

# RML Example 21: Dynamic



RML (Report Markup Language) is ReportLab's own language for specifying the appearance of a printed page, which is converted into PDF by the utility rml2pdf.

These RML samples showcase techniques and features for generating various types of output and are distributed within our commercial package as test cases. Each should be self explanatory and stand alone.

---

## Dynamic RML

This adds dynamic capabilities. Say that again a few times with a LOOP tag..

```
pass 0: this adds dynamic capabilities
pass 1: this adds dynamic capabilities
pass 2: this adds dynamic capabilities
pass 3: this adds dynamic capabilities
pass 4: this adds dynamic capabilities
```

The author's name was assigned earlier, and it is: Andy Robinson. Here is the some info about this process:

```
sys.version: 2.7.3 (default, May 8 2012, 10:11:50)
[GCC 4.4.3]
cwd: /home/rptlab/tmp/make-rml-examples.4V2v/rlextra/tests/rml2pdf
location: /home/rptlab/tmp/make-rml-examples.4V2v/py27/lib/python2.7/site-packages/reportlab
version: 3.4.26
date: 20180209
now: Sun Feb 18 20:19:05 2018
```

## Data acquisition

This document uses a data tag to slurp in a source XML document in "sample\_invoice.xml". The source is here...(in fact, this could be useful once I have the escaping sorted out....)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<invoice>
  <customerDetails>
    <customerName>Bats Ltd.</customerName>
    <address1>156 The Belfry</address1>
    <address2>Ghastly</address2>
    <county>Lincolnshire</county>
    <postCode>LN1 3FD</postCode>
  </customerDetails>
  <totalNet>400.00</totalNet>
  <totalVat>70.00</totalVat>
  <totalGross>470.00</totalGross>
  <invoiceNo>00134</invoiceNo>
  <dueDate>31 December 2003</dueDate>
  <lines>
    <item qty="10" unitPrice="£30.00">sprockets</item>
    <item qty="10" unitPrice="£4.00">widgets</item>
    <item qty="5" unitPrice="£12.00">thingummies</item>
  </lines>
</invoice>
```

This can be referred to with the dynamic tags. Let's pull out the customer name from a node in the file: **Bats Ltd.** . Now let's do a loop over some elements...

Description	Unit Price	Qty	Line Price
sprockets	£30.00	10	etc
widgets	£4.00	10	etc
thingummies	£12.00	5	etc

# RML Example 21: Dynamic



## Doclets

This is a test of doclets. Doclets are a new part of the framework in 2005 and aim to allow creation of smart "reusable objects" which know how to render themselves at a lower level.

Doclets know how to pull data from a data model, and draw themselves in response.

A simple doclet is provided in the class `rlxtra.rml2pdf.doclet.TestReverseDoclet`. This draws a little table with a string (which is the data passed in) and the same string reversed. We can pass it a single (quoted) string in an attribute:

Reverse Doclet Example	
Normal:	<b>Cowabunga!</b>
Reversed:	<b>!agnubawoC</b>

The key thing is that the data parameter is a variable name in the dynamic RML namespace. If absent, the doclet sees the whole namespace. So we can use doclets inside loops and conditionals and pass through actual variables to them - not just flattened-out strings. This example shows dynamic use; it should generate reverse doclets for each inner planet name:

Reverse Doclet Example	
Normal:	<b>Mercury</b>
Reversed:	<b>yrucreM</b>

Reverse Doclet Example	
Normal:	<b>Venus</b>
Reversed:	<b>suneV</b>

Reverse Doclet Example	
Normal:	<b>Earth</b>
Reversed:	<b>htraE</b>

Reverse Doclet Example	
Normal:	<b>Mars</b>
Reversed:	<b>sraM</b>

"Yeah, right, big deal", I hear you say. But consider that the doclet might know how to connect to a customer database in a fund manager and plot a table and chart of the investor's holdings, given just a customer ID and an open database connection as parameters.: