

Functions

- [Set](#)
- [Prop](#)
 - [Specification](#)
 - [Notation](#)
 - [Example](#)
- [Image Loader](#)
- [QRCode](#)

Set

This function allows you to create new variables that will be treated as fields (mappings), and then use it to change the values of those variables. The new value passed to the Set function can be plain text, an existing field (mapping), or a Function field.

Definition:

```
${set(<variable name>,<new value>)}
```

Expand to see the examples on sample code

Define a variable "count" with value "0" and then use it as a normal field.

```
${set(count,0)}
${count}
```

Define a variable "newKey" with the value of field \${Key} and then use it as a normal field.

```
${set(newKey,${Key})}
${newKey}
```

Define a variable "TodayDate" with the value of the current date calculated with Javascript and then use it as a normal field

```
${set(TodayDate,%{(new Date()).getDate() + "/" + ((new Date()).getMonth()+1) + "/" + (new Date()).getFullYear()})}
${TodayDate}
```



Note

The function Set can be also used to change the value of existing fields (i.e., existing mappings), but it has a limitation: all occurrences of that field in the document will be replaced with the value of the latest Set that is applied.



Note

The values of all the variables (mappings) created with the Set function and used on the header/footer of a template will correspond to the last value passed to the Set function.

Prop

This function allows you to define properties that will be used later by Document Generator during document processing. Usually, integrations require some extra configuration such as security tokens, API keys, and others.

Specification

Function Name	prop
First Parameter	Key
Second parameter	Value

Notation

```
${prop(Key,Value)}
```

Example

```
${prop(my.first.property.key, qwerty12345)}  
${prop(service.api.key, 84f7970c-5f70-47d9-a4f2-c75151820ba0)}  
${prop(service.api.username, userAdmin)}
```

Image Loader

This function basically loads an image from a URL and puts that on the document. You can also specify the width and/or height of the images.

Definition:

```
!{<URL>}
```

Below are two examples:

Image from template ! {url width=200}

!{https://www.google.pt/images/srpr/logo11w.png width=200}

or

Images of Execution Evidences

!{\$TestRuns[n].ExecutionEvidences[d].FileURL}

After exporting the document, the generated file is going to see this:

Image from template ! {url width=200}



QRCode

This function allows you to create a QR code image on the template. It accepts as content other Document Generator mappings.

Definition:

```
${qrcode(' <content> ', (Optional) <width>, (Optional) <type> )}
```

Expand to see the examples on sample code

Draws a QR Code image with the content Hello World.
`${qrcode('Hello World')}`

Draws a QR Code image with the Issue Key as content.
`${qrcode('${Key}')}`

Draws a QR Code image with a Javascript result as content.
`${qrcode('%{'${Key}'.substring(0,2)}')}`

Draws a QR Code image with the content noreply@xpanse-it.com and the QR Code type email.
`${qrcode('noreply@xpanse-it.com',email)}`

Draws a QR Code image with the content Hello World and width size 200px.
`${qrcode('Hello World',200)}`

Draws a QR Code image with the content Hello World, width size 200px and the QR Code type email.
`${qrcode('hugo.freixo@xpanse-it.com',200,email)}`

QR Code Content

You can use any mapping on the content of the QR Code function, **but you must be careful because Markups won't generate the expected result.**

If you try to export `${qrcode('${wiki:Description}')}` or `${qrcode('${html:Description}')}`, the content of the QR Code won't be correct.

You can apply format mappings such as `dateformat`, `numberformat`, `fullname` or `emailaddress`.

QR Code Types



You can learn more about QR Code types [here](#). Note that these types may depend on the QR Code reader and the obtained result may be different than expected.

- **Text** - Plain text. It's the same as not having a type defined.
- **URL** - The content will be read as a URL link.
- **Phone** - The content will be read as a phone number.
- **Geolocation** - The content will be read as GPS coordinates. Content example: "41.714316, -8.811993".
- **Email** - The content will be read as an email.
- **Wifi** - The content will be read as a Wifi connection. Content example: T:WPA;S:NetworkName;P:NetworkPassword;
- **VCard** - The content as a VCard. Please note that not all devices are prepared to understand VCard details. More information [here](#).

QR Code Width

The width of the QR Code must be a value between 0 and 409 (pixels).

If the user introduces a value outside this interval, the width will be set to its default value (177px).

The user must choose a width large enough for the amount of content in the QR Code. Usually, 100px - 200px is enough.