Documentation

of the MyCoRe Maven module

dptbase-viaf

by

Jens Kupferschmidt, University of Leipzig, Germany

Version 2019-06

Leipzig, 18/06/2020

Abstract

This paper should document the VIAF module of the dptbase-module collection. This module is implemented to support all MyCoRe-applications used in scope of projects in Leipzig, Germany. **This module was named als ‘VIAF’ cause it is a container of different nataional authority files like the VIAF.**

Changes

Version	Changes
2014-06-01	first implementation
2014-12-15	corrections
2015-10-22	corrections
2018-04-03	add sex and bibReferences
2018-05-08	add ethnikon and instMembership
2018-08-16	add inheritance for GND data
2019-04-18	correct some GND mappings
2020-02-25	added information about search via SOLR
2020-04-01	add relations by AgRelOn
2020-04-17	correct some LCNAF mappings

Table of content

1 Purpose

For a lot of applications we need a storage structure for authority file entries of big national libraries like the Library of Congress or the Deutsche Nationalbinliothek. On the other side, some applications store their own data on persons without adding them to an inventory in a global authority file systems. In order to enable both local storage as well as the implementation of international authority files, the most common fields for both are implemented in the data model. To convert data from the module to the library data model MARC21, we have an import / export interface. This way, you can migrate your own (local) data to an authority file system of the libraries.

The implementation is a MAVEN-module which can be used in a lot of different MyCoRe[1] applications. You can also use the editor subselect of this datamodel in your own editor forms.

1 see http://www.mycore.de

2 Data model in MyCoRe

2.1 Data model

fieldname	requiered	repeater	MCR-Type	remark
preferredName	yes	for lang	MCRMetaPersonName	
variantName	no	yes	MCRMetaPersonName	
recordType	yes	no	MCRMetaLangText	
recordSource	yes	no	MCRMetaClassification viafRecordSource	
identifier	no	for types	MCRMetaLangText	
ethnikon[2]	no	no	MCRMetaClassification viafEthnika	an empty classification that should be overwritten by special applications
firstDate	no	no	MCRMetaHistoryDate	
firstPlace	no	for lang	MCRMetaLangText	
firstPlaceIdentifier	no	no	MCRMetaLangText	
secondDate	no	no	MCRMetaHistoryDate	
secondPlace	no	for lang	MCRMetaLangText	
secondPlaceIdentifier	no	no	MCRMetaLangText	
sex	yes	no	MCRMetaClassification viafSexISO5218	see Wikipedia ISO/IEC 5218
confession	no	no	MCRMetaClassification viafConfessionType	defined by Boris Liebrenz
profession	no	yes	MCRMetaLangText	
professionIdentifier	no	yes	MCRMetaLangText	
areaOfActivity	no	yes	MCRMetaLangText	
areaOfActivityIdentifier	no	yes	MCRMetaLangText	
floruit	no	yes	MCRMetaHistoryDate	
instMembership	no	no	MCRMetaLangText	
relation	no	yes	MCRMetaLink	uses AgRelOn classification
literarySource	no	yes	MCRMetaLangText	
bibReference	no	yes	MCRMetaLangText	
note	no	yes	MCRMetaLangText	
internalRemark	no	yes	MCRMetaLangText	
link	no	yes	MCRMetaLink	
gndxml	no	no	MCRMetaXML	autofilled
lcnafxml	no	no	MCRMetaXML	autofilled
editor	no	yes	MCRMetaLangText	autofilled or filled by editor

Table 1: Data model dptbase-viaf

Field name	Content	Remark
firstname		
callname		
surname		
fullname		Required or will build from the name parts

2 Ethnikon is per default empty

Field name	Content	Remark
academic	Academic title like PD, Prof. ...	
peerage	Like Lord, Freiherr ...	
numeration	II.	
title		
prefix		
affix		

