# DiVA administrator's viewpoint

Gerald Q. Maguire Jr.

July 2025

**This document is a work in progress.**

This document describes the implications of the thesis templates I have developed for use at KTH Royal Institute of Technology (KTH) from the viewpoint of a Digitala Vetenskapliga Arkivet (DiVA) administrator. Other documents provide the template, information for authors, and information about why the template is the way that it is.

This document provides information for DiVA administrators about the template and how it can be used in conjunction with some scripts to **greatly** reduce the work needed to report a thesis into DiVA and Lokalt adb–baserat dokumentationssystem (LADOK), while improving the accuracy of these records. The aim is to avoid the need to cut-and-paste ⇒ more fika time ☺.

**NB** In the case of third-cycle theses, the **student** has to enter the thesis and their included publications.

## 1 Introduction to the template

The assumption is that a student is writing their thesis using my LaTeX template*. The key idea is to have the thesis include (*i*) the information needed for input into DiVA, the TRITA list/database, and LADOK, and (*ii*) information that can be used by US-AB to produce the covers, title page, and book information page.

The Portable Document Format (PDF) file produced has one or more pages at the end that are in a section with a heading of "For DIVA" or this same string with four euro symbols (*i.e.,* '€') before and after the string, with a space before and

*Or perhaps a Microsoft Word Open XML Document (DOCX) file that also produces a `fordiva.json` file.

after the string. An example of the top of such a page is shown in Figure 1, and the bottom is shown in Figure 2. Note that in the bottom figure, we can see two push-pins, representing two attached files: `fordiva.json` and `acronyms.tex`. These attached files make it easier to automate adding the metadata to DiVA* Additionally, some scripts have been developed to mechanically extract this data, as will be described in Section 2.

*At a minimum, all of the data is now collected in one spot from which, in the worst case, one can cut-and-paste it into a single JavaScript Object Notation (JSON) file.

**€€€€ For DIVA €€€€**


```
{
  "Author1":{
    "organisation":{
      "L1":"*****School of XXX*****"
    },
    "ORCiD":"xxxxx-xxxx-xxxx-xxxx",
    "Local User Id":"u1XXXXXX",
    "Last name":"Student",
    "E-mail":"XXXXXXXXXXX@kth.se",
    "First name":"Fake A."
  },
  "Course Info":{
    "Cycle":"3"
  },
  "Degree1":{
    "Educational program":{
      "subjectArea":"*****Unknown subject area*****",
      "Degree":"XXX",
      "programcode":"*****Unknown subject area*****"
    }
  },
  "Title":{
    "Subtitle":"A subtitle in the language of the thesis",
    "Language":"eng",
    "Main title":"This is the title in the language of the thesis"
  },
  "Alternative title":{
    "Subtitle":"Detta är den svenska översättningen av undertiteln",
    "Language":"swe",
    "Main title":"Detta är den svenska översättningen av titeln"
  },
  "Supervisor1":{
    "organisation":{
      "L2":"XXX",
      "L1":"*****School of XXX*****"
    },
    "First name":"A. Busy",
    "E-mail":"XXXXXXXXXXX@kth.se",
    "Last name":"Supervisor",
    "Local User Id":"u1XXXXXX"
  },
  "Supervisor2":{
    "organisation":{
      "L2":"XXX",
      "L1":"*****School of XXX*****"
    },
    "First name":"Another Busy",
    "E-mail":"XXXXXXXXXXX@kth.se",
    "Last name":"Supervisor",
    "Local User Id":"u1XXXXXX"
  },
  "Supervisor3":{
    "Last name":"Supervisor",
    "First name":"Third Busy",
    "E-mail":"XXXXXXXXXXX@tu.va",
    "Other organisation":"Timbuktu University, Department of Pseudoscience"
  },
  "Opponents":{
    "Name":"A. B. Normal \\& A. X. E. Normalè"
  },
  "National Subject Categories":"ddddd, ddddd",
  "SDGs":"XXX, XXX",
  "Other information":{
    "Year":"2025",
    "Number of pages":"1, 39"
  },
  "Series":{
    "Title of series":"TRITA -- XXX-AVL",
    "No. in series":"2025:0000"
  },
  "Copyrightleft":"copyright",
  "Presentation":{
    "Date":"2025-04-15 14:00",
    "Address":"Isafjordsgatan 22 (Kistagången 16)",
    "City":"Stockholm",
    "Language":"eng",
    "Room":"SAL-C and via Zoom https://kth-se.zoom.us/j/ddddddddddd"
  },
  "abstracts":{
```


