Creation of printable electronic documents in OpenRIMS

(release 2024-04-12)

Contents

resources.help.elreference

Motivation

OpenRIMS provides the possibility to produce printable documents using pre-defined templates and data collected from the application and workflow activities.

The printable documents in OpenRIMS are Office Open XML[1] docx files. This standard complies with Digital Public Good Principles and allows the usage of popular free or commercially available office software.

A template can be created by the office software that supports docx – Open Office, Microsoft Word, etc.

To create a printable document, OpenRIMS injects data from the application and workflow into a template.

The data injection requires definitions of:

- The Data Source to determine where to find the data in the application or workflow pages
- The Data Representation to determine how to inject the data found

EL Expression determines the Data Source and the Data Representation. It is a text delimited by ${ and }. To produce a printable document, OpenRIMS injects application and workflow data into a template instead of EL Expressions.

---

[1] http://officeopenxml.com/

An example

Suppose we would like to produce a company registration invoice as a printable document.

The template docx file looks like below

Invoice

Attn to ${/contactName@literal}   2   1

Bank	Neverland Royal Bank
Bank Code	00001
Account #	26002345
Account name	NMRA fees for office ${/map_location@office} 2
Amount, currency	1000 NLA
Payment details	Fee for registration new company ${/prefLabel@literal} at ${/map_location@choice} 2

Additional users to register:

${/users/users/0/prefLabel@literal}
${/users/users/1/prefLabel@literal}
${/users/users/2/prefLabel@literal}
${/users/users/3/prefLabel@literal}   2
${/users/users/4/prefLabel@literal}
${/users/users/5/prefLabel@literal}
${/users/users/6/prefLabel@literal}

This invoice has been issued at 2/14/2023 and will valid next 10 business days.

Chief of Financial Department Mr. Nevermore F Neverland

3

Figure 1 Template is a plain docx file. It doesn't contain fields, hidden data, and any other particularities

1. The plain text
2. EL[2] expressions
3. The plain picture

The producing a printable document starts automatically while uploading a docx file from a "resources" component placed in an electronic form.

---

[2] Expression Language inspired by good old https://docs.oracle.com/javaee/6/tutorial/doc/gjddd.html

Such steps should be followed to implement the possibility of producing a printable document:

1. Create a docx template document with plain text
2. Include EL expressions in the template to allow collect data
3. Upload the template to a resource definition
4. Place the resource as a field in the electronic form to allow a user to download a printable document directly from the electronic form

After downloading from the resource, a printable electronic document will look like this:

Invoice

Attn to Phelim O'Neal

Bank	Neverland Royal Bank
Bank Code	00001
Account #	26002345
Account name	NMRA fees for office Kathmandu
Amount, currency	1000 NLA
Payment details	Fee for registration new company My company at Bagmati Province, BHAKTAPUR, Bhaktapur Municipality

Additional users to register:

Zeb Stump

This invoice has been issued at 2/13/2023 and will valid next 10 business days.2/14/2023

Chief of Financial Department Mr. Nevermore F Neverland

Figure 2 Template that is resolved by OpenRIMS

The EL Expression

To produce a printable document, OpenRIMS injects application and workflow data into a template instead of EL expressions. An EL expression defines where to get data and how to represent it in a printable document:

${Data Source@Data Representation}

Where:

- **${** and **}** are delimiters to distinct an EL Expression and plain text in a template
- Data Source defines where to find the data to inject. For the reference, see [

- **Data** Sources]
- Data Representation manages the data injection. For the reference, see [**Data representation**].

For example, an application contains the address of the applicant in the field "address" on the first page of the vaccine registration application. The page and the field define the Data Source. The address may be represented as:

- A sequency of administrative units, e.g., Calcasieu Parish, Louisiana, USA
- Any administrative unit, e.g. Louisiana
- GIS Coordinates, e.g. 30.216198856576895, -93.35905854846415
- NRA office, that is server this address, e.g., Deep South Federal Office

Data Sources

The whole application

The printable representation of an application can be injected using EL expression ${/@form}.

