zijn cache en – als het record niet aanwezig is – roept op het Fedora Commons digitale object voor de dataset de bij het gewenste metadataformat behorende methode aan.
   1. Fedora Commons zoekt het stylesheet op dat bij het gewenste metadataformat hoort
   2. Fedora Commons roept de Saxon XSLT Processor aan via een SOAP request en geeft daarbij de EASY Metadata van de da­taset en het stylesheet mee.
   3. Saxon levert het getransformeerde document terug.
   4. Fedora Commons geeft dit document door aan EASY PrOAI.
3. EASY PrOAI bedt dit metadata document in een OAI-PMH record.
4. EASY PrOAI slaat dit record op in de cache.
5. EASY PrOAI stuurt dit record naar de aanvrager.

## Componenten in detail

### EASY PrOAI

EASY PrOAI is de naam van een aangepaste versie van PrOAI. Het originele PrOAI wordt slecht onderhouden. Ook is de versie die met Fedora Commons wordt meegeleverd niet dezelfde als de “officiële versie.” Bovendien traden er problemen op met deze versie als gevolg van de query’s die op de Fedora Commons triplestore werden afgevuurd. Dit alles leidde tot de beslissing om een aangepaste versie te gaan gebruiken.

PrOAI biedt een framework waarin plug-ins of “drivers” kunnen worden geïnstalleerd, die gericht zijn op een specifieke back-end. EASY gebruikt(e) t/m versie 2.7 een (aangepaste) versie van de Fedora Commons driver. PrOAI biedt echter geen mechanisme om drivers aan de web-applicatie te binden. Het is dus sowieso nodig om PrOAI te herbouwen als je een nieuwe driver wilt installeren.

Vanaf versie 2.8 is alle aangepaste PrOAI-code inclusief de driver in één project samengevoegd. Het idee van losse drivers is daarmee losgelaten. Op termijn moet wellicht naar een andere, beter onderhouden OAI-PMH-provider worden uitgezien.

### Saxon XSLT Processor

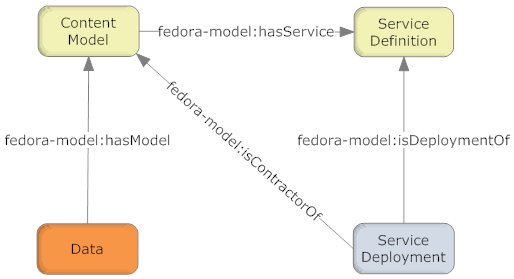
Saxon XSLT Processor is een webservice voor het doen van transformaties op basis van gegeven XSLT stylesheets. De versie van Saxon die we gebruiken is ook verouderd en de herkomst is niet geheel duidelijk.

### Fedora Commons

PrOAI vraagt metadata bij Fedora Commons op via een disseminatiemethode. Per metadataformaat is er een disseminatiemethode. Digitale objecten in Fedora krijgen disseminatiemethodes door lid te worden van een content model.

#### Content Model Architecture

Om verschillende disseminaties van een digital object te ondersteunen heeft Fedora een content model architecture (CMA). Hieronder volgt een schema met de verschillende onderdelen van een content model.



De vier onderdelen zijn elk weer digital objects. De pijlen geven relaties weer die in de RELS-EXT datastream van het object van waaruit de pijl vertrekt zijn vastgelegd. Bijvoorbeeld, in het “Data” digital object staat in de RELS-EXT een RDF-statement dat een content model van dit object beschreven staat in het digital object “Content Model”.

Het content model dat van belang is voor het begrijpen van de disseminaties die bij OAI-PMH een rol spelen heet “info:fedora/easy-model:oai-item1.” Alle datasets hebben dit content model. In de RELS-EXT van dit object staat dat objecten met dit content model de service “info:fedora/easy-sdef:oai-item1” ondersteunen (via hasService).

In “info:fedora/easy-sdef:oai-item1” staat een MethodMap, d.w.z. er staat welke methoden door deze service-definitie worden ondersteund.

In “info:fedora/easy-sdep:oai-item1” is ten slotte vastgelegd hoe de service-definitie wordt gerealiseerd.

De MethodMap komt hier ook voor, maar is uitgebreid met informatie over welke datastreams als input moeten worden gebruikt.

Verder bevat dit object een DSINPUTSPEC datastream waarin details over genoemde datastreams worden gespecificeerd. Hier kan bijvoorbeeld in staan dat ze uit een vast object moeten worden gelezen. Dit is het geval bij de XSLT stylesheets, die in het service deployment object zelf zijn opgeslagen. By default zoekt Fedora de opgegeven datastreams in het object waarop de methode wordt aangeroepen. De EMD datastream wordt dus steeds uit de betreffende dataset gelezen.

