DataMAD Document Generation Example

Table of Contents

[Introduction 1](#_Toc54620031)

[Formatting 1](#_Toc54620032)

[Loops, paragraphs and tables 2](#_Toc54620033)

[Paragraphs 2](#_Toc54620034)

[Table rows 2](#_Toc54620035)

[Table Columns 2](#_Toc54620036)

[Dynamic Table 2](#_Toc54620037)

[Available Variables 3](#_Toc54620038)

[Grant 3](#_Toc54620039)

[Imported Grant 4](#_Toc54620040)

[Data Products 5](#_Toc54620041)

[Digital Data Product 5](#_Toc54620042)

[Model Source Data Product 6](#_Toc54620043)

[Physical Data Product 7](#_Toc54620044)

[Hard-copy Data Product 7](#_Toc54620045)

[Third Party Data Product 8](#_Toc54620046)

# Introduction

Documents templating is accomplished using the python package **python-docx-template.** More detailed documentation can be found here ([https://docxtpl.readthedocs.io/en/latest/)](https://docxtpl.readthedocs.io/en/latest/) for writing templates but the majority of the template design questions can be answered by looking a the Jinja2 template documentation (<https://jinja.palletsprojects.com/en/2.11.x/templates/>). This pages provides everything from the proper formatting of [variables](https://jinja.palletsprojects.com/en/2.11.x/templates/#variables), [filters](https://jinja.palletsprojects.com/en/2.11.x/templates/#list-of-builtin-filters) and [control structures](https://jinja.palletsprojects.com/en/2.11.x/templates/#list-of-control-structures).

This document serves as a guide and an example for the types of things you can do with document generation and DataMAD.

The following sections describe different aspects of templating.

# 

# Formatting

Formatting is done by formatting the variable in the way you would like it to appear:

|  |  |
| --- | --- |
| Format | Example |
| Bold | **{{ grant.grant\_ref }}** |
| Italic | *{{ grant.grant\_ref }}* |
| Underlined | {{ grant.grant\_ref }} |
| Coloured | {{ grant.grant\_ref }} |
| Highlighted | {{ grant.grant\_ref }} |

This can extend to things such as bullets and numbering:

* {%p for item in grant.importedgrant.science\_areas %}
* {{ item }}
* {%p endfor %}

# 

# Loops, paragraphs and tables

When there is a list of values, it makes sense to loop through them to display. This can be done using the Jinja syntax for [control structures](https://jinja.palletsprojects.com/en/2.11.x/templates/#list-of-control-structures).

Paragraphs, bullets and table columns/rows are treated a little differently. The python-docx-template documentation gives an explanation (<https://docxtpl.readthedocs.io/en/latest/#extensions>)

### Paragraphs

* {%p for item in grant.importedgrant.science\_areas %}
* {{ item }}
* {%p endfor %}

### Table rows

Table rows can be created with a loop. Make sure to use the **tr** syntax as explained in the docx-tpl documentation (<https://docxtpl.readthedocs.io/en/latest/#extensions>). The tags with the **tr** will be deleted in the final result.

Add an extra row above and below where you want to place the templated content. You can merge the columns to make it clearer when reading again.

The below example will loop through a list of items and put the values in the columns.

|  |  |
| --- | --- |
| Col1 | Col2 |
| {%tr for row in table\_test.table\_contents %} | |
| {{ row.col1 }} | {{ row.col2 }} |
| {%tr endfor %} | |

### 

### Table Columns

Columns work in a similar way to table rows except that you need to use the **tc** tag to inform the templating engine what it is you want the end result to be. In this instance, you would add an extra column either side of where you want to place the templated content.

As before, these columns will be deleted in the final result.

|  |  |  |
| --- | --- | --- |
| {%tc for col in table\_test.col\_labels %} | **{{ col|title }}** | {%tc endfor %} |

### Dynamic Table

You can combine columns and rows to generate a fully dynamic table:

|  |  |  |
| --- | --- | --- |
| {%tc for col in table\_test.col\_labels %} | **{{ col|title }}** | {%tc endfor %} |
| {%tr for row in table\_test.table\_contents %} | | |
| {%tc for col, value in row.items() %} | {{ value }} | {%tc endfor %} |
| {%tr endfor %} | | |

# 

# Available Variables

You can insert fields in the middle of a sentence. The documents are fed the grant as context.