An example

20220208 Pharmacy	
register	
Register Number	000070
Registration Date	Feb 14, 2022
certificate	
Register Number	000080
Registration Date	Feb 7, 2022
Expiry Date	Feb 7, 2024
Classifiers	
System of Medicine	Allopathy, GA
Service	Government, A) Lower level Health facility
Company Detail	
Estimated Capital	1 000 000
Name	Pilulkin Ltd
Authorized Person	
Name	Isidora Covarubio de los Llanos
Phone Number	+2(067)1234567
Email Address	nobody.nowhere@gmail.com

Figure 3 The whole application in printable form (simplified)

resources.help.elreference

Modifications on any page in the application for modification (amendment) workflows

The EL Expression ${/@changes} injects modifications of an application. This expression can be used in the modification workflow. For example:

Owner		
	Previous	**Modification**
Estimated Capital	300 000	1 233 330
Name	Abdurahmann, Zeb Stump	Isidora Covarubio de los Llanos, Abdurahmann, Zeb Stump

A page in the application

Each page in the application form, except the first, has a name. This name can be found in the application form data configuration as names of "thing" components. For example:

retail.site.owned.persons

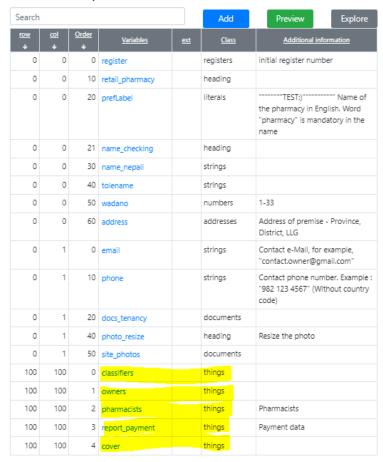| row ↓	col ↓	Order ↓	Variables	ext	Class	Additional information
0	0	0	register		registers	initial register number
0	0	10	retail_pharmacy		heading	
0	0	20	prefLabel		literals	""""""""TEST:)"""""""""" Name of the pharmacy in English. Word "pharmacy" is mandatory in the name
0	0	21	name_checking		heading	
0	0	30	name_nepali		strings	
0	0	40	tolename		strings	
0	0	50	wadano		numbers	1-33
0	0	60	address		addresses	Address of premise - Province, District, LLG
0	1	0	email		strings	Contact e-Mail, for example, "contact.owner@gmail.com"
0	1	10	phone		strings	Contact phone number. Example : "982 123 4567" (Without country code)
0	1	20	docs_tenancy		documents	
0	1	40	photo_resize		heading	Resize the photo
0	1	50	site_photos		documents	
100	100	0	classifiers		things	
100	100	1	owners		things	
100	100	2	pharmacists		things	Pharmacists
100	100	3	report_payment		things	Payment data
100	100	4	cover		things	

Figure 4 classifiers, owners, pharmacists, report_payment, and cover are application page names

Thus, EL Expression ${/cover@form} injects the content of the whole page "cover". There is no way to inject only the first page of the application.

A field in an application's page

Each field on an application's page has a name and a class. The name is used to access the field's data, the class is used to determine the kind of data to be accessed.

retail.site.owned.persons

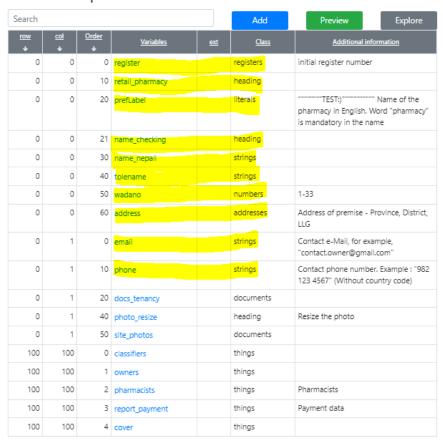| row ↓	col ↓	Order ↓	Variables	ext	Class	Additional information
0	0	0	register		registers	initial register number
0	0	10	retail_pharmacy		heading	
0	0	20	prefLabel		literals	"""""""""TEST:)"""""""""" Name of the pharmacy in English. Word "pharmacy" is mandatory in the name
0	0	21	name_checking		heading	
0	0	30	name_nepali		strings	
0	0	40	tolename		strings	
0	0	50	wadano		numbers	1-33
0	0	60	address		addresses	Address of premise - Province, District, LLG
0	1	0	email		strings	Contact e-Mail, for example, "contact.owner@gmail.com"
0	1	10	phone		strings	Contact phone number. Example : "982 123 4567" (Without country code)
0	1	20	docs_tenancy		documents	
0	1	40	photo_resize		heading	Resize the photo
0	1	50	site_photos		documents	
100	100	0	classifiers		things	
100	100	1	owners		things	
100	100	2	pharmacists		things	Pharmacists
100	100	3	report_payment		things	Payment data
100	100	4	cover		things	

Figure 5 Fields on the first page of the electronic form and their classes

In the example above, the EL Expression ${/email@literal} will access the string representation of the email field on the first page.

