ParaTools 1.00 Documentation

Mike Jewell

September 2, 2004

Contents

Chapter 1

Introduction

1.1 What is ParaTools?

ParaTools, short for ParaCite Toolkit, is a collection of Perl modules for reference parsing that is designed to be easily expanded and yet simple to use. The parsing modules make up the core of the package, but there are also useful modules to assist with OpenURL creation and the extraction of references from documents. The toolkit is released under the GNU Public License, so can be used freely as long as the source code is provided (see the COPYING file in the root directory of the distribution for more information).

The toolkit came about as a result of the ParaCite resource, a reference search engine located at http://paracite.eprints.org, which uses a template-based reference parser to extract metadata from provided references and then provides search results based on this metadata. Biblio::Document::Parser is an offshoot from ParaTools, which specialises in document parsing.

1.2 Who should use ParaTools?

The ParaTools package has many applications, including:

- Converting reference lists into valid OpenURLs
- Converting existing metadata into valid OpenURLs
- Collecting metadata from references to carry out internal searches
- Extracting reference lists from documents
- Carrying out searches using ParaCite

The modularity of ParaTools means that it is very easy to add new techniques (and we would be very pleased to hear of new ones!).

1.3 What will it run on?

ParaTools should work on any platform that supports Perl 5.6.0 or higher, although testing was primarily carried out using Red Hat Linux 7.3 with Perl 5.6. Where possible platform-agnostic modules have been used for file functionality, so temporary files should be placed in the correct place for the operating system. Memory requirements for ParaTools are minimal, although the template parser and document parser will require more memory as the number of templates and sizes of documents increase.

1.4 This Documentation

This documentation is written in perl POD format and converted into Postscript (which is 2 pages to a sheet for printing), ASCII, PDF, and HTML.

The latest version of this documentation can be obtained from http://paracite.eprints.org/files/doc

Chapter 2

Required Software

2.1 What software does Biblio::Document::Parser need?

Perl Modules

The ParaTools::Utils module provides functions to retrieve and convert files both on the Internet and on a local file-system. The former requires a few extra Perl modules to function:

LWP::Simple and LWP::UserAgent

These are Perl modules that provide an interface to the World Wide Web, and are used by the ParaTools Document::Parser to retrieve documents from the Internet.

File::Temp

This module handles temporary files across multiple platforms.

There are also some dependencies for the above modules, including MIME::Base64, HTML::TagSet, and Digest::MD5.

All of the above are available at http://paracite.eprints.org/files/perlmods/. Although these are not guaranteed to be the most recent versions, they are the versions that ParaTools has been tested with. For the most recent releases, the Perl modules can also be found at http://www.cpan.org.

Installing Perl Modules

This describes the way to install a simple perl module, some require a bit more effort. We will use the non-existent FOO module as an example.

Unpack the archive:

```
% gunzip FOO-5.23.tar.gz
% tar xf FOO-5.23.tar
```

Enter the directory this creates:

```
% cd FOO-5.23
```

Run the following commands:

```
% perl Build.PL
% ./Build
% ./Build test
% ./Build install
```

Document Converters

These programs are used by the Biblio::Document::Parser::Utils module to convert documents to ASCII from other formats. If you would like to add other formats, see the HOWTO later in this manual.

wvText

This is part of the wvWare package, and provides a command to convert Word documents into ASCII, as well as into other formats.

wvWare is available from: http://www.wvware.com/wvWare.html

pdftotext

This is provided with xpdf, and can convert PDF to ASCII.

Xpdf is available from: http://www.foolabs.com/xpdf/download.html

pstotext

pstotext is a program that works with GhostScript to convert PS and PDF files to ASCII.

pstotext is available from: http://www.research.compaq.com/SRC/virtualpaper/pstotext.html

GhostScript is available from: http://www.cs.wisc.edu/~ghost/

links

Links is an excellent ASCII web browser that can display complex pages with tables and frames. It also has a very effective ASCII dump option, which ParaTools::Utils uses to convert HTML to ASCII.

Links is available from: http://artax.karlin.mff.cuni.cz/~mikulas/links/

Chapter 3

How to Install Biblio::Document::Parser

3.1 Installation

First unpack the Biblio::Document::Parser archive:

```
% tar xfvz <packagename>.tar.gz
```

Move into the unpacked folder, and then do the following:

```
% perl Build.PL
% ./Build
```

You can optionally run

```
% ./Build test
```

which will carry out a few checks to ensure everything is working correctly. Finally, become root and do:

```
% ./Build install
```

This will install the modules and man pages into the correct locations.

Chapter 4

Troubleshooting

4.1 Troubleshooting

If you cannot find a solution to your problem here, make sure you are using the latest version of the toolkit and ask on the ParaTools mailing list (see http://paracite.eprints.org/developers/).

Chapter 5

How-To Guides

5.1 HOW TO: Modify Converters in Document::Parser::Utils

- Locate where your Utils.pm file has been installed.

  On Linux systems this should just involve doing 'locate Utils.pm', otherwise 'find / -name Utils.pm' should work. Alternatively, you can edit the Utils.pm in the Document/Parser/ directory of an unpacked distribution, and install it once you have finished.

- Add the converter to the list.

  If you are editing an already installed Utils.pm file you will probably have to be root to do this. If you are editing the Utils.pm inside an unpacked distribution, you will have to reinstall the modules once you are finished (see the Installation section).

  The %CONVERTERS hash maps from file extension to converter - _IN_ is replaced by the input file, _OUT_ is replaced by the output (ASCII) file. For example:

  ```
  html => "links --dump _IN_ > _OUT_"
  ```

  This takes an input HTML file (say, in.html) and an output ASCII file (out.txt), and carries out 'links –dump in.html > out.txt'.

  NB: Don't forget a comma after your converter.

HOW TO: Create a Document Parser

All new document parsers should be named Biblio::Document::Parser::SomeName, where SomeName is replaced with a unique name (ideally the author's surname). The parser should extend the Biblio::Document::Parser module like so:

```
package Biblio::Document::Parser::SomeName;
require Exporter;
@ISA = ("Exporter", "Biblio::Document::Parser");
our @EXPORT_OK = ( 'new', 'parse' );
```

You should then override the 'new' and 'parse' methods:
e.g.

```
sub new
{
        my($class) = @_;
        my $self = {};
        return bless($self, $class);
}

sub parse
{
       my($self, $lines, %options) = @_;

       # Do something with the lines
       my @lines = split("\n", $lines);
       my @references = get_refs(@lines);
       return @references;
}
```

This makes it easy for users to swap out one document parser for another.

Chapter 6

Problems, Questions and Feedback

6.1 Bug Report Policy

There is currently no online bug tracking system. Known bugs are listed in the BUGLIST file in the distribution and a list will be kept on the http://paracite.eprints.org/developers/ site.

If you identify a bug or "issue" (issues are not bugs, but are things which could be clearer or better), and it's not already listed on the site, please let us know at paracite@ecs.soton.ac.uk - include all the information you can: what version of Biblio::Document::Parser (see VERSION if you're not sure), what operating system etc.

6.2 Where to go with Questions and Suggestions

There is a mailing list for ParaTools (encompassing Biblio::Document::Parser) which may be the right place to ask general questions and start discussions on broad design issues.

To subscribe send an email to majordomo@ecs.soton.ac.uk containing the text

```
subscribe paratools
```