

Developer Guide

Software architecture

The EBI-RDA linkage system is comprised of two main parts (Figure 1):

1. **Data extractor:** this system is a standalone Java Program (executable jar) that pulls query items from a local data store (MySQL database) and queries them (using web services) against the Uniprot and ENA databases made available by EBI.
2. **Landing page:** a website to allow visual display of the data. This was developed using JSP/Servlets and displays the relevant data that is stored in the MySQL database.

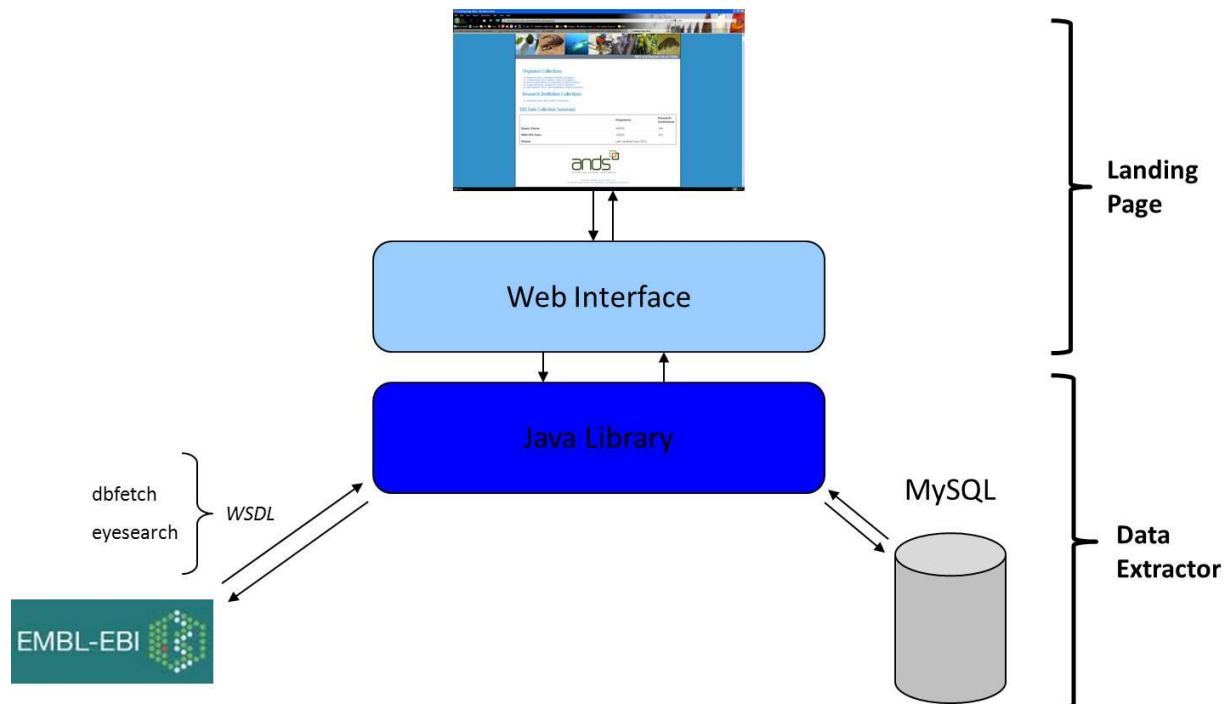


Figure 1: Software architecture overview

A more complete depiction of the software interfaces used is displayed in Figure 2:

- Public data repositories (Atlas of Living Australia, National Library of Australia) are used to generate lists of potential “Australian” query items (species and research institutions).

- These query items are used to query the European Nucleotide Archive (DNA sequences) and Uniprot (protein sequences) databases at EBI via web services.
- These results are stored in the Australian data collection repository (MySQL database)
- The data in the Australian data collection repository is visualised using the landing page.
- Data in the Australian data collection repository is used to create collections which are submitted to the RDA. These collections can be discoverable by interested parties, which links from the RDA to the landing page back to the original data stored in EBI databases.

