DEUTSCHE BUNDESBANK
EUROSYSTEM

Document automation at the Research Data and Service Centre using the RDSC Contract Generator

Technical Report 2020-02

Deutschen Bundesbank, Research Data and Service Centre

Jannick Blaschke
Frederik Hering
Lars Huth

Research Data and Service Centre

Abstract

The Research Data and Service Centre (RDSC) of Deutsche Bundesbank provides selected micro-datasets in the context of scientific research projects. The RDSC Contract Generator is a tool developed to automate the creation of all contracts external researchers have to sign in order to get data access. This technical report gives a high-level description of the conceptual and technical set-up and highlights the tool's main benefits.

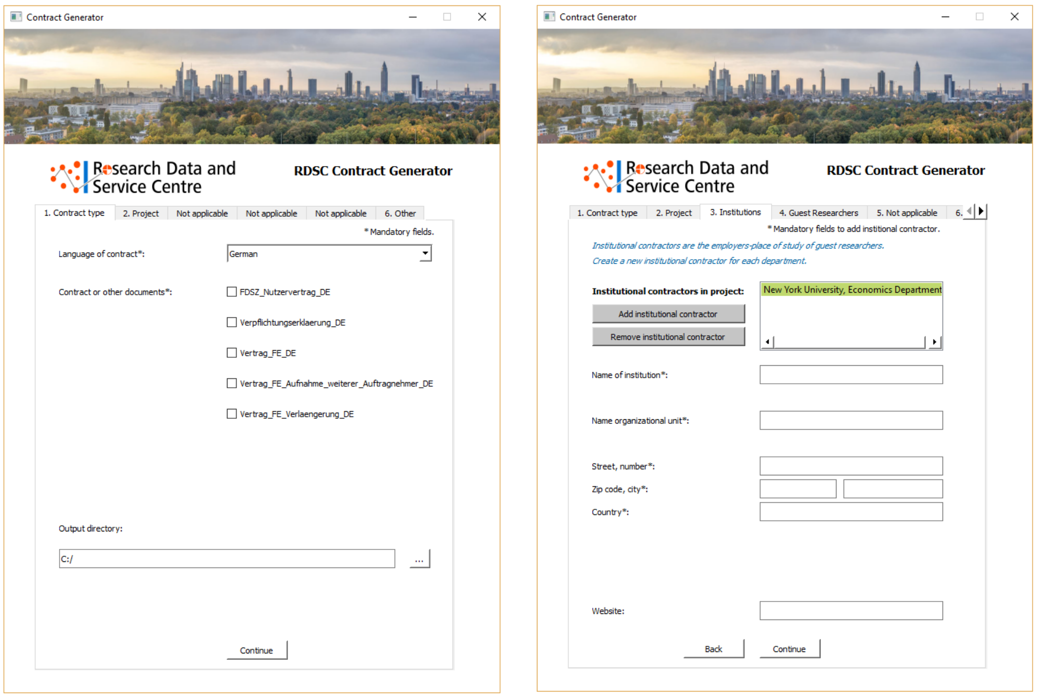
Key words Document automation, RDSC Contract Generator, Python, LaTeX.

Citation Blaschke, J., Hering, F. and L. Huth (2020). Document automation at the Research Data and Service Centre using the RDSC Contract Generator. Technical Report 2020-02, Deutsche Bundesbank, Research Data and Service Centre.

1 Introducing the RDSC Contract Generator

The Research Data and Service Centre (RDSC) of Deutsche Bundesbank provides researchers and analysts with access to selected micro-level data for independent research purposes.[1] Since access to such granular data is subject to legal and data protection requirements, external researchers are required to sign contracts before the start of each research project. Currently, the RDSC distinguishes between two main contracts and a declaration of commitment for the formal undertaking that researchers have to sign before being grated data access (for simplicity referred to as "contracts" in the following). Due to dataset- and researcher-specific legal requirements, the different contract types strongly vary in terms of required information as well as the overall level of complexity.

The RDSC Contract Generator is an application developed to automate the creation of these contracts allowing to create contracts in significantly less time. It was programmed in Python and contains a graphical user interface (GUI) that guides users through the preparation of each individual contract making it as user-friendly as possible.



* Mandatory fields.

Language of contract*:	German
Contract or other documents*:	☐ FDSZ_Nutzervertrag_DE
	☐ Verpflichtungserklaerung_DE
	☐ Vertrag_FE_DE
	☐ Vertrag_FE_Aufnahme_weiterer_Auftragnehmer_DE
	☐ Vertrag_FE_Verlaengerung_DE
Output directory:	C:/

* Mandatory fields to add institional contractor.

Institutional contractors are the employers-place of study of guest researchers.
Create a new institutional contractor for each department.