Ten slotte is er de WSDL datastream. Deze bevat de koppeling van de Fedora methodedefinitie aan het commando dat de methode uitvoert. Dit is in de huidige versie altijd een http GET request. In het geval van OAI-PMH een request naar de SaxonServlet. De argumenten worden door Fedora in de placeholders in de request gezet. De argumenten voor Saxon zijn URLs waar de stylesheet en input XML kunnen worden opgehaald.

## Taken

### Implementeren van een nieuw metadataformaat

Wat volgt is een stappenplan voor het installeren van ondersteuning voor een nieuw metadataformaat in EASY OAI-PMH. Het is gebaseerd op een concrete casus, namelijk het formaat “OpenAIREplus-flavoured DataCite.” Om het algemeen toepasbaar te maken is met gele markeringen aangegeven waar specifieke waarden zijn ingevuld. In een andere casus zullen hier dus andere waarden ingevuld moeten worden.

#### Benodigdheden

1. Een XSLT stylesheet dat een EMD XML document transformeert naar het doelformaat. Concreet geval: EMD\_OAP.xsl

#### Stappen

1. Voeg het nieuwe XSLT-stylesheet toe als datastream op “info:fedora/easy-sde**p**:oai-item1” (het Service **Deployment** digital object voor de oai-item1 service). Dit kan bijvoorbeeld met de Fedora Web Adminstrator tool (zie Appendix A. Toevoegen datastream met Fedora Web Administrator). Gebruik de volgende gegevens:
   * Datastream ID: EMD\_OAP.xsl
   * Label: Stylesheet for transformation from EMD to OpenAIREplus-flavoured DataCite
   * Control Group: M
   * MIME Type: text/xml
   * State: Active
   * Versionable: Updates will replace most recent version
   * Checksum: Disabled
   * Content: de inhoud van het bestand “EMD\_OAP.xsl” – upload het bestand (XML from a file) of plak de inhoud in de multi-line editbox (XML from text).
2. Breid de datastream METHODMAP in easy-sdef:oai-item1 uit met het volgende element (zie voor het bewerken van een bestaande datastream: Appendix B. Bewerken van een bestaande datastream met Fedora Web Administrator):

|  |
| --- |
| <fmm:Method  label="Get metadata formatted as OpenAIREplus-flavoured DataCite"  operationName="getOpenAIREplus">  </fmm:Method> |

1. Breid de datastream METHODMAP in easy-sdep:oai-item1 uit met het volgende element:

|  |
| --- |
| <fmm:Method  operationLabel="Get metadata formatted as OpenAIREplus-flavoured DataCite"  operationName="getOpenAIREplus"  wsdlMsgName="getOpenAIREplusRequest"  wsdlMsgOutput="dissemResponse">  <fmm:DatastreamInputParm  parmName="EMD"  passBy="URL\_REF"  required="true">  </fmm:DatastreamInputParm>  <fmm:DatastreamInputParm  parmName="EMD\_OAP.xsl"  passBy="URL\_REF"  required="true">  </fmm:DatastreamInputParm>  <fmm:MethodReturnType  wsdlMsgName="dissemResponse"  wsdlMsgTOMIME="text/xml">  </fmm:MethodReturnType>  </fmm:Method> |

1. Breid de datastream DSINPUTSPEC in easy-sdep:oai-item1 uit met het volgende element:

|  |
| --- |
| <fbs:DSInput  DSMax="1"  DSMin="1"  DSOrdinality="false"  wsdlMsgPartName="EMD\_OAP.xsl"  pid="easy-sdep:oai-item1">  <fbs:DSInputLabel>XSL Stylesheet</fbs:DSInputLabel>  <fbs:DSMIME>text/xml</fbs:DSMIME>  <fbs:DSInputInstruction></fbs:DSInputInstruction>  </fbs:DSInput> |

1. Breid de WSDL datastream in easy-sdep:oai-item1 uit met de volgende elementen. Groepeer elk van de elementen bij de bestaande elementen van dezelfde soort.

|  |
| --- |
| <wsdl:message name="getOpenAIREplusRequest">  <wsdl:part name="EMD" type="this:inputType"></wsdl:part>  <wsdl:part name="EMD\_OAP.xsl" type="this:inputType"></wsdl:part>  </wsdl:message>  …  <wsdl:portType>  …  <wsdl:operation name="getOpenAIREplus">  <wsdl:input message="this:getOpenAIREplusRequest">  </wsdl:input>  <wsdl:output message="this:response"></wsdl:output>  </wsdl:operation>  </wsdl:portType>  …  <wsdl:operation name="getOpenAIREplus">  <http:operation location="saxon/SaxonServlet?source=(EMD)\  &amp;style=(EMD\_OAP.xsl)\  &amp;clear-stylesheetcache=true">  </http:operation>  <wsdl:input>  <http:urlReplacement></http:urlReplacement>  </wsdl:input>  <wsdl:output>  <mime:content type="text/xml">  </mime:content>  </wsdl:output>  </wsdl:operation>  </wsdl:binding> |