In case the field is not on the first page, the variable name should follow the page name to access fields on other pages. The delimiter is "/". For example, the EL Expression ${/classifiers/product@choice} will access the field "product" on the "classifier" page.

The printable representation of the form field, (i.e., @literal, @choice, etc.) depends on the class of the field. Refer to [**Data representation**] to select the printable representation for a field by the class.

A field in the current on-screen page

Sometimes, it will be necessary to inject into a printable document a field from a page that is displayed, but not saved yet. The possible cases are:

- The register number is just calculated and should be included in the printable document
- A field value is just defined by the user and should be included in the printable document

The field name should follow the "**this**" word to indicate that the field data should be taken directly from the on-screen page. The delimiter is "/"

For example, the EL Expression ${this/prefLabel@literal} will inject into a printable document a string value of the field with the name "prefLabel"

The printable representation of the form field, (i.e., @literal, @choice, etc.) depends on the class of the field. Refer to [**Data representation**] to select the printable representation for a field of any given class.

Detailed data attached to an application

An application form can contain data input fields with the class "persons". It allows attach to the application detailed data using multi-page electronic forms. For example:

- Warehouses and their details
- Pharmacists and their details

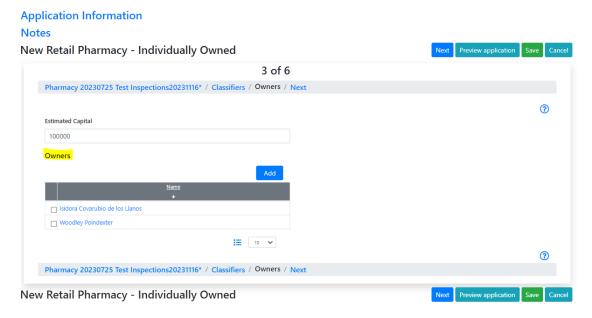The field with the class "persons" looks in the application form like this:

Figure 6 The field "Owners" allows adding detailed data about owners.

The detailed data is accessible by index in the table. For example:

- ${/owners/owners/0/@form} injects the whole details of Isidora Covarubio de Los Llanos
- ${/owners/owners/1/@form} injects the whole details of Woodley Poindexter
- ${/owners/owners/2/@form} doesn't inject data, however, will not raise an error

A page in the detailed data

Also, for the application form, see [**A page in the application**], it is possible to inject detailed page data using the page name:

- ${/owners/owners/0/details@form} injects only the details page of the owner Isidora Covarubio de Los Llanos
- ${/owners/owners/1/details@form} injects only the details page of the owner Woodley Poindexter
- ${/owners/owners/1/details@form} doesn't inject data, however, will not raise an error

resources.help.elreference

A field on a page in the detailed data

As well as for [**A field in an application's page**] it is possible to inject the value of any page in the detailed data. The variable name should follow:

1. The application page contains detailed data[3]
2. Index in the detailed data
3. Name of a page in the detailed data, or / for the first page

The delimiter is "/":

- ${/owners/owners/0/prefLabel@literal} injects the name of the owner - Isidora Covarubio de Los Llanos
- ${/owners/owners/0/details/fathername@literal} – inject the name of the father of Isidora Covarubio de Los Llanos
- ${/owners/owners/2/details/fathername@literal} does not inject data

The printable representation of the form field, (i.e., @literal, @choice, etc.) depends on the class of the field. Refer to [**Data representation**] to select the printable representation for a field of any given class.

---

[3] A field of class "persons"

resources.help.elreference

A field in an electronic form in the current workflow

Suppose, a printable document should include data collected in the current workflow. For example, inject the inspection date into the printable form of the registration certificate.

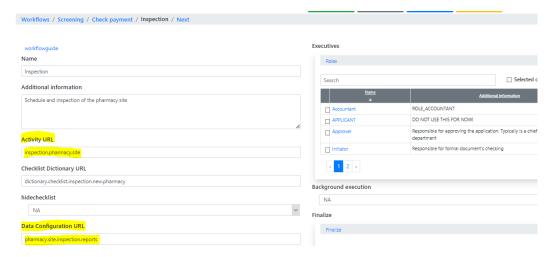The inspection date is defined in the inspection activity.

