

Practical Semantic Web and Linked Data Applications

Common Lisp Edition

Uses the Free Editions of Franz Common Lisp and AllegroGraph

Mark Watson

Copyright 2010 Mark Watson. All rights reserved.
This work is licensed under a Creative Commons
Attribution-Noncommercial-No Derivative Works
Version 3.0 United States License.

November 3, 2010

Contents

Preface	xi
1. Getting started	xi
2. Portable Common Lisp Code Book Examples	xii
3. Using the Common Lisp ASDF Package Manager	xii
4. Information on the Companion Edition to this Book that Covers Java and JVM Languages	xiii
5. AllegroGraph	xiii
6. Software License for Example Code in this Book	xiv
1. Introduction	1
1.1. Who is this Book Written For?	1
1.2. Why a PDF Copy of this Book is Available Free on the Web	3
1.3. Book Software	3
1.4. Why Graph Data Representations are Better than the Relational Database Model for Dealing with Rapidly Changing Data Requirements	4
1.5. What if You Use Other Programming Languages Other Than Lisp?	4
2. AllegroGraph Embedded Lisp Quick Start	7
2.1. Starting AllegroGraph	7
2.2. Working with RDF Data Stores	8
2.2.1. Creating Repositories	9
2.2.2. AllegroGraph Lisp Reader Support for RDF	10
2.2.3. Adding Triples	10
2.2.4. Fetching Triples by ID	11
2.2.5. Printing Triples	11
2.2.6. Using Cursors to Iterate Through Query Results	13
2.2.7. Saving Triple Stores to Disk as XML, N-Triples, and N3	14
2.3. AllegroGraph's Extensions to RDF	14
2.3.1. Examples Using Triple and Graph IDs	15
2.3.2. Support for Geo Location	16
2.3.3. Support for Free Text Indexing	19
2.3.4. Comparing AllegroGraph With Other Semantic Web Frame- works	20
2.4. AllegroGraph Quickstart Wrap Up	21

I. Semantic Web Technologies	23
3. RDF	25
3.1. RDF Examples in N-Triple and N3 Formats	27
3.2. The RDF Namespace	30
3.2.1. rdf:type	30
3.2.2. rdf:Property	31
3.3. Dereferenceable URIs	31
3.4. RDF Wrap Up	32
4. RDFS	33
4.1. Extending RDF with RDF Schema	33
4.2. Modeling with RDFS	34
4.3. AllegroGraph RDFS++ Extensions	36
4.3.1. owl:sameAs	37
4.3.2. owl:inverseOf	37
4.3.3. owl:TransitiveProperty	38
4.4. RDFS Wrapup	38
5. The SPARQL Query Language	41
5.1. Example RDF Data in N3 Format	41
5.2. Example SPARQL SELECT Queries	44
5.3. Example SPARQL CONSTRUCT Queries	46
5.4. Example SPARQL ASK Queries	46
5.5. Example SPARQL DESCRIBE Queries	46
5.6. Wrapup	47
6. RDFS++ and OWL	49
6.1. Properties Supported In RDFS++	49
6.1.1. owl:sameAs	50
6.1.2. owl:inverseOf	50
6.1.3. owl:TransitiveProperty	51
6.2. RDF, RDFS, and RDFS++ Modeling Wrap Up	51
II. AllegroGraph Extended Tutorial	53
7. SPARQL Queries Using AllegroGraph APIs	55
7.1. Using Namespaces	55
7.2. Reading RDF Data From Files	56
7.3. Lisp APIs for Queires	56
7.4. Wrap Up	58
8. AllegroGraph Reasoning System	59
8.1. Enabling RDFS++ Reasoning on a Triple Store	59