2.2 Mapping with Marc21 for GND

fieldname	marc21 import / replace	marc21 update
preferredName → fullname	datafield[@tag='100']/ subfield[@code = 'a'] or datafield[@tag='110']/ @ind1 = '1' and subfield[@code = 'b'] or @ind1 = '2' and ubfield[@code = 'a'] or datafield[@tag='111']/ @ind1 = '2' and subfield[@code = 'a'] or datafield[@tag='151']/ subfield[@code = 'a']	use new current entry
preferredName → callname	datafield[@tag='100' and @ind1 ="0']/ subfield[@code = 'a']	use new current entry
preferredName → numeration	datafield[@tag='100' and @ind1 ="0']/ subfield[@code = 'b']	use new current entry
preferredName → title	datafield[@tag='100' and @ind1 ="0']/ subfield[@code = 'c']	use new current entry
preferredName → peerage	datafield[@tag='100' and @ind1 ="1']/ subfield[@code = 'c']	use new current entry
variantName → fullname	datafield[@tag='400' and @ind1 ="1']/ subfield[@code = 'a'] or datafield[@tag='410' and @ind1 ="1 or 2']/ subfield[@code = 'a'] or datafield[@tag='411'']/ subfield[@code = 'a'] or datafield[@tag='451']/ subfield[@code = 'a']	copy from old data with type attribute ‘gnd’ and merge
variantName → callname	datafield[@tag='400' and @ind1 ="0']/ subfield[@code = 'a']	copy from old data with type attribute ‘gnd’ and merge
variantName → numeration	datafield[@tag='400' and @ind1 ="0']/ subfield[@code = 'b']	copy from old data with type attribute ‘gnd’ and merge
variantName → title	datafield[@tag='400' and @ind1 ="0']/ subfield[@code = 'c'] or	copy from old data with type attribute ‘gnd’ and merge

fieldname	marc21 import / replace	marc21 update
	datafield[@tag='410' and @ind1 ="0']/ subfield[@code = 'b']	
variantName → peerage	datafield[@tag='400' and @ind1 ="1']/ subfield[@code = 'c']	copy from old data with type attribute ‘gnd’ and merge
recordType	fix entry tag 100 → 'person' fix entry tag 110 → 'person' fix entry tag 111 → 'person' fix entry tag 151 → 'person'	replace with local data
recordSource	fix categid 'gnd' if data from GND	copy from old dataset
identifier	controlfield[@tag='001']copy from old dataset	copy from old dataset
ethnikon	not set	copy from old dataset
firstDate	datafield[@tag='548']/subfield[@code='a'] [@code='4' and text() = ‘datx’]	copy from old dataset
firstPlace	datafield[@tag='551']/subfield[@code='4' and text() = 'ortg'][@code='a']	copy from old dataset
firstPlaceIdentifier	datafield[@tag='551']/subfield[@code='4' and text() = 'ortg'][@code='0']copy from old dataset	copy from old dataset
secondDate	datafield[@tag='548']/subfield[@code='a'] [@code='4' and text() = ‘datx’]	copy from old dataset
secondPlace	datafield[@tag='551']/subfield[@code='4' and text() = 'orts']][@code='a']	copy from old dataset
secondPlaceIdentifier	datafield[@tag='551']/subfield[@code='4' and text() = 'orts'][@code='0']	copy from old dataset
sex	datafield[@tag='375'/subfield[@code='a'] 1 – male / 2 – female / null = not known	replace with local data
confession	not set	replace with local data
profession	datafield[@tag=’550’/subfield[@code=’a’]/[@code='4' and text() = 'berc']][@code='a']	copy from old data with type attribute ‘gnd’ and merge
professionIdentifier	datafield[@tag=’550’/subfield[@code='4' and text() = 'berc'][subfield[@code=’0’]	copy from old dataset
areaOfActivity	datafield[@tag=’551’/subfield[@code = 'a'] [@code='4' and text() = ‘ortw’]	copy from old data with type attribute ‘gnd’ and merge
areaOfActivityIdentifier	datafield[@tag=’551’/subfield[@code = '0']	copy from old dataset
floruit	datafield[@tag=’548’/subfield[@code = ‘a’]	copy from old data with type attribute ‘gnd’ and merge
instMembership	datafield[@tag=’510’ and @ind1 ="2']/subfield[@code=’4’ and text() = ‘affi’] [@code=’a’]	copy from old data with type attribute ‘gnd’ and merge
relation	not set	copy from old dataset
literarySource	datafield[@tag='530']/subfield[@code='4' and text()=’vorl’][@code='a']	copy from old data with type attribute ‘gnd’ and merge
bibReference	not set	replace with local data
note	not set	replace with local data
internalRemark	not set	replace with local data
link	not set	replace with local data
gndxml	complete XML tree if GND	complete XML tree if GND
lcnafxml	not set	not set
editor	not set	replace with local data

Current problems:

- language detection in Latin script (no ISO lang codes for names)
- very simple date notation

2.3 Mapping with Marc21 for LCNAF

fieldname	marc21 import	marc21 update
preferredName → fullname	datafield[@tag='100']/ subfield[@code = 'a']	datafield[@tag='100']/ subfield[@code = 'a']
preferredName → callname	datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'a']	datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'a']
preferredName → numeration	datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'b']	datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'b']
preferredName → title	datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'c']	datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'c']
variantName → fullname	datafield[@tag='400' and @ind1 ='1']/ subfield[@code = 'a']	copy data that is not with type ‘gnd’ datafield[@tag='400' and @ind1 ='1']/ subfield[@code = 'a']
variantName → callname	datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'a']	copy data that is not with type ‘gnd’ datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'a']
variantName → numeration	datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'b']	copy data that is not with type ‘gnd’ datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'b']
variantName → title	datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'c']	copy data that is not with type ‘gnd’ datafield[@tag='100' and @ind1 ='0']/ subfield[@code = 'c']
recordType	fix entry 'person'	copy from old dataset
recordSource	fix categid 'lcnaf' if data from LCNAF	copy from old dataset
identifier	controlfield[@tag='001']	copy from old dataset
firstDate	datafield[@tag='046']/subfield[@code='f']	Copy from old dataset
firstPlace	not set	replace with local data
firstPlaceIdentifier	not set	replace with local data
secondDate	datafield[@tag='046']/subfield[@code='g']	Copy from old dataset
secondPlace	not set	replace with local data
secondPlaceIdentifier	not set	replace with local data
sex	datafield[@tag='375'/subfield[@code='a'] male – male / female – female	datafield[@tag='375'/subfield[@code='a'] male – male / female – female
profession	datafield[@tag='374'/subfield[@code='a']	copy from old data with type attribute ‘lcnaf’ and merge
professionIdentifier		replace with local data
areaOfActivity	not set	replace with local data
areaOfActivityIdentifier	not set	replace with local data
relation	not set	replace with local data
literarySource	not set	replace with local data
bibReference	not set	replace with local data
note	not set	replace with local data
internalRemark	not set	replace with local data
link	not set	replace with local data
gndxml	not set	not set
lcnafxml	complete XML tree if LCNAF	complete XML tree if LCNAF

Current problems:

- special characters in LCNAF names (see Saladin)

2.4 Mapping and Export to JSON-LD

The export to JSON-LD[3] is recommended for searching the semantic web and simplify the exchange of data. Google recommends the use of the vocabulary of schema.org[4] for better search results, which is also used in viaf in order to provide a certain consistency. The Structured Data Testing *Tool* of Google at https://search.google.com/structured-data/testing-tool tests JSON-LD data with regard to their readability and intelligibility for Google.

Viaf (Persons):

MyCoRe	Fieldname	Schema.org Type / Property
		"@type":"Person"
		"@id":"https://$WebApplicationBaseURL/receive/@ID"
preferredName → fullname	Name	"name"
preferredName → academic	academic title	"honorificPrefix"
preferredName → title	title	"honorificPrefix"
preferredName → peerage	peerage	"honorificPrefix"
preferredName → firstname	first name	"givenName"
preferredName → callname	call name	"additionalName"
preferredName → surname	surname	"familyName"
variantName → fullname	Alternative names	"additionalName"
sex	Sex	"gender"
ethnikon	Ethnikon	"nationality"
firstDate	Date of birth	"birthDate"
firstPlace	Place of birth	"birthPlace"
secondDate	Date of Death	"deathDate"
secondPlace	Place of Death	"deathPlace"
profession	Profession(s)	"jobTitle"
areaOfActivity	Area of competence	"workLocation"
instMembership	Institutional membership	"memberOf"
identifier[@type='viaf']	VIAF-ID	"@type":"PropertyValue","propertyID":"VIAF-ID","value":"http://viaf.org/viaf/*viaf-number*"
identifier[@type='gnd']	GND-ID	"identifier":"@type":"PropertyValue", "propertyID":"GND-ID","value":"http://d-nb.info/gnd/*gnd-number*"
identifier[@type='lcnaf']	LCNAF-ID	"identifier":"@type":"PropertyValue", "propertyID":"LCNAF-ID","value":"http://id.loc.gov/authorities/names/*lcnaf-number*.html"
identifier[@type='trisperson']	TM-Person-ID	"identifier":"@type":"PropertyValue", "propertyID":"TM-PPerson-ID","value":"http://www.trismegistos.org/person/*TM-Person-number*.html"
identifier[@type='trisname']	TM-Name-ID	"identifier":"@type":"PropertyValue", "propertyID":"TM-Name-ID","value":"http://www.trismegistos.org/name/*TM-Name-number*.html"
identifier[@type='other']	Other-ID	"identifier":"@type":"PropertyValue", "propertyID":"Other-ID","value":"*other-text*"

3 http://json-ld.org/
4 https://schema.org/

Table 2: Concordance of dptbase-viaf to Schema.org

2.5 Fields in SOLR

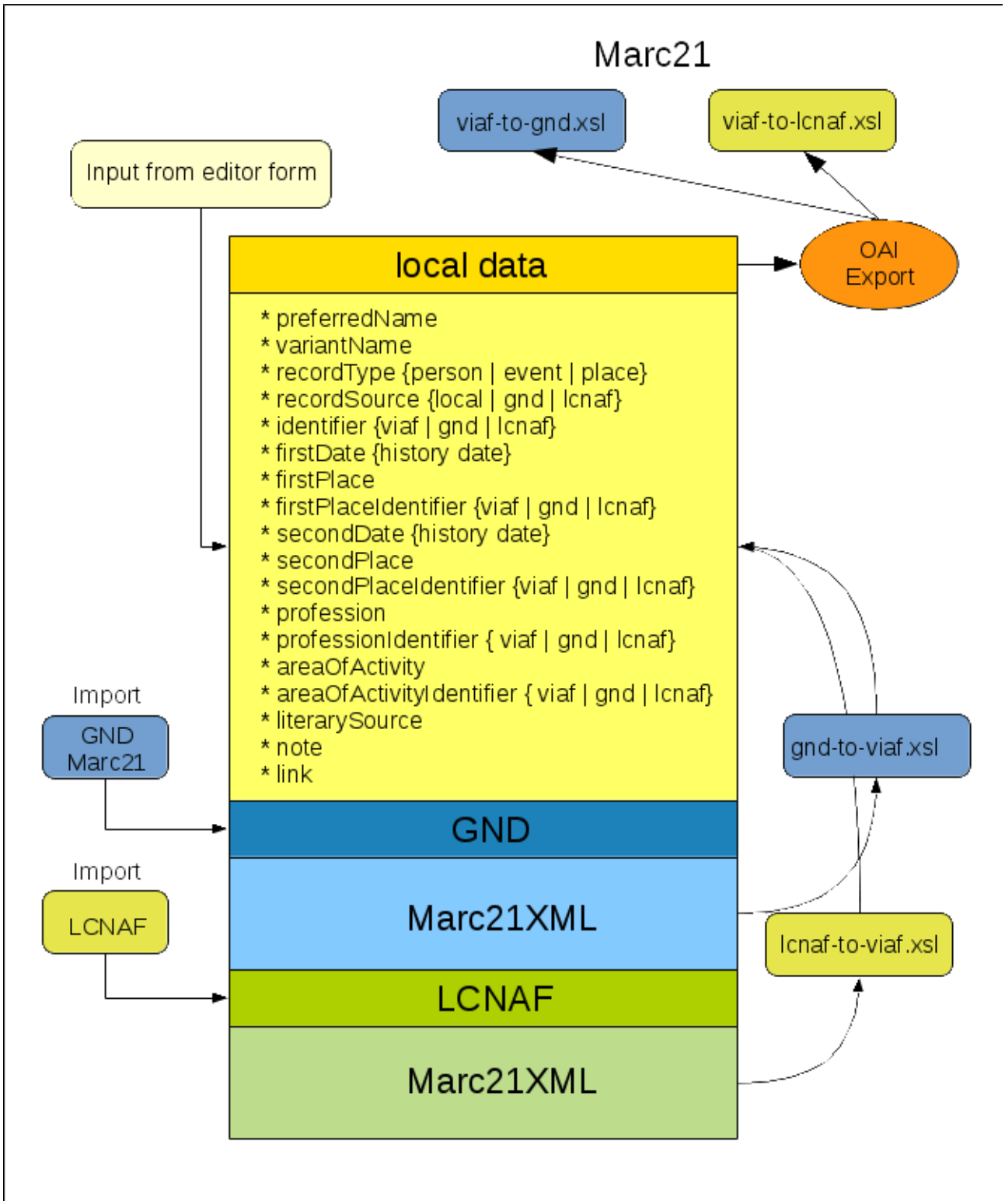
MyCoRe use SOLR[5] as search engine. So we use a mapping to defined SOLR fields wit special transformation and filters.

SOLR field	Source	Tokenizer / filter / mapping	Sort.
viaf_bibref	bibReference	text_general	
viaf_firstdate_from	firstDate/von	pdate	
viaf_firstdate_to	firstDate/bis	pdate	
viaf_floruit_from	floruit/von	pdate	
floruit_to	floruit/bis	pdate	
viaf_identifier	identifier	string	
viaf_identifier_geonames	identifier[@type='geonames']	string	
viaf_identifier_gnd	identifier[@type='gnd']	string	x
viaf_identifier_lcnaf	identifier[qtype='lcnaf']	string	
viaf_identifier_sort	identifier[@type='gnd'] or identifier[qtype='lcnaf'] or identifier[@type='viaf'] or MyCoRe-ID	string	
viaf_identifier_suggest	Copy of viaf_identifier	string	
viaf_identifier_trisperson	identifier[@type='trisperson']	string	
viaf_identifier_trisplace	identifier[@type='trisplace']	string	
viaf_identifier_viaf	identifier[@type='viaf']	string	
viaf_name_all	preferredName/fullname and preferredName/callname and andpreferredName/title and variantName/fullname and variantName/title	**dpt_viaf_text_general** StandardTokenizerFactory special_character_mapping.txt latin_diacritics_mapping.txt synonyms.txt LowerCaseFilterFactory	
viaf_name_all_suggest	**Copy of viaf_name_all**		
viaf_name_ar	preferredName[lang('ar')]/fullname and variantName[lang('ar')]/fullname	**dpt_viaf_text_sort_general** KeywordTokenizerFactory special_character_mapping.txt latin_diacritics_mapping.txt LowerCaseFilterFactory	
viaf_name_de	preferredName[lang('de')]/fullname and variantName[lang('de')]/fullname	**dpt_viaf_text_sort_de** KeywordTokenizerFactory special_character_mapping.txt german_diacritics_for_sort_mapping.txt latin_diacritics_mapping.txt LowerCaseFilterFactory	
viaf_name_en	preferredName[lang('en')]/fullname and variantName[lang('en')]/fullname	**dpt_viaf_text_sort_general** KeywordTokenizerFactory special_character_mapping.txt latin_diacritics_mapping.txt LowerCaseFilterFactory	

5 https://lucene.apache.org/solr/

SOLR field	Source	Tokenizer / filter / mapping	Sort.
viaf_name_id	preferredName[lang('id')]/fullname and variantName[lang('id')]/fullname	**dpt_viaf_text_sort_general** KeywordTokenizerFactory special_character_mapping.txt latin_diacritics_mapping.txt LowerCaseFilterFactory	
viaf_literary	literarySource	text_general	
viaf_place	firstplace and secondplace and areaOfActivity	text_general	
viaf_place_suggest	Copy of viaf_name_all	text_general	
viaf_preferredname	preferredName[lang(DefaultLang)]/fullname or preferredName/fullname	**dpt_viaf_text_sort_general** KeywordTokenizerFactory special_character_mapping.txt latin_diacritics_mapping.txt LowerCaseFilterFactory	x
viaf_preferredname_ar	preferredName[lang('ar')]/fullname or preferredName/fullname	**dpt_viaf_text_sort_general** KeywordTokenizerFactory special_character_mapping.txt latin_diacritics_mapping.txt LowerCaseFilterFactory	x
viaf_preferredname_de	preferredName[lang('de')]/fullname or preferredName[lang('en')]/fullname or preferredName/fullname	**dpt_viaf_text_sort_de** KeywordTokenizerFactory special_character_mapping.txt german_diacritics_for_sort_mapping.txt latin_diacritics_mapping.txt LowerCaseFilterFactory	x
viaf_preferredname_en	preferredName[lang('en')]/fullname or preferredName[lang('de')]/fullname or preferredName/fullname	**dpt_viaf_text_sort_general** KeywordTokenizerFactory special_character_mapping.txt latin_diacritics_mapping.txt LowerCaseFilterFactory	x
viaf_preferredname_id	preferredName[lang('id')] or preferredName/fullname/fullname	**dpt_viaf_text_sort_general** KeywordTokenizerFactory special_character_mapping.txt latin_diacritics_mapping.txt LowerCaseFilterFactory	x
viaf_profession	profession	text_general	
viaf_seconddate_from	secondDate/von	pdate	
viaf_seconddate_to	secondDate/bis	pdate	
viaf_status	servstate/@categid	string	

3 Overview


Marc21
viaf-to-gnd.xsl
viaf-to-lcnaf.xsl
Input from editor form
local data
OAI Export
* preferredName
* variantName
* recordType {person | event | place}
* recordSource {local | gnd | lcnaf}
* identifier {viaf | gnd | lcnaf}
* firstDate {history date}
* firstPlace
* firstPlaceIdentifier {viaf | gnd | lcnaf}
* secondDate {history date}
* secondPlace
* secondPlaceIdentifier {viaf | gnd | lcnaf}
* profession
* professionIdentifier { viaf | gnd | lcnaf}
* areaOfActivity
* areaOfActivityIdentifier { viaf | gnd | lcnaf}
* literarySource
* note
* link
Import
GND Marc21
gnd-to-viaf.xsl
GND
Import
LCNAF
Marc21XML
lcnaf-to-viaf.xsl
LCNAF
Marc21XML

4 XSLT API

For the integration of the **dptbase-viaf** module in your MyCoRe application XSLT templates are provided that you can use/apply in your own XSLT templates. All templates use the parameter *mycoreobject_node* and *next*.

template	decription
printVIAFFirstDate	print the first date text in the current language
printVIAFFirstPlace	print the first place as text in the current language or as output of the Identifier data link
printVIAFSecondDate	print the second date text in the current language
printVIAFSecondPlace	print the second place as text in the current language or as output of the Identifier data link
printVIAFAreaOfActivity	print the area of activity as text in the current language or as output of the Identifier data link
printVIAFProfession	print the profession as text in the current language or as output of the Identifier data link