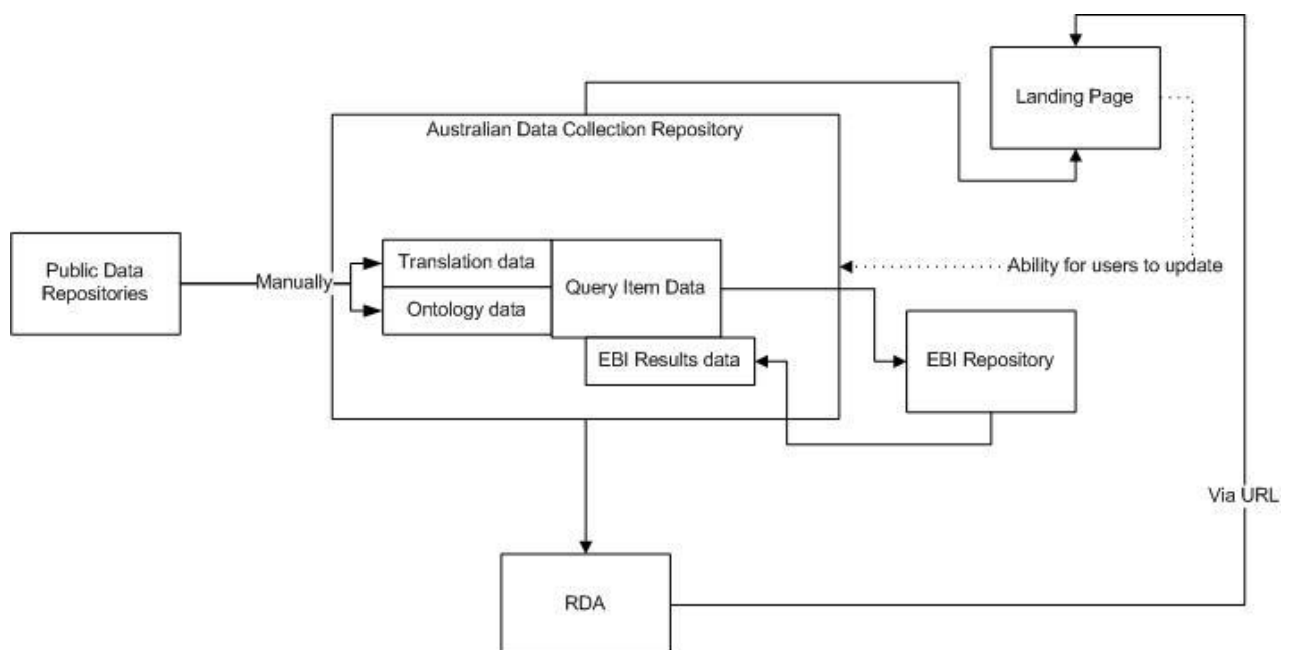


Figure 2: Software interfaces

Installation

The data extractor and landing page can be used and installed using the installation instructions available for download here: <http://ebi-rda-linkage.sourceforge.net/ebi-rda-linkage-installation-guide.pdf>

Development

- The system was developed as a Java Library that can be interfaced by any other application compatible with reading Java APIs.
- The data collection system is runnable by a Main class called Runner.
- For full details see JavaDoc: <http://sourceforge.net/projects/ebi-rda-linkage/files/doc.zip/download>
- The Landing Page web system is written in JSP that sources data from the MySQL database.

Dependencies

Use the following xml snippet to update your ivy.xml with the following dependencies

```
<dependencies>
```

```
<dependency org="junit" name="junit" rev="4.8.1">
```

```
  <artifact name="junit" ext="jar"/>
```

```
</dependency>
```

```
<dependency org="rifcs-api" name="rifcs-api" rev="1-2">
```

```
  <artifact name="rifcs-api" ext="jar"/>
```

```
</dependency>
```

```
<dependency org="EBeye_Axis1" name="EBeye_Axis1" rev="EBeye_Axis1">
```

```
  <artifact name="EBeye_Axis1" ext="jar"/>
```

```
</dependency>
```

```
<dependency org="WSDbfetch" name="WSDbfetch" rev="WSDbfetch">
```

```
<artifact name="WSDbfetch" ext="jar"/>
</dependency>
<dependency org="mysql" name="mysql-connector-java" rev="5.0.7"/>
<dependency org="org.apache.axis" name="axis" rev="1.4">
<artifact name="axis" ext="jar"/>
</dependency>
<dependency org="commons-cli" name="commons-cli" rev="1.2">
<artifact name="commons-cli" ext="jar"/>
</dependency>
<dependency org="commons-discovery" name="commons-discovery" rev="0.4">
<artifact name="commons-discovery" ext="jar"/>
</dependency>
<dependency org="commons-logging" name="commons-logging" rev="1.1">
<artifact name="commons-logging" ext="jar"/>
</dependency>
<dependency org="org.apache.axis" name="axis-saaj" rev="1.4">
<artifact name="axis-saaj" ext="jar"/>
</dependency>
<dependency org="org.apache.axis" name="axis-jaxrpc" rev="1.4">
<artifact name="axis-jaxrpc" ext="jar"/>
</dependency>
<dependency org="org.apache.axis" name="axis-jaxrpc" rev="1.4">
<artifact name="axis-jaxrpc" ext="jar"/>
```

```
</dependency>
```

```
<dependency org="wsdl4j" name="wsdl4j" rev="1.5.1">
```

```
<artifact name="wsdl4j" ext="jar"/>
```

```
</dependency>
```

```
</dependencies>
```