Institutional contractors in project:	New York University, Economics Department
Name of institution*:	
Name organizational unit*:	
Street, number*:	
Zip code, city*:	
Country*:	
Website:	

Figure 1: Selected screenshots from the RDSC Contract Generator's GUI

1 For more information see Schönberg, T. (2018). Data Access to Micro Data of the Deutsche Bundesbank, Technical Report 2019-02, Deutsche Bundesbank, Research Data and Service Centre.

We put great effort in the design of the tool to make its usage intuitive and already cut the decision tree's branches at an early stage. Therefore, we clearly structured the GUI in different tabs asking for information on the

- **Contract type.** Selection of language, contract type and output directory.
- **Project.** Information on the research project, such as the contract number, project title, project start and end.
- **Institutional Contractors.** Information on researcher's professional affiliation, such as the institution's name and address.
- **Researchers.** Information on involved researchers, such as first and last name, gender, job title and personal address.
- **Other.** Other information, such as name of requested datasets or internal co-researchers.

To avoid burdening users with too many different options and possible entry fields, we structured the GUI in a way, that minimizes the number of decisions and only asks to input information that is really necessary. For this reason, the RDSC Contract Generator's structure is based on pre-defined rules deciding which tabs, input and selection fields are displayed in the GUI. These rules are built directly into the Python code and are determined by the following parameters:

- **Language selection.** While the GUI is always in English, contracts can be created in both English and German.
- **Contract selection.** Since each contract type requires different information, some input and selection fields are only displayed for certain contracts.
- **Other.** In addition, the visibility of some input and selection fields is based on previous decisions made during the preparation of the contract.

Overall, the RDSC Contract Generator significantly reduces the time and effort to create contracts. In particular, our approach brings the following main benefits:

1. **Creating multiple documents at once.** Researchers have to sign a unique contract for each research project they are involved in. Our tool allows the simultaneous creation of all contracts required in the context of a selected project (e.g. for a project with three researchers three contracts have to be created). In addition, it also covers the case where multiple contracts are needed per researcher (e.g. when both a contract and the declaration of commitment for the formal undertaking are necessary).
2. **Importing information.** Most information on the contractors (i.e. researchers and institutions) and their projects can be imported directly from the RDSC's internal project database. Alternatively, users can manually enter the information into corresponding fields in the GUI. All entries can be reviewed, edited and deleted.
3. **Creation of contracts in different languages:** The RDSC Contract Generator supports the creation of contracts in English and German.
4. **Gender specific terms.** Some documents contain gender specific terms, e.g. her/his. The RDSC Contract Generator enables the effortless use of inclusive and legally binding expressions.

2 The RDSC Contract Generator's set-up

In general, the RDSC Contract Generator's technical set-up follows a basic five-step document automation process:

1. **Create a plaintext template for each document which you want to create automatically.** First, we created a plaintext template using TEX[2] with a plaintext format, which enables easy opening and editing of files through scripts.
2. **Set placeholders for variable terms in the document.** Once a template has been created we replaced all terms, which will be changed in the document's creation, with unique placeholders (e.g. *"ZZZ-Project-ID-ZZZ"*).
3. **Write a short script to match user input with previously set placeholders.** After that, we mapped all information which users can insert into the RDSC Contract Generator to the respective placeholders in the templates.
4. **Let the script replace placeholders in a copy of the template.** The RDSC Contract Generator copies the template while replacing all placeholders with user input.
5. **Compile plaintext file into a PDF file.** The plaintext file created from the template is then compiled into PDF.

Taking into account the case of short-term changes in a contract (e.g. contract amendments) that are not yet covered by the tool, users have the option to output the filled-in LaTeX template along with the standard PDF version.

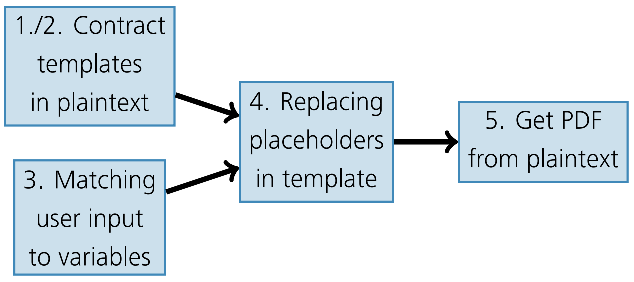

1./2. Contract templates in plaintext
3. Matching user input to variables
4. Replacing placeholders in template
5. Get PDF from plaintext

Figure 2: Functional schema of the document automation process

In current version the following technical prerequisites are required:

- Python Version 3.6.4
- Python library PyQt5[3]
- Lualatex

2 Alternatively, we could have also used RMarkdown or any other document preparation system.
3 The GUI was build using the QtDesigner application, which is based on the Python library PyQt5.