Figure 1: Top of the page of the For DiVA page

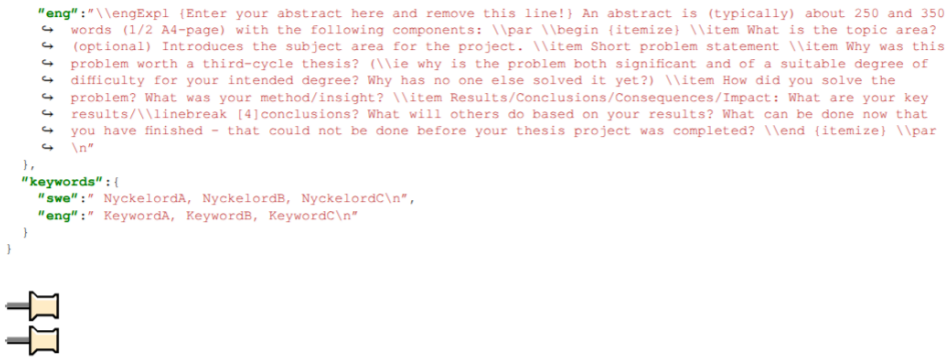


Figure 2: Bottom of the page of For DiVA page

# 2 Getting the necessary data

The collected data shown in Figure 1 is in a format called JavaScript Object Notation (JSON) and is described in Section 2.1. When the author compiles their LaTeX project, as a side-effect, a `fordiva.json` file is generated and attached to the PDF file. This file can be extracted from a PDF file that uses the template, as described in Section 2.2; then, given the information in JSON, it is possible to generate a Metadata Object Description Schema (MODS) file that can be imported into DiVA:

## 2.1 JSON

JSON is a standard way of representing structured data as text. An example of this text is shown in Listing 1. This format is (*i*) readily readable by both humans and computers and (*ii*) is easy to edit, and (*iii*) is commonly supported by many programming languages. The basic element is of the form: *label*: *value*, where *label* is a string and *value* can be another string or a structure (an element inside curly braces). String values are within double quote marks (*i.e.,* "). A label of "First name" and the value "Fake A." is shown in line 6 of the listing below. This is part of a structure that gives a value for "Author1" and this value is a structure with a list of elements for "Last name", "First name", "Local User Id" (*i.e.,* the

kthid), "E-mail", and "organisation ". The organisation in turn can have multiple levels where the first level L1 is the school, L2 is a department, and L3 is a division. In a discussion with DiVA administrators at KTH on 2021-04-29, the consensus was that the organizational affiliation of students should be the school of the thesis examiner, since 1st and 2nd cycle students are in **programs of study** and not schools, department, *etc*.

Listing 1: Text version of the top of the For DIVA output (reformatted to bring out the structure and improve readability - line numbers are added just in the listing)

```
1  {
2    "Author1":{
3      "Local User Id":"u1XXXXXX",
4      "ORCiD":"xxxx-xxxx-xxxx-xxxx",
5      "Last name":"Student",
6      "First name":"Fake A.",
7      "E-mail":"XXXXXXXXXXXX@kth.se",
8      "organisation":{
9        "L1":"*****School of XXX*****"
10     }
11   },
12   "Course Info":{
13     "Cycle":"3"
14   },
15   "Degree1":{
16     "Educational program":{
17       "programcode":"*****Unknown subject area*****",
18       "Degree":"XXX",
19       "subjectArea":"*****Unknown subject area*****"
20     }
21   },
22   "Title":{
23     "Subtitle":"A subtitle in the language of the thesis",
24     "Language":"eng",
25     "Main title":"This is the title in the language of the
         ↪ thesis"
26   },
27   "Alternative title":{
28     "Subtitle":"Detta är den svenska översttningen av
         ↪ undertiteln",
29     "Language":"swe",
```

```
30    "Main title":"Detta är den svenska öäversttningen av
          ↪ titeln"
31  },
```

## 2.2 Given a PDF file that was made using the template

If you have a PDF file made using the template, then one can extract the information using (*i*) a tool that lets you save an attached file (such as Adobe Acrobat Pro) or (*ii*) using a script, such as `extract_pseudo_JSON-from_PDF.py`.

Given such a PDF file, to use a script:

1. Save the PDF file of the thesis, for example: `oscar.pdf`
2. Extract the "For DIVA" information as JSON, as shown in Listing 2

   If acronyms are used in the abstracts and you want to expand them, you can add the "–acronyms acronyms.tex" argument to the extract command line and the script will process the acronyms from the acronyms.tex file (this means that you will also need this file*. The output of the program is a JSON file (oscar.json).

Listing 2: Commands to extract pseudo JSON from the PDF file for Oscar

```
./extract_pseudo_JSON-from\_PDF.py --pdf oscar.pdf --json oscar.json
```

An alternative script is designed to process one or more PDF files. For each PDF file in the input directory, it extracts the embedded files If the `output_directory` is not given, it creates an **output_directory** in the **input_directory**. Otherwise, it creates the **output_directory** if it does not already exist. In both cases, a target directory is created in the output directory based on the basename of the input file with '.pdf' removed and extended with '_embeded_files'. The program outputs each of the embedded files into the target directory. This script is invoked as shown in Listing 3.

Listing 3: Commands to extract embedded or attached from the PDF files

```
extract_embedded_files_from_PDF.py  input_directory  { output_directory ]
```

*Note that if the student has used the glossaries package to use acronyms in the abstract(s) they also need to provide an `acronyms.tex` file, the template automatically attaches this file to the PDF file.

Given the `fordiva.json` you can clean it up and transform it to make a MODS file with the following commands shown in Listing 4.

Listing 4: Cleanup pseudo JSON produced by the LaTeX compiler and then make a MODS file

```
./cleanup_pseudo_JSON-from_LaTeX.py --json fordiva.json --acronyms acronyms.tex
./JSON_to_MODS.py --json fordiva-cleaned.json
```

Now all you have to do is rename the XML file (`modsXML.xml`) that was produced to `xxx.mods` and you are all set to upload the MODS file into DiVA!

## 2.3 TRITA

The Office of Student Affairs assigns the TRITA number. This requires the student to provide the information about their document that is needed so that a TRITA number can be assigned. Fortunately, this material is in the `fordiva.json` file.

An overview of the flow of data is shown in Figure 3. Note that a DiVA administrator only needs to receive a `fordia.json` file from a student and then use this information to assign a TRITA number. The assigned TRITA number is communicated to the student.

Should the student submit just the `fordiva.json` file or the PDF file with the attached `fordiva.json` file?