Figure 7 Activity configuration contains Activity URL and Data Configuration URL

- The activity URL defines the activity
- The Data Configuration URL defines the page to collect the data while the execution of the activity
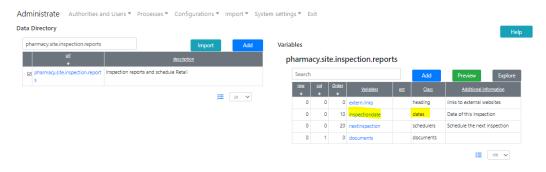

Figure 8 The page to collect data during the execution of the activity. Contains the inspection date field

The date of inspection is defined by the inspectiondate field.

The variable name should follow:

4. The word "process" indicates the current workflow as a Data Source
5. URL of the activity from which data is collected

The delimiter is "/":

For the example above:

${process/inspection.pharmacy.site/inspectiondate@date}

Where

- **process** indicates that the data source is the current workflow
- **inspection.pharmacy.site** is the Inspection activity URL
- **inspectiondate** The field in the activity data form

resources.help.elreference

Current date and user

To inject today's date and the current user's name the following EL Expressions are recommended[4]

EL Expression	Data	Example
${/@today}	Today's date AD Gregorian calendar in the current locale Medium format	${/@today} will return 08.03.2023
${/@todayBS}	Today's date BS Nepali calendar in short format, Arabic numbers	${/@todayBS} will return 2079-11-24
${/@todayBS1}	Today's date BS Nepali calendar in short format, Nepali numbers	${/@todayBS1} will return २०७९-११-२४
${/@author}	The name of the current user	${/@author} will return Phelim O'Neal

---

[4] Current Date and User Name can provide Office Software, however it is a proprietary feature that should be avoided

Data representation

addresses

Addresses allow users to select an administrative unit in a country and the geographical coordinates of a place inside this unit.

Data Representation	Printable form	Example	/	this	process
@gis	Geographical coordinates selected	27.703430;85.318418	+	+	+
@office	NMRA office that serves this address	South Branch Office	+	+	+
@choice	All administrative units selected	Province, District, Municipality	+	+	+
@level1	The highest administrative unit selected	Province	+	+	+
@level2	An administrative unit inside the level1	District	+	+	+
@level3	An administrative unit inside the level	Municipality	+	+	+
@level4	An administrative unit inside the level	Neighborhood (rarely used)	+	+	+

atc

Not implemented yet.

Workaround 1. Use the "copy to prefLabel" configuration feature and get the selected value from the "prefLabel" literal

Workaround 2. Use "links" to select ATC

dates

Date input control.

Data Representation	Printable form	Example	/	this	process
@date	Date AD Gregorian calendar in the current locale Medium format	Mar 12, 2023	+	+	+
@dateBS	Date BS Nepali calendar in short format, Arabic numbers	2032-01-01	+	-	+
@dateBS1	Date BS Nepali calendar in short format, Nepali numbers	२०३२-०१-०१	+	-	+
@years	Full AD Gregorian calendar years passed from a date given to the current date	12	+	-	+
@yearsBS	Full BS Nepali calendar years passed from a date given to the current date in Arabic numbers	12	+	-	+
@yearsBS1	Full BS Nepali calendar years passed from a date given to the current date in Nepali numbers	१२	+	-	+

dictionaries

Allows selection from the pre-defined choices. The selections may be single or multiple. The predefined choices may be single-level or multi-level.

Data Representation	Printable form	Example	/	this	process
@choice	The current user's choice or choices	Province, District, Municipality for single choice or Province, District/Munitipality1, Munitipality2, Municipality 3 for multi-choice	+		+
@level1	The first level of choice	Province For single-level dictionary the same as the @choice	+		+
@level2	The choice inside the first level	District	+		+
.. Up to 9 levels			+		+
@office	In the case of the dictionary.admin.units the office that servers the unit selected	South Branch Office	+		+

documents

Documents that have been uploaded by a user.

Data Representation	Printable form	Example	/	this	process
@filename	File names divided by "/"	mauricio.jpg/backtofuture.txt	+	-	+
@preflabel	File purpose or human readable names divided by "/"	State register certificate/Taxpayer certificate	+	-	+
@description	File descriptions divided by "/"	Form # 64. Allowance to do business in a country/Form # 63. Proof that the taxpayer number has been assigned	+	-	+
@choice		mauricio.jpg, backtofuture.txt	+	-	+
@image	Image if the first file is graphical		+	-	+

droplist

A droplist is a simplified representation of a one-level, single-choice dictionary.

