RML Example 31: Codepages

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 ouput and are distributed within our commercial package as test cases. Each should be self explanatory and stand alone.

Unicode Tests

In version 2.0 of the ReportLab library and later, all RML input is strictly UTF8 - the default standard for XML files. This considerably simplifies things compared to version 1.0, where you had to encode data to match the fonts.

Most good editors can be configured to display UTF8 characters. To see non-Latin characters, you'll need to make sure you use a multilingual font.

Brazilian (Portuguese)	English
maçã	apple
abóbora	pumpkin
pé	foot
limão	
limões	lemons
maracujá	passion fruit

The above table should display the names of various fruits, which contain accents. These are in Helvetica, a Type 1 font.

This paragraph is in Helvetica, a Type 1 font. You should see trademark (™), copyright (©) and registered (®) characters. Ole!

This paragraph is in Vera, a TrueType font. You should see trademark (™), copyright (©) and registered (®) characters. Ole!

This paragraph is in VeraItalic, a TrueType font. You should see trademark (™), copyright (©) and registered (®) characters. Ole!

Characters in UTF8 can always be represented with XML numeric entity codes. Thus, the character 'A' can be shown with the decimal escape sequence "A" in the document source, or the hex sequence "A". Here we go, should get three As: A A A. Trademark can be written as ™, copyright as ©, and registered as ®.

There are some other non-paragraph contexts for text display - drawing strings directly, and placing strings (not paragraphs) in table cells. The title at the top should have a ® symbol at the right, and all three should appear in the table below:

Symbol Name	Displays
Copyright	©
Registered	®
Trademark	™
En Dash	–
Em Dash	—

Test 31 does Unicode with Japanese characters, but you'll need the Adobe font pack to read that one.

This uses the unichar tag to render an umlaut:ü

This uses the unichar tag to render a bullet:•

This uses a numeric entity to render a bullet •

This uses inline utf8 to render a bullet •