8.2. Inferring New Triples: rdf:type vs. rdfs:subClassOf Example	60
8.3. Using Inverse Properties	61
8.4. Using the Same As Property	63
8.5. Using the Transitive Property	63
8.6. Wrap Up	65
9. AllegroGraph Prolog Interface	67
III. Portable Common Lisp Utilities for Information Processing	71
10. Linked Data and the World Wide Web	73
10.1. Linked Data Resources on the Web	74
10.2. Publishing Linked Data	74
10.3. Will Linked Data Become the Semantic Web?	75
10.4. Linked Data Wrapup	75
11. Common Lisp Client Library for Open Calais	77
11.1. Open Calais Web Services Client	77
11.2. Storing Entity Data in an RDF Data Store	80
11.3. Testing the Open Calais Demo System	81
11.4. Open Calais Wrap Up	82
12. Common Lisp Client Library for Natural Language Processing	85
12.1. KnowledgeBooks.com Natural Language Processing Library	85
12.2. KnowledgeBooks Natural Language Processing Library Wrapup	87
13. Common Lisp Client Library for Freebase	89
13.1. Overview of Freebase	89
13.2. Accessing Freebase from Common Lisp	91
13.3. Freebase Wrapup	93
14. Common Lisp Client Library for DBpedia	95
14.1. Interactively Querying DBpedia Using the Snorql Web Interface	95
14.2. Interactively Finding Useful DBpedia Resources Using the gFacet Browser	97
14.3. The lookup.dbpedia.org Web Service	97
14.4. Using the AllegroGraph SPARQL Client Library to access DBpedia	99
14.5. DBpedia Wrapup	100
15. Library for GeoNames	101
15.1. Using the cl-geonames Library	101
15.2. Geonames Wrapup	103

IV. Example Semantic Web Application	105
16. Semantic Web Portal Back End Services	107
16.1. Implementing the Back End APIs	108
16.2. Unit Testing the Backend Code	110
16.3. Backend Wrapup	112
17. Semantic Web Portal User Interface	113
17.1. Portable AllegroServe	113
17.2. Layout of CLP files for Web Application	113
17.3. Common Lisp Code for Web Application	114
17.4. Web Application Wrap Up	118

List of Figures

- 1.1. Example Semantic Web Application 2

- 14.1. DBpedia Snorql Web Interface 96
- 14.2. DBpedia Graph Facet Viewer 98
- 14.3. DBpedia Graph Facet Viewer after selecting a resource 98

- 17.1. Example Semantic Web Application Login Page 114
- 17.2. Example File Upload Page 116
- 17.3. Example Application Search Page 118

List of Tables

13.1. Subset of Freebase API Arguments 90

Preface

This book is primarily intended to be a practical guide for using RDF data in information processing, linked data, and semantic web applications using the Common Lisp APIs for the AllegroGraph product. A second use for this book is to help you, the reader, set up an interactive Lisp development environment for writing knowledge intensive applications. So, while the Semantic Web applications using the AllegroGraph RDF data store is the main theme in this book, I will also cover using data from a variety of sources like Freebase, DBpedia and other public RDF repositories, use of statistical Natural Language Processing (NLP), and the GeoNames database and public web service.¹

1. Getting started

I expect you to install the Franz Free Edition Common Lisp environment with the Free Edition of AllegroGraph to work through the examples in this book. If you own professional or enterprise licenses for these projects you can use that also. The free editions have restrictions on their use so read the license agreement.

Download the Free Lisp Edition and then install AllegroGraph using:

```
(require :update)
(system.update:install-allegrograph)
```

Whenever you start Lisp using evaluate the following form to load AllegroGraph:

```
(require :agraph)
```

I do not duplicate information in this book that appears in the documentation available on the Franz web site. I urge you to read how to set up an Emacs development environment.

¹The geonames.org web service is limited to 2000 queries per hour from any single IP address. Commercial support is available, or, with some effort, you can also run GeoNames on your own server.

2. Portable Common Lisp Code Book Examples

Even though many examples in this book use AllegroGraph I also provide many portable Common Lisp examples and utilities that I have written for my own work and research:

1. KnowledgeBooks.com Lisp Natural Language Processing (NLP) library
2. Client library for using the Open Calais web service²³
3. Example code for using the Freebase web service⁴
4. Example code for using DBpedia web services⁵
5. Example code for using GeoBase.org web services⁶

3. Using the Common Lisp ASDF Package Manager

There are several package managers available for Common Lisp and I have chosen to use ASDF for the examples in this book that are not self contained in a single source directory. ASDF uses a search path list as a source of directories to find package definition files (that end with the extension *.asd*). For instance, since some of the examples in this book will need to make web service calls I have an example directory *aserve_client* to show you how to use Franz's open source **aserve** client library. The example code needs to use the Yason JSON parsing and generation library in *utils/yson*:

```
(push "../utils/yason/" asdf:*central-registry*)  
(asdf:operate 'asdf:load-op 'yson)
```

The first statement pushes the Yson package directory on the ASDF search path list and the second line loads the package named *yson*.

²Requires the open source Portable AllegroServe and split-sequence libraries.