Data Representation	Printable form	Example	/	this	process
@label	The selected value	Diploma in Pharmacy	+	+	+
@description	The description of the selected value	Proficiency Certificate Level in Pharmacy	+	+	+

heading

Displays a label or a link to an external resource.

The EL expressions are not appropriate for this class.

intervals

An interval between two dates.

Data Representation	Printable form	Example	/	this	process
@from	Begin of the interval AD Gregorian, the current locale middle format	16.02.2023	+	-	+
@fromBS	Begin of the interval BS Nepali short format, Arabic numbers	2079-11-04	+	-	+
@fromBS1	Begin of the interval BS Nepali short format, Nepali numbers	२०७९-११-०४	+	-	+
@to	End of the interval AD Gregorian, the current locale middle format	28.02.2023	+	-	+
@toBS	End of the interval BS Nepali short format, Arabic numbers	2079-11-16	+	-	+
@toBS1	End of the interval BS Nepali short format, Nepali numbers	२०७९-११-१६	+	-	+

resources.help.elreference

legacy

Allows to select objects uploaded from the legacy data. Applicable for manual legacy data export, usage of the legacy data for new applications, etc.

The EL expressions are not appropriate for this class.

links

This component allows the usage of data from other applications. This component allows multi-choice; thus, the index should be used.

Suppose the first page of the application contains a "links" component "manufacturer". Regardless of this component configuration for multi-choice, the EL should be /manufacturers/0/@link_role

Data Representation	Printable form	Example	/	this	process
@link_role	The selected value	Main unit	+	-	+

All fields in the linked component are accessible the same way as for the "persons" component, e.g., /manufacturers/0/prefLabel@literal. See [persons] for details.

literals

The text input field. The "literals" allow distinct text for each language.

Data Representation	Printable form	Example	/	this	process
@literal	Text from the input field	New Retail Pharmacy	+	+	+

logical

Yes/No/NA logical field

Data Representation	Printable form	Example	/	this	process
@choice	The current user's choice	Yes	+	-	+

numbers

Number input field

Data Representation	Printable form	Example	/	this	process
@number	An input field for numbers	8 000	+	+	+
@numberBS	The number from the input field, Nepali numbers	८,०००	+	-	+

persons

Refer to:

[**Detailed data attached to an application**] to inject the whole data

[**A page in the detailed data**] to inject a page

[**A field on a page in the detailed data**] to inject a value of the field

registers

Provides assigning numbers, registration dates, and expiration dates like a typical filing system register.

Data Representation	Printable form	Example	/	this	process
@literal	Register number	123456789	+	+	+
@registered	Registration date AD Gregorian, current locale middle format	17.02.2023	+	+	+
@expired	Expiration date AD Gregorian, current locale middle format	22.09.2023	+	+	+
@registeredBS	Registration date BS Nepali short format, Arabic numbers	2079-11-05	+	+	+
@expiredBS	Expiration date BS Nepali short format, Arabic numbers	2080-06-05	+	+	+
@registeredBS1	Registration date BS Nepali short format, Nepali numbers	२०७९-११-०५	+	+	+
@expiredBS1	Expiration date BS Nepali short format, Nepali numbers	२०८०-०६-०५	+	+	+

resources

Download documents, pictures, and templates.

The EL expressions are not appropriate for this class.

schedulers

Allows to run another follow-up application after the application is approved. Examples are "scheduled renewal payment", "scheduled inspection", etc.

Data Representation	Printable form	Example	/	this	process
@date	The current user's choice	23.02.2024	+	+	+

strings

The text input field. The "strings" keep the same text for all languages.

Data Representation	Printable form	Example	/	this	process
@literal	Text from the input field	Additional comments	+	+	+

things

An electronic form that placed at the right of the main form in the user's application.

Data Representation	Printable form	Example	/	this	process
@form	The whole electronic form	See [**Error! Reference source not found.**]	+	+	+

Unfortunately, currently, it is impossible to render the main page as a printable document.

Errors

To collect EL expression errors, place the special EL expression ${_ERROR_} at the end of the document

An example of an error message is

[choice: valueToString. Wrong data type choice/[label, description]]; [office: valueToString. Wrong data type office/[label, description]]

It will be a good idea not to place this EL into the production version.

Tools notes

Usually, vendors implement Open Office XML specifications strictly. In our opinion, the strictest implementation is the Microsoft Word issued after 2019. Also, we've tested it using Open Office.

Please note, that any office tools provide proprietary features like fields, improved layouts, etc. Please, avoid usage of them. To make the layout it will be a good idea to use plain tables instead of frames, anchors, etc.