A grant is related to several different objects:

|  |  |
| --- | --- |
| Related Object | Description |
| Imported Grant | The raw grant which comes from the NERC central grant database. If changes are made in this central database they will be reflected in the importedgrant and linked accordingly. |
| Data Centre | The assigned data centre and other data centre |
| Data Products | These can be used to automatically fill in tables for DMP documents. |

Here are the available fields and their reference strings for each of the objects:

## Grant

All fields are accessed ***using grant.<field>*** e.g. ***grant.grant\_ref***

|  |  |  |
| --- | --- | --- |
| Field | Type | Example |
| grant\_ref | String | {{ grant.grant\_ref }} |
| alt\_data\_contact | String | {{ grant.alt\_data\_contact }} |
| alt\_data\_contact\_email | String | {{ grant.alt\_data\_contact\_email }} |
| assigned\_data\_centre | Data Centre | {{ grant.assigned\_data\_centre }} |
| other\_data\_centre | Data Centre | {{ grant.other\_data\_centre }} |
| hide\_record | Boolean | {{ grant.hide\_record }} |
| date\_contacted\_pi | Date | {{ grant.date\_contacted\_pi }} |
| will\_grant\_produce\_data | Boolean | {{ grant.will\_grant\_produce\_data }} |
| datasets\_delivered | Boolean | {{ grant.datasets\_delivered }} |
| sanctions\_recommended | Boolean | {{ grant.sanctions\_recommended }} |
| case\_for\_support\_found | Boolean | {{ grant.case\_for\_support\_found }} |
| claimed | Bool | {{ grant.claimed }} |
| science\_area | String | {{ grant.science\_area }} |
| jira\_ticket | URL | {{ grant.jira\_ticket }} |
| dmp\_agreed | Boolean | {{ grant.dmp\_agreed }} |
| dmp\_agreed\_date | DateTime | {{ grant.dmp\_agreed\_date }} |

## Imported Grant

Imported grant comes from NERC Siebel system. The most recent version of the imported grant can be accessed using ***grant.importedgrant.<field> e.g. grant.importedgrant.title***

|  |  |  |
| --- | --- | --- |
| Field | Type | Example |
| title | String | {{ grant.importedgrant.title }} |
| grant\_ref | String | {{ grant.importedgrant.grant\_ref }} |
| creation\_date | DateTime | {{ grant.importedgrant.creation\_date }} |
| grant\_status | String | {{ grant.importedgrant.grant\_status }} |
| amount\_awarded | Decimal | {{ grant.importedgrant.amount\_awarded }} |
| call | String | {{ grant.importedgrant.call }} |
| overall\_score | Float | {{ grant.importedgrant.overall\_score }} |
| facility | String | {{ grant.importedgrant.facility }} |
| grant\_type | String | {{ grant.importedgrant.grant\_type }} |
| scheme | String | {{ grant.importedgrant.scheme }} |
| lead\_grant | Boolean | {{ grant.importedgrant.lead\_grant }} |
| parent\_grant | Grant | {{ grant.importedgrant.parent\_grant }} |
| grant\_holder | String | {{ grant.importedgrant.grant\_holder }} |
| department | String | {{ grant.importedgrant.department }} |
| research\_org | String | {{ grant.importedgrant.research\_org }} |
| address1 | String | {{ grant.importedgrant.address1 }} |
| address2 | String | {{ grant.importedgrant.address2 }} |
| city | String | {{ grant.importedgrant.city }} |
| post\_code | String | {{ grant.importedgrant.post\_code }} |
| email | String | {{ grant.importedgrant.email }} |
| work\_number | String | {{ grant.importedgrant.work\_number }} |
| data\_contact | String | {{ grant.importedgrant.data\_contact }} |
| data\_contact\_email | String | {{ grant.importedgrant.data\_contact\_email }} |
| data\_contact\_phone | String | {{ grant.importedgrant.data\_contact\_phone }} |
| routing\_classification | String | {{ grant.importedgrant.routing\_classification }} |
| secondary\_classification | String | {{ grant.importedgrant.secondary\_classification }} |
| science\_areas | List | {{ grant.importedgrant.science\_areas }} |
| ncas | Boolean | {{ grant.importedgrant.ncas }} |
| nceo | Boolean | {{ grant.importedgrant.nceo }} |
| comments | String | {{ grant.importedgrant.comments }} |
| proposed\_start\_date | Date | {{ grant.importedgrant.proposed\_start\_date }} |
| proposed\_end\_date | Date | {{ grant.importedgrant.proposed\_end\_date }} |
| actual\_start\_date | Date | {{ grant.importedgrant.actual\_start\_date }} |
| actual\_end\_date | Date | {{ grant.importedgrant.actual\_end\_date }} |
| abstract | String | {{ grant.importedgrant.abstract|truncate(100) }} |
| objectives | String | {{ grant.importedgrant.objectives }} |

## Data Products

The data products can be accessed using an attribute of the grant object. This will produce a list of the relevant data products. The following sections assume you are looping the data products using a for loop, using ***dataproduct*** as the name for your loop variable.

