

TDWG Life Sciences Identifiers Authority Setup Guide

Using the LSID Perl Software Library

Date:

21-Dec-2007

Authors:

Ricardo Pereira (TDWG Infrastructure Project, Brazil)

Task Group:

TDWG Globally Unique Identifiers Task Group (GUID)

<http://www.tdwg.org/activities/guid/>

Abstract:

This document provides guidance on how to set up an LSID authority using the **LSID Perl software library**. Readers must first follow the instructions on the companion document titled "[TDWG LSID Authority Setup Guide – Programming Language Independent Steps](#)" which provides guidance for the part of the process is independent of the programming language used.

Status:

Accompanying (type 3) documentation for the [TDWG LSID Applicability Statement](#).



Copyright © Biodiversity Information Standards - TDWG (2007). Some Rights Reserved.
This work is licensed under a [Creative Commons Attribution United States 3.0 License](#).

To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/us/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Disclaimer:

This document and the information contained herein are provided on an "AS IS" basis. TDWG MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

Table of Contents

| | |
|--------------------------------------------------------------|---|
| Preparation – Platform Independent Steps | 4 |
| Setup Outline..... | 4 |
| 1. Install Perl | 5 |
| 2. Download and Install the LSID Perl Software Library | 5 |
| 3. Install Other Required Software | 5 |
| 4. Modify the Template Authority | 6 |
| 4.1. Setting up More Namespaces | 7 |
| 5. Set up Apache..... | 7 |
| 6. Finishing up and Testing the Authority..... | 7 |
| Appendix A - Mapping Between LSID Parts and Perl Code..... | 8 |

Introduction

This document provides guidance on how to set up an LSID authority using the **LSID Perl software library**. Readers must first follow the instructions on the companion document titled "[TDWG LSID Authority Setup Guide – Programming Language Independent Steps](#)" which provides guidance for the part of the process of setting up an LSID authority that are independent of programming language used.

The instructions in this guide refer to version 1.1.4 of the LSID Perl software library, but should apply to version 1.1.3 of the libraries as well.

Preparation – Programming Language Independent Steps

Before proceeding setting up your LSID authority using the LSID Perl software library, make sure you have decided-

1. What categories of objects you will assign LSIDs to.
2. What **Namespace Identifications** you will use for each category listed above.
3. How to form the **Object Identification** for objects in each namespace.
4. Whether to use and how to form the optional **Revision Identification** for the objects in each namespace.
5. What **data** and **metadata** will be associated with each LSID.
6. What RDF representation will be used to express LSID metadata
7. What your LSID **Authority Identification** will be.

Also make sure that you have the **DNS SRV record** corresponding to your LSID authority identification set up and fully propagated.

For more information on how to make the decisions above or to set up DNS for work with your LSID authority, please see the [TDWG LSID Authority Setup Guide – Programming Language Independent Steps](#). If you are setting up an LSID authority for biodiversity information, please follow requirements and recommendations from the [TDWG LSID Applicability Statement](#) as well.

Setup Outline

The process of setting up an LSID Authority using the LSID Perl software libraries is as follows:

1. Install Perl (if you already don't have it on your server).
2. Download and install the latest version of the LSID Perl software library.
3. Install required Perl modules.
4. Modify the template authority with your database details.
5. Set up Apache to work with your LSID authority.

1. Install Perl

Make sure Perl (version 5.6 or 5.8) is installed on the server. Use the following command to check the version of Perl you have installed:

```
perl -v
```

If the correct version is not installed, download and install Perl for your platform.

2. Download and Install the LSID Perl Software Library

Download the LSID Perl software library from-

http://sourceforge.net/project/showfiles.php?group_id=130827&package_id=144608

Uncompress the contents of the downloaded file and follow the installation instructions on the README file.

3. Install Other Required Software

Now check whether you have all required Perl modules installed. The LSID Perl 1.1.4 requires the following modules:

```
SOAP-Lite-0.60a
URI-1.21
libwww-perl-5.53
Net::FTP 2.65
Net-DNS-0.28
XML-XPath-1.12
MIME-Base64-2.12
File::Temp 0.12 (Install from CPAN, File::Temp in the Perl distribution is not enough)
RDF-Core-0.30
MIME-tools >6.100 (Needed for Apache AXIS compatibility - v5.x is fine for HTTP bindings)
```

You can check the versions of these Perl packages you have installed by running the test script from the downloaded package using the following command:

```
perl test_perl_modules.pl
```

Identify the Perl modules that are missing or too old and download them from CPAN (or using your favourite method for locating Perl modules). Install them and run the tests again until you have all dependencies met.