³The example program that puts entities found in text into an RDF data store requires the AllegroGraph library.

⁴Requires the Franz Portable AllegroServe client library. This is open source, but you will need to manually install Portable AllegroServe if you are using an alternative Common Lisp implementation (e.g., SBCL or Clozure Common Lisp).

⁵Requires the AllegroGraph SPARQL client APIs but this code could be rewritten.

⁶Requires several open source libraries that are included in the ZIP file for this book's examples: cl-json, s-xml, split-sequence, usocket, trivial-gray-streams, flexi-streams, chungu, cl-base64, puri, drakma, and cl-geonames.

4. Information on the Companion Edition to this Book that Covers Java and JVM Languages

When you browse through the directories containing the examples in this book you will notice that I use the convention of placing short snippets of test code that I often include in the book text in files named *test.lisp* and put longer examples in files named *example.lisp*.

This book is organized in layers:

1. Quick introduction to the AllegroGraph and Franz Lisp.
2. Theory (with some AllegroGraph-specific short examples).
3. Detailed treatment of AllegroGraph APIs.
4. Development of Useful Common Lisp libraries information processing, and importing linked data sources like Freebase and Open Calais data to an AllegroGraph RDF store.
5. Development of a complete web portal using Semantic Web technologies.

4. Information on the Companion Edition to this Book that Covers Java and JVM Languages

This book has a companion edition that covers the use of both AllegroGraph and the open source Sesame project using JVM based languages like Java, Clojure, JRuby, and Scala. If you primarily work with JVM languages then you will likely be better off working through the other edition of this book. The JVM edition of this book offers some portability: I provide the basic functionality of AllegroGraph for RDF storage, SPARL queries, Geo Location, and free text search using the Sesame RDF data store and my own search and Geo Location libraries.

The Free Java Edition of AllegroGraph does not have the commercial use restrictions that the Free Lisp Edition does.

If you want a free commercial friendly Lisp development and deployment environment: I recommend that you use the companion book with the Clojure programming language.

5. AllegroGraph

AllegroGraph is written in Common Lisp and comes bundled in several different products:

1. As a standalone server that supports Lisp, Ruby, Java, Clojure, Scala, Common Lisp, and Python clients. A free version (limited to 50 million RDF triples - a large limit) that can be used for any purpose, including commercial use.
2. The `WebView` interface for exploring, querying, and managing AllegroGraph triple stores. `WebView` is standalone because it contains an embedded AllegroGraph server.
3. The `Gruff` for exploring, querying, and managing AllegroGraph triple stores using table and graph views. `Gruff` is standalone because it contains an embedded AllegroGraph server.
4. A library that is used embedded in Franz Common Lisp applications. A free version is available (with some limitations) for non-commercial use.

This book uses AllegroGraph in embedded mode because this is the most agile way to experiment and learn the APIs. What you learn in this book is applicable to all of the AllegroGraph products. Currently, the server release is version 4.0 and the embedded library release is 3.3.

6. Software License for Example Code in this Book

The small example code snippets listed in this book text and my code for the larger example applications and libraries in the code ZIP file are licensed using the AGPL.⁷

Commercial waiver of some AGPL terms for people or organizations who purchase this book:

If you purchase the print edition or purchase the PDF file⁸ of this book then I grant you a partial commercial use waiver to the AGPL deploying your applications on a single server: you can use my examples in applications on a single server without the requirement of releasing the source code for your application under the AGPL (all other AGPL license terms apply). If you need to run an application using my code on multiple servers, then please purchase one copy of the book for each server.

I enjoy writing and purchasing copies of this book helps fund future writing projects.

Acknowledgements

I would like to thank my wife Carol Watson for copyediting this book.

⁷For your convenience, I include in the code ZIP file third party libraries, most of which are released under MIT, BSD, Lisp LGPL, or Apache licenses.

⁸downloading the free PDF from <http://markwatson.com/opencontent> does not give you the rights to this waiver.

1. Introduction

Franz has good online documentation¹ for all of their AllegroGraph products. One purpose of this book is to provide a brief introduction to AllegroGraph but I assume that you also reference the documentation on the Franz web site. The broader purpose of this book is to provide application programming examples using AllegroGraph and Linked Data sources on the web. This book also covers some of my own open source Common Lisp projects that you may find useful for Semantic Web applications. The combination of interactive Lisp development with embedded AllegroGraph and my utilities covered later should provide you with an agile development environment for writing knowledge based and semantic web applications.