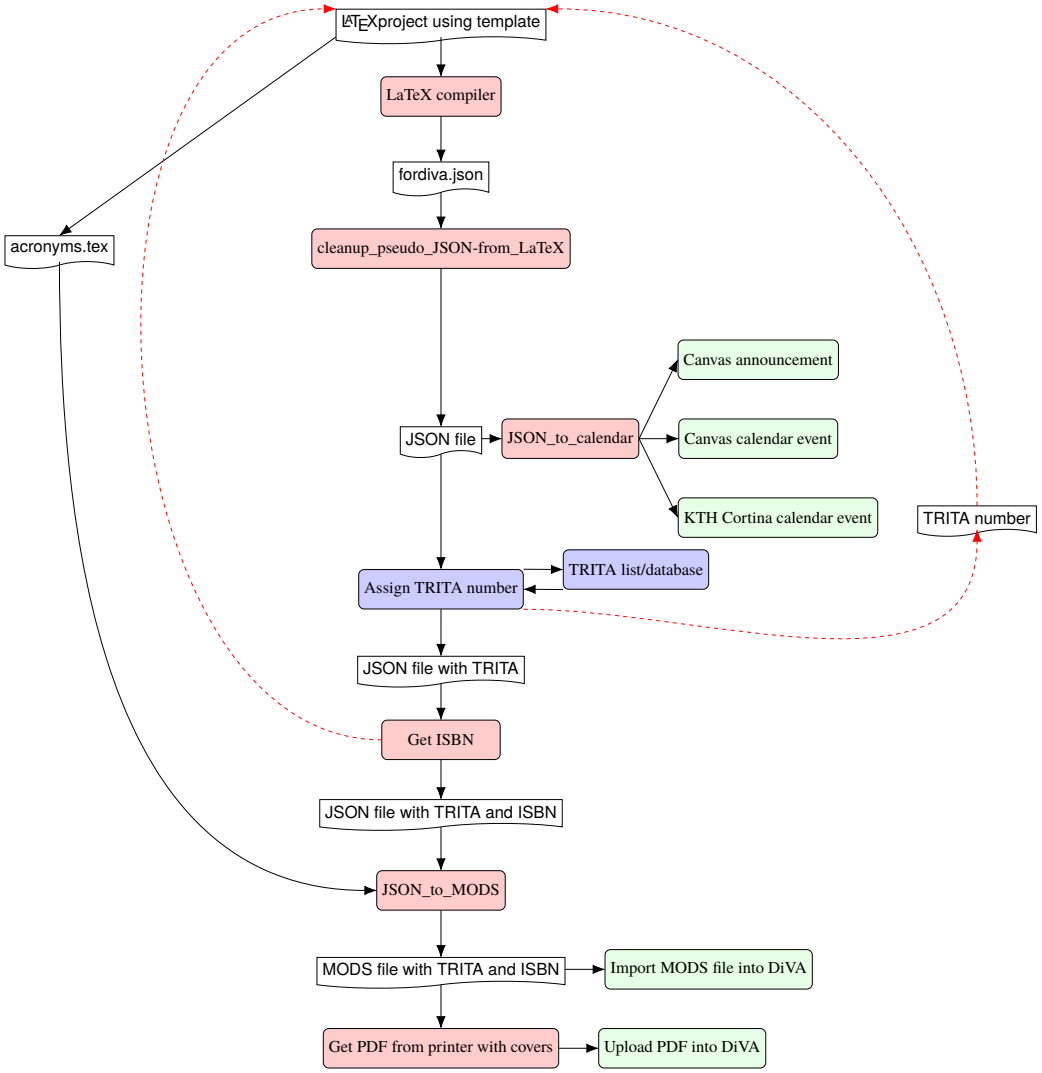


Figure 3: Overall data flow from a LaTeX project using the templates. The DiVA administrators' activities are illustrated with the blue boxes: Assign TRITA number and the TRITA list/database. The dashed red line shows the flow of the assigned TRITA back to the author to input into their LaTeX -project, while the black lines show other types of data.

# 3 Inserting information into LADOK

As part of the effort to minimize the amount of cutting and pasting. I have made a program `JSON_to_ladok.py` that takes the extracted JSON information and uses the information about the author(s) and the title and alternative title to insert this information into LADOK for the module (*i.e.*, "moment" in Swedish) that requires a project title, *i.e.*, 'KravPaProjekttitel' is True. Listing 5 shows an example of using this program to try to put the title and alternative title into LADOK. Basically, the program logic should work, but I do **not** have the required permission to make entries of this sort of data for a degree project (*i.e.*, Rapporteringsrättighet saknas).

Note that this program uses the `ladok3` python library but extends it with some features that are not (yet) in the library. **It should be regarded as very much a work in progress.** However, it illustrates what could be done using the information in the JSON file.

Listing 5: Using the extracted JSON to produce a LADOK entry for a student in the DA231X degree project course

```
./JSON_to_ladok.py --json xxx.json
...
author={'Last␣name': 'xxx', 'First␣name': 'yyy', 'Local␣
↪ User␣Id': 'u1xxxx', 'E-mail': 'oxxx@kth.se',
↪ 'organisation': {'L1': 'School␣of␣Electrical␣Engineering␣
↪ and␣Computer␣Science␣'}}
Canvas user_id=dddd
integration_id=gggggg-gggg-gggg-gggg-gggggggg
ladoK_course_info={'id':
↪ '6683207e-5a5d-11eb-9b32-eeb44fb14647', 'round_id':
↪ '8e15ae14-1d86-11ea-a622-3565135944de', 'education_id':
↪ '374ea085-73d8-11e8-afa7-8e408e694e54', 'instance_id':
↪ '8eee8da9-dd0a-11e8-bb7a-19f8cd1a470e', 'swe_name':
↪ 'Examensarbete␣i␣datalogi␣och␣datateknik,␣avancerad␣
↪ nivå', 'eng_name': 'Degree␣Project␣in␣Computer␣Science␣
↪ and␣Engineering,␣Second␣Cycle'}
moment code=PRO1, requires title=False
moment code=PRO2, requires title=False
moment code=PRO3, requires title=True
trying to store a passing grade for moment=PRO3
Traceback (most recent call last):
  File "./JSON_to_ladok.py", line 533, in <module>
```

```
    sys.exit(main(sys.argv[1:]))
  File "./JSON_to_␣ladok.py", line 519, in main
    status=save_result_degree_project3(ladok,
    ↪ integration_id, course_code, mom['Utbildningskod'],
    ↪ '2021-07-14', 'P', "PF", main_title,
    ↪ alternative_main_title)
  File "./JSON_to_␣ladok.py", line 374, in
  ↪ save_result_degree_project3
    raise Exception("Couldn't␣register␣" + course_moment +
    ↪ "=" + grade_raw + "␣" + result_date_raw + ":␣" +
    ↪ r.json()["Meddelande"])
Exception: Couldn't␣register␣PRO3=P␣2021-07-14:␣Hinder␣mot␣
↪ skapa␣resultat␣påträffat:␣Rapporteringsrättighet␣saknas
```

There is currently a transfer to LADOK project within the group of developers at the IT unit who are working on E-learning - they have been working on an API for entering the title(s) and other information (just as they are making it possible to use a program to enter grades and dates for other courses).

# 4 Final notes

**Best of success in using the scripts (programs)!** If there are questions, contact me at maguire@kth.se .

# Acronyms

| | |
|---|---|
| DiVA | Digitala Vetenskapliga Arkivet |
| DOCX | Microsoft Word Open XML Document |
| JSON | JavaScript Object Notation |
| KTH | KTH Royal Institute of Technology |
| LADOK | Lokalt adb–baserat dokumentationssystem |

MODS Metadata Object Description Schema

PDF Portable Document Format