#### Troubleshooten

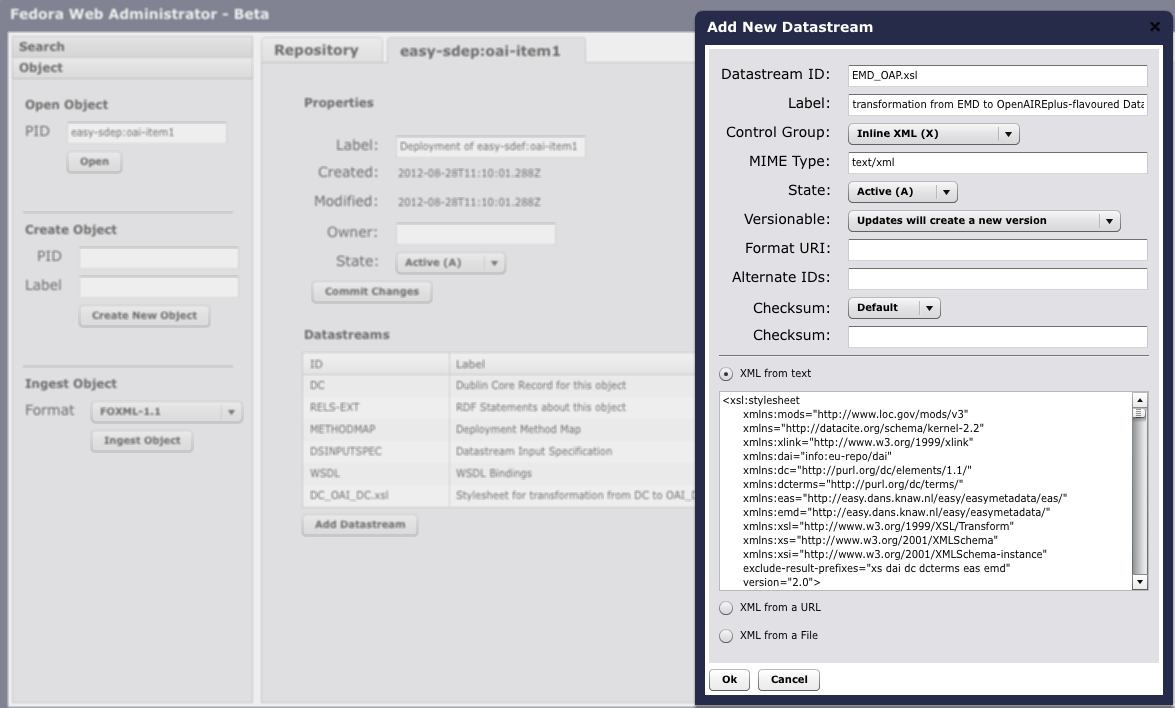
* Testen: klik op Run naast de nieuwe method in <http://evm:8080/fedora/objects/easy-dataset:1/methods>
  + RuntimeException bij het oproepen van de methods  easy-model:EDM1DATASET en/of dans-model:recursive-item-v1 bestaat niet.
  + Nieuw formaat is niet te zien  Sdef bestaat niet.
  + GeneralException: Service Mapper returned error  Foutje in WSDL, “Request” ontbrak in message name.
* In WSDL wordt steeds localhost vervangen door “evm” (op de EASY virtual machine) In /etc/hosts staat een verkeerd IP-adres voor evm.

## Appendices

### Appendix A. Toevoegen datastream met Fedora Web Administrator

#### Stappen

1. Leg indien nodig een ssh-tunnel aan naar poort 8080 (de Tomcat-poort) van de host waarop Fedora Commons draait.
2. Browse naar <http://localhost:8080/fedora/admin> (vervang localhost:8080 als de ssh-tunnel een ander eindpunt heeft, of als directe toegang tot the Fedora Commons host mogelijk is)
3. Vul de Fedora Commons PID van het digital object in waaraan de datastream moet worden toegevoegd en klik op Open.
4. Klik op Add Datastream.
5. Vul de benodigde gegevens in in de “Add New Datastream” specificeer een datastream content.
6. Klik op OK.



### Appendix B. Bewerken van een bestaande datastream met Fedora Web Administrator

#### Stappen

1. Leg indien nodig een ssh-tunnel aan naar poort 8080 (de Tomcat-poort) van de host waarop Fedora Commons draait.
2. Browse naar <http://localhost:8080/fedora/admin> (vervang localhost:8080 als de ssh-tunnel een ander eindpunt heeft, of als directe toegang tot the Fedora Commons host mogelijk is)
3. Vul de Fedora Commons PID van het digital object in waaraan de datastream moet worden toegevoegd en klik op Open.

## Referenties

* <https://wiki.duraspace.org/display/FEDORA36/Creating+a+Service+Definition> - Fedora Commons documentatie over de Service Definition.
* <https://wiki.duraspace.org/display/FEDORA36/Creating+a+Service+Deployment> - Fedora Commons documentatie over de Service Deployement.