If you get an error such as “Can’t locate XXX.pm in @INC”, that probably means that you don’t have any version of the corresponding library installed.

4. Modify the Template Authority

The downloaded package includes a number of sample LSID authorities in the examples directory. The authority named `hugo_authority` is closest to a working generic LSID authority. We will use that one as a template.

The template authority has two main files:

1. **hugo_authority.pl** – a regular Perl script which implements the LSID authority; and
2. **hugoNamespaces.pm** – a Perl module that implements an LSID namespace.

Copy both files to the location you want your web server to serve LSIDs from.

Rename **hugoNamespaces.pl** to reflect your namespace you selected and make the following modifications¹ to it:

1. Change the namespace identification in line 31 to the one you selected.
2. Change the database connection parameters starting at line 40 to let the authority script connect to your database. Avoid using your database root account for this. Best is to use a database account with read-only (select) privileges on the required database tables.
3. Modify the code from line 69 to 165 to retrieve your database records and format the metadata response as follows:
 - a. Change the SQL statement on line to retrieve the database records you will use to form the metadata response.
 - b. Change the code from line 85 to 118 to create the metadata representation from your database records.
4. Modify package predicates to create your own RDF predicates or use existing ones from the Web or TDWG Schema Repository at <http://rs.tdwg.org/ontology/>.

Rename **hugo_authority.pl** to reflect your authority name and make the following modifications to it:

5. Change the name of the included namespaces module to the one you set in step X above.
6. Change the code that creates the authority service starting at line 53 to reflect your authority identification and name.
7. If needed, change the code that adds ports to the authority starting at line 65. Make changes only if you want to modify the port endpoints. Otherwise leave the default values.

¹ Line numbers refer to revision 1455 in Subversion source code repository:
https://lsids.svn.sourceforge.net/svnroot/lsids/trunk/lsid-perl/examples/hugo_authority.

4.1. Setting up More Namespaces

The instructions above will get you a single namespace in your authority that is defined by the Perl package named **hugo**. You can set up additional namespaces within the same LSID authority by creating new copies of the hugo package. To do so, follow these instructions:

1. Replicate the code from line 11 to 170 which corresponds to the Perl package named **hugo** somewhere in the same file for each namespace you want to set up. Make the changes described above each copy you made. Make sure you also change the name of the package (originally as line 11) for each additional copy you made, otherwise the package names will collide.
2. On the **hugo_authority.pl** file create a new copy of line 77 for each new namespace you created above. Replace the reference to hugo package to the new package names you created.

5. Set up Apache

Use the sample apache configuration in file **http.conf.EXAMPLE** to have your Apache instance to resolve LSIDs.

You can set up your LSID authority scripts to run as regular CGI scripts or you can use mod_perl. We strongly recommend the latter for production setups as our tests showed that mod_perl improved performance of our resolver more than 25 times.

6. Finishing up and Testing the Authority

Three more steps are required to finish up setting up the LSID authority:

1. Test the LSID Authority
2. Advertise the LSID Authority
3. Tag the objects in your information systems with their LSIDs.

Please see the document "[TDWG LSID Authority Setup Guide – Programming Language Independent Steps](#)" for instructions on how to perform those last steps.

Appendix A - Mapping Between LSID Parts and Perl Code

Each part of your LSIDs is mapped into the Perl code as follows:

Network Identifier (NID) – the “urn:lsid” prefix; it is fixed and unchanged for all LSIDs.

Authority Identification – defined via a parameter in method LS::Service::Authority->new (in line 55 of file hugo_authority.pl)

Namespace Identifications – created by each package in file hugoNamespaces.pm and associated to the authority via method LS::Service::DataService::addNamespace (in line 77 of file hugo_authority.pl).

Object Identification – defined by the parameter used in the lookup SQL query (in lines 63 and 71 of file hugo_authority.pl).

Revision Identification – In the hugo authority the revision is simply concatenated to the object identification (in line 47 of file hugoNamespaces.pm), but the revision could be mapped and processed separately for more complex databases.

Object Metadata – generated by hugo::getMetadata() method or equivalent for each namespace defined.