The Data Product lists are accessed using **grant.<field > e.g. grant.digital\_data\_products**

|  |  |  |
| --- | --- | --- |
| Field | Type | Example |
| digital\_data\_products  Field Guide: [Digital Data Product](#_Digital_Data_Product) | DataProduct List | * {%p for dataproduct in grant.digital\_data\_products %} * {{ dataproduct }} * {%p endfor %} |
| model\_source\_data\_products  Field Guide: [Model Source Data Product](#_Model_Source_Data_1) | * {%p for dataproduct in grant.model\_source\_data\_products %} * {{ dataproduct }} * {%p endfor %} |
| physical\_data\_products  Field Guide: [Physical Data Product](#_Physical_Data_Product_1) | * {%p for dataproduct in grant.physical\_data\_products %} * {{ dataproduct }} * {%p endfor %} |
| hardcopy\_data\_products  Field Guide: [Hard-copy Data Product](#_Hard-copy_Data_Product_1) | * {%p for dataproduct in grant.hardcopy\_data\_products %} * {{ dataproduct }} * {%p endfor %} |
| third\_party\_data\_products  Field Guide: [Third Party Data Product](#_Third_Party_Data_1) | * {%p for dataproduct in grant.third\_party\_data\_products %} * {{ dataproduct }} * {%p endfor %} |

### 

### Digital Data Product

List accessor: ***grant.digital\_data\_products***

{% with dataproduct=grant.digital\_data\_products.first() %}

Where you have a data volume, you will likely want it to display in its shortened form, e.g. 10GB rather than in its byte form. You can do this by using the *filesizeformat* filter. E.g.

{{ dataproduct.data\_volume|filesizeformat }}

|  |  |  |
| --- | --- | --- |
| Field | Type | Example |
| description | String | {{ dataproduct.description }} |
| contact | String | {{ dataproduct.contact }} |
| data\_volume | Integer | {{ dataproduct.data\_volume|filesizeformat }} |
| delivery\_date | Date | {{ dataproduct.delivery\_date }} |
| emabargo\_date | Date | {{ dataproduct.embargo\_date }} |
| doi | Boolean | {{ dataproduct.doi }} |
| preservation\_plan | Preservation Plan | {{ dataproduct.preservation\_plan }} |
| data\_format | DataFormat | {{ dataproduct.data\_format }} |
| additional\_comments | String | {{ dataproduct.additonal\_comments }} |

{% endwith %}

### 

### Model Source Data Product

List accessor: ***grant.model\_source\_data\_products***

{% with dataproduct=grant.model\_source\_data\_products.first() %}

|  |  |  |
| --- | --- | --- |
| Field | Type | Example |
| name | String | {{ dataproduct.name }} |
| contact | String | {{ dataproduct.contact }} |
| description | String | {{ dataproduct.description }} |
| sample\_destination | String | {{ dataproduct.sample\_destination }} |
| additional\_comments | String | {{ dataproduct.additonal\_comments }} |

{% endwith %}

### 

### Physical Data Product

List accessor: ***grant.physical\_data\_products***

{% with dataproduct=grant.physical\_data\_products.first() %}

|  |  |  |
| --- | --- | --- |
| Field | Type | Example |
| name | String | {{ dataproduct.name }} |
| contact | String | {{ dataproduct.contact }} |
| sample\_type | String | {{ dataproduct.sample\_type }} |
| sample\_destination | String | {{ dataproduct.sample\_destination }} |
| additional\_comments | String | {{ dataproduct.additional\_comments }} |

{% endwith %}

### 

### Hard-copy Data Product

List accessor: ***grant.hardcopy\_data\_products***

{% with dataproduct=grant.hardcopy\_data\_products.first() %}

|  |  |  |
| --- | --- | --- |
| Field | Type | Example |
| name | String | {{ dataproduct.name }} |
| contact | String | {{ dataproduct.contact }} |
| data\_format | DataFormat | {{ dataproduct.data\_format }} |
| issues | String | {{ dataproduct.issues }} |
| delivery\_date | Date | {{ dataproduct.delivery\_date }} |
| additional\_comments | String | {{ dataproduct.additional\_comments }} |

{% endwith %}

### 

### Third Party Data Product

List accessor: ***grant.third\_party\_data\_products***

{% with dataproduct=grant.third\_party\_data\_products.first() %}

|  |  |  |
| --- | --- | --- |
| Field | Type | Example |
| name | String | {{ dataproduct.name }} |
| contact | String | {{ dataproduct.contact }} |
| data\_location | String | {{ dataproduct.data\_location }} |
| description | String | {{ dataproduct.description }} |
| data\_volume | Integer | {{ dataproduct.data\_volume|filesizeformat }} |
| responsibility | String | {{ dataproduct.responsibility }} |
| issues | String | {{ dataproduct.issues }} |
| additional\_comments | String | {{ dataproduct.additional\_comments }} |

{% endwith %}