AllegroGraph is an RDF data repository that can use RDFS and RDFS+ inferencing. AllegroGraph also provides three non-standard extensions:

1. Text indexing and search
2. Geo Location support
3. Network traversal and search for social network applications

1.1. Who is this Book Written For?

I assume that you both already know how to program in Common Lisp and that you write applications that require handling large amounts of unstructured information. AllegroGraph is a powerful tool for handling large amounts of data and Lisp programming environments are excellent for rapidly prototyping new applications. Along with extra libraries I have written for using linked data sources on the web, this book will hopefully provide you with new tools to rapidly solve application problems that would be more difficult to handle using relational databases.

Franz also provides support for embedding AllegroGraph in Lisp applications and for using it in a client mode with external AllegroGraph servers. Since the APIs are almost identical, I take a shortcut in writing this book and concentrate on using AllegroGraph in embedded mode.

¹<http://franz.com/agraph/support/documentation/current/agraph-introduction.html>

Typical Semantic Web Application

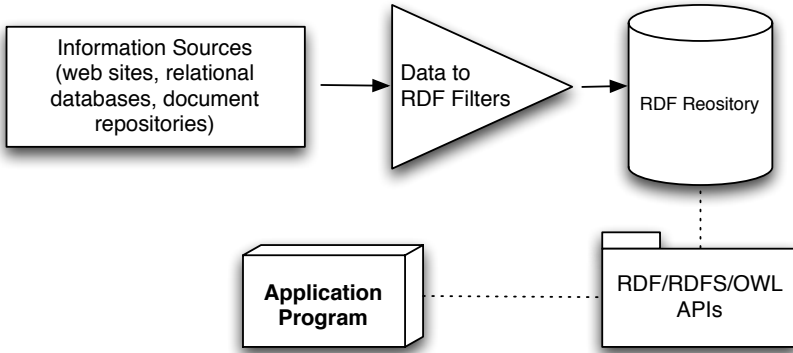


Figure 1.1.: Example Semantic Web Application

There are many books, good tutorials and software about the Semantic Web on the web. However, there is not a single reference for developers who want to use the combination of Common Lisp and AllegroGraph for development using technologies like RDF/RDFS/OWL modeling, descriptive logic reasoners, and the SPARQL query language.

If you own a Franz Lisp and AllegroGraph development license, then you are set to go. If not, you need to download and install a free edition copy at:

<http://www.franz.com/downloads/>

You may also want to download and install the free versions of the AllegroGraph standalone server, Gruff, and WebView.²

Franz Inc. has provided support for my writing this book in the form of technical reviews and my understanding is that even though you will need to periodically refresh your free non-commercial license, there is no inherent time limit for non-commercial use. I would also like to thank Franz for providing me with an Enterprise developers license for my MacBook that I use for my own research and development projects.

²I do not use these associated products in this book but I do in the Java, Clojure, Scala, and JRuby edition of this book.

1.2. Why a PDF Copy of this Book is Available Free on the Web

As an author I want to earn a living writing and have many people read and enjoy my books. By offering for sale the print version of this book I can earn some money for my efforts and also allow readers who can not afford to buy many books or may only be interested in a few chapters to read it from my web site. If you support my future writing projects by purchasing either the print or PDF version of this book, I thank you by offering you more flexibility in the software license terms for the example programs and libraries I developed (see Section 6 in the Preface).

Please note that I do not give permission to post the PDF version of this book on other people's web sites: I consider this to be at least indirectly commercial exploitation in violation the Creative Commons License that I have chosen for this book.

1.3. Book Software

You can download a large ZIP file containing all code and test data from the URL:

http://markwatson.com/opencontent/lisp_semantic_web_code.zip

The book example code, libraries, and applications are organized in subdirectories organized by topic:

1. dbpedia - use the DBPedia web services
2. freebase_client - use the Freebase web services
3. geonames - use the Geonames web service
4. knowledgebooks_nlp - my natural language processing library
5. opencalais - use the OpenCalais web services
6. quick_start_allegrograph_lisp_embedded - code snippets used to introduce Allegrograph
7. quick_start_allegrograph_standalone_server - code snippets for Chapter 2
8. rdf - additional code snippets for created RDF triples and making queries
9. reasoning - code snippets for Chapter 8
10. sparql - code snippets and sample data for SPARQL queries

Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

