CHARMe WP6

(Implementation in archives)

Image18.png

DWD CHARMe-Plugin

DOI Publishing Process

written by: f.kratzenstein

DWD, Offenbach, 2015-03-06

Table of Contents

[Global Parameters / Glossary 2](#_Toc413653717)

[CM SAF - CHARMe enabled DOI Publishing - Components 3](#_Toc413653718)

[CM SAF - CHARMe enabled DOI Publishing - Activitiy 4](#_Toc413653719)

[Summary 5](#_Toc413653720)

[Details 5](#_Toc413653721)

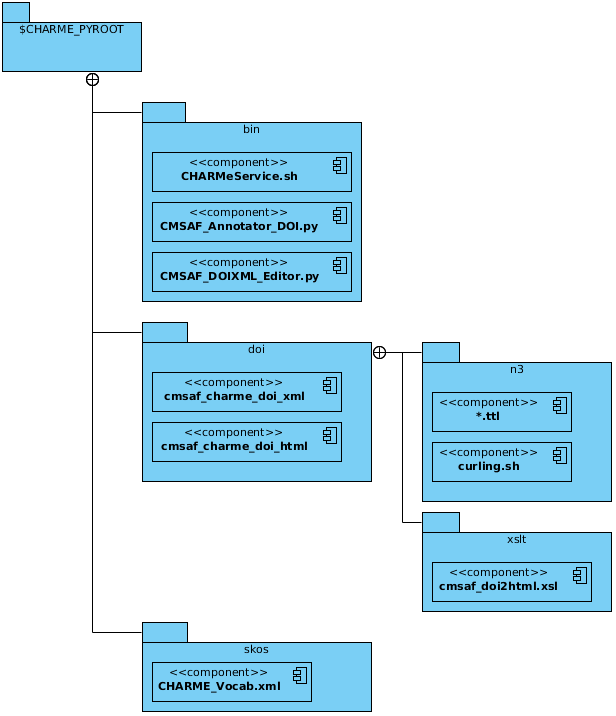
[CMSAF\_DOIXML\_Editor.py 7](#_Toc413653722)

[CMSAF\_Annotator\_DOI.py 9](#_Toc413653723)

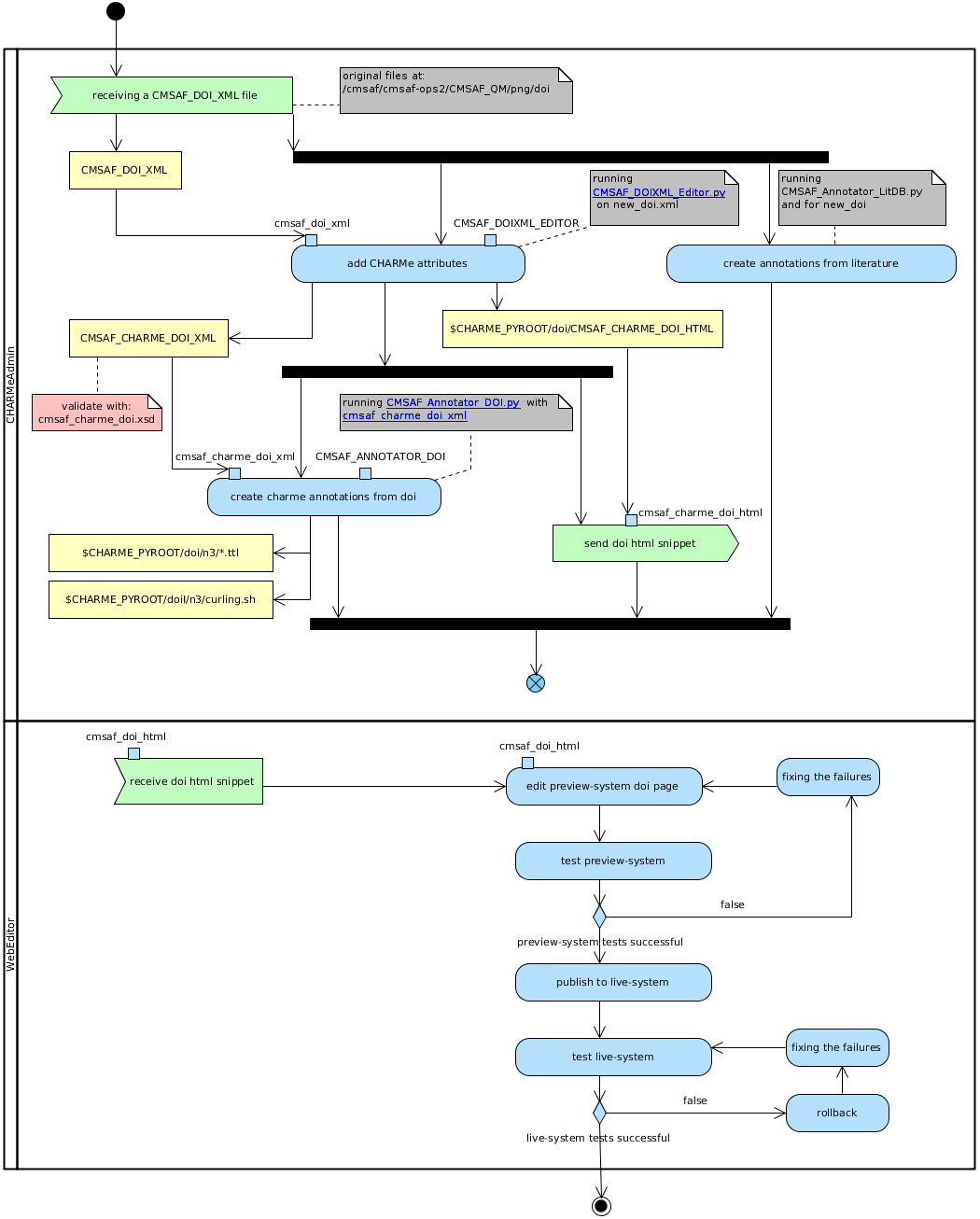
## ****Global Parameters / Glossary****

|  |  |  |
| --- | --- | --- |
| Name | Aliases | Documentation |
| CMSAF\_WUI\_DOI\_DIR |  | /media/nas/x18768/CHARMe | //ofnfa213/CHARMe |
| CHARME\_PYROOT |  | $CHARME\_ROOT/WP6/WP6x |
| CMSAF\_CHARME\_DOI\_HTML |  | html-Snippet mit den CHARMe-Class Attributen zur Aktualisierung der CHARMe-enabled Webpage |
| CMSAF\_CHARME\_DOI\_XML |  | CM SAF WUI erweiterte DOI xml-Datei nochmals erweitert um die CHARMe Attribute |
| CMSAF\_CHARME\_DOI\_XSD | cmsaf\_charme\_doi.xsd | XML-Schema für die CHARMe Erweiterung der CM SAF WUI Doi xml-Dateien |
| WebEditor |  |  |
| CHARMeAdmin |  |  |
| CMSAF\_DOI\_XML |  | DOI xml-Datei, wie sie im WUI benutzt wird |
| CHARME\_ROOT |  | Quellverzeichnis des CHARMe Projektes |
| CMSAF\_ANNOTATOR\_DOI | CMSAF\_Annotator\_DOI.py |  |
| CMSAF\_DOIXML\_EDITOR | CMSAF\_DOIXML\_Editor.py | Python Skript um aus einem CMSAF\_DOI\_XML ein CMSAF\_CHARME\_DOI\_XML zu machen. Zusätzlich wird noch ein CMSAF\_CHARME\_DOI\_HTML erstellt. |

## CM SAF - CHARMe enabled DOI Publishing - Components



## CM SAF - CHARMe enabled DOI Publishing - Activitiy



## Summary

Die Aktivität "CM SAF CHARMe enabled DOI publishing" beschreibt den Prozess, wie nach der Veröffentlichung einer neuen DOI weitestgehend automatisiert entsprechende CHARMe Standard Annotations erzeugt werden können und zusätzlich auch ein html-Snippet erstellt wird, um die CHARMe-enabled doi.html Seite auf der CM SAF Website zu aktualisieren.

Der Prozess ist unterteilt in 2 Verantwortungsbereiche, den des CHARMe Admin's und der Bereich des Web Editor's.

Während der Verantwortungsbereich des CHARMeAdmin weitestgehend automatisiert durch Python-basierte Skripts ablaufen kann, sind die Aktionen des WebEditor's manuell auszuführen.

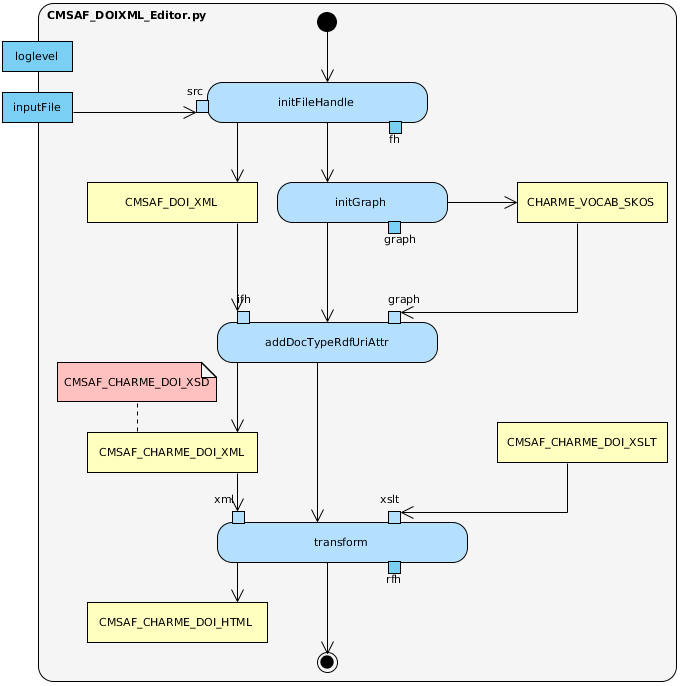
In der hier beschriebenen Aktivität ist die Hauptaufgabe des CHARMeAdmin's, die Standard Annotations, wie ATBD, PUM und VAL zu der neuen DOI zu erzeugen und ggf. Annotations aus der CMSAF-LitDB bzw. aus der EndNote-Library hinzuzufügen. Zusätzlich wird noch ein html-Snippet erstellt, mit dessen Hilfe der WebEditor die CHARMe-enabled doi.html Seite auf der CM SAF Website aktualisieren kann.

## Details

|  |  |
| --- | --- |
| Name | Documentation |
| Image2.png CHARMeAdmin |  |
| **Image3.png receiving a CMSAF\_DOI\_XML file** |  |
| **Image5.png add CHARMe attributes** | Run CMSAF\_DOIXML\_Editor.py with a CMSAF\_DOI\_XML. This will add an attribute docTypeRdfUri to the cmsaf\_documentation tag. This attribute will be populated with a CHARMe Vocab URI of the document type. The result is a CMSAF\_CHARME\_DOI\_XML file, which should be validated against the cmsaf\_charme\_doi.xsd.  In addition a CMSAF\_CHARME\_DOI\_HTML snippet is created. |
| Image5.png **create charme annotations from doi** | For each annotation extracted from the CMSAF\_CHARME\_DOI\_XML a turtle-file is created and stored in a subfolder.  In addition a curling.sh file is created and executed to transfer the turtle-file to the CHARME\_NODE and to perform the insert. For each turtle-file a corresponding log-file for the CHARME\_NODE operation is created in the same subfolder. |
| Image8.png **send doi html snippet** |  |

|  |  |
| --- | --- |
| Image2.png WebEditor |  |
| Image3.png **receive doi html snippet** |  |
| Image5.png **edit preview-system doi page** | adding the CMSAF\_CHARME\_DOI\_HTML snippet to the cmsaf web doi\_charme.html page |
| Image5.png **test preview-system** |  |
| **Image5.png fixing the failures** |  |
| **Image5.png publish to live-system** |  |
| **Image5.png test live-system** |  |

### CMSAF\_DOIXML\_Editor.py



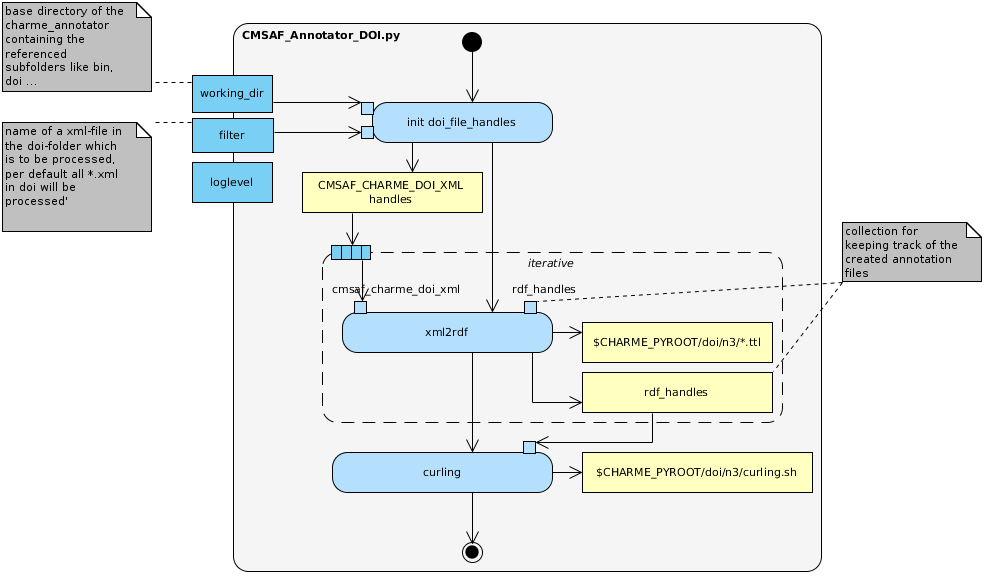
#### Summary

Der CMSAF\_DOIXML\_Editor ist ein Python Skript um aus einem CMSAF\_DOI\_XML ein CMSAF\_CHARME\_DOI\_XML zu machen. Zusätzlich wird noch ein CMSAF\_CHARME\_DOI\_HTML snippet erstellt.

#### Details

|  |  |
| --- | --- |
| Name | Documentation |
| **Image4.png initFileHandle** | creating a local copy of the input file |
| **Image4.png initGraph** | initializes the rdf-graph with the charme vocabulary that we want to use |
| **Image4.png addDocTypeRdfUriAttr** | editing the cmsaf\_doi\_xml by adding a docTypeRdfUri attribute;  creating the cmsaf\_charme\_doi\_xml file |
| **Image4.png transform** | creating the cmsaf\_charme\_doi\_html snippet for the new doi by using a xsl-stylesheet |

### CMSAF\_Annotator\_DOI.py



#### Summary

The CMSAF\_Annotator\_DOI.py is a pyhton script which takes CMSAF\_CHARME\_DOI\_XML files as an input source.

From the CMSAF\_CHARME\_DOI\_XML file it extracts the CHARMe relevant information and transforms it into turtle files. The turtle files are posted to the CHARMe Node, where an insert operation is executed.

#### Details

|  |  |
| --- | --- |
| Name | Documentation |
| **Image2.png init doi\_file\_handles** | initializing the CMSAF\_CHARME\_DOI\_XML files, which we want to process |
| **Image2.png xml2rdf** | handling the doi-xml-file and writing the annotations into turtle-files |
| **Image2.png curling** | creating and executing a shell-script to transfer the annotations to the charme node |

**Step by step instructions**

* Login on oflxs337 as user cmsaf
* Change to /media/x18768/CHARMe/WP6-cmsaf/cmsaf-annotator/bin
* Load python: module load python
* Export python path:
  + module use "/cmsaf/nfshome/routcm/Modules"
  + module load python/2.7.9

currently:

* + export PATH=/cmsaf/cmsaf-ops3/sfinkens/usr/bin:$PATH
  + export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/cmsaf/cmsaf-ops3/sfinkens/usr/lib:/cmsaf/cmsaf-ops3/sfinkens/usr/lib/python2.7/site-packages
  + export PYTHONPATH=/cmsaf/cmsaf-ops3/sfinkens/usr/lib/python2.7/site-packages
* python CMSAF\_DOIXML\_Editor.py -i ../templates/testDOI.xml –d 2.2 (or –d 3)
  + adds CHARME specific content to xml-fil
  + send html file testDOI.html to Iris for implementation on CM SAF DOI page
* create and upload annotations:
* python CMSAF\_Annotator\_DOI.py -w .. -f testDOI.xml ( add -d 3 if version 3 of datacite kernel)