

# TDWG Life Sciences Identifiers Authority Setup Guide

## Programming Language Independent Steps

**Date:**

21-Dec-2007

**Authors:**

Ricardo Pereira (TDWG Infrastructure Project, Brazil)  
Richard Pyle (Bishop Museum / Pacific Basin Information Node - NBII, U.S.A.)  
Kevin Richards (Landcare Research, New Zealand)

**Task Group:**

TDWG Globally Unique Identifiers Task Group (GUID)  
<http://www.tdwg.org/activities/guid/>

**Abstract:**

This document describes how to set up LSID authorities for biodiversity information systems. It describes the part of the process that is independent of the programming language used to implement the authority. To complete the setup process, you must also read one of the accompanying documents describing the specific steps for setting up the LSID authority using Java, Perl or .NET programming platforms.

**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:**

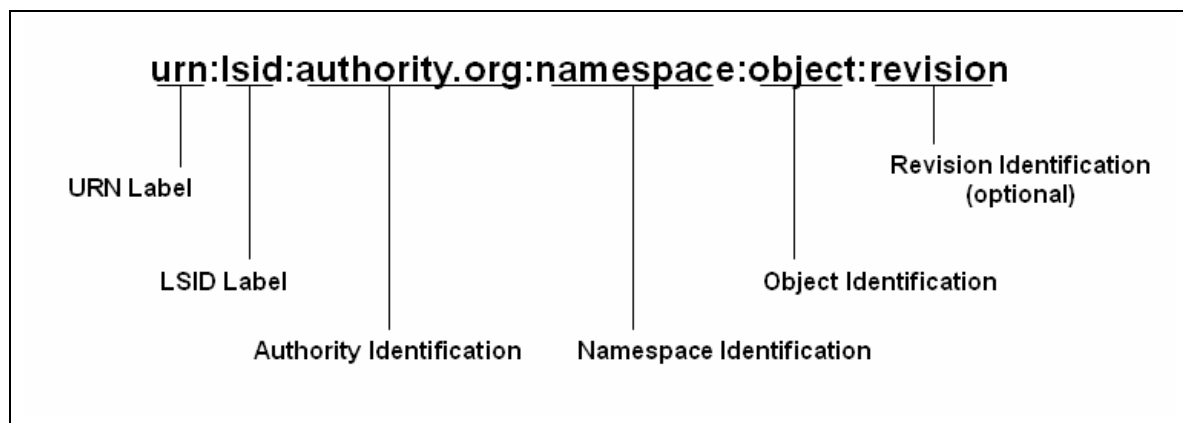
This document and the information contained herein are provided on an "AS IS" basis. TDWG MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

## Table of Contents

Introduction.....	4
Setting up an LSID Authority: an Outline.....	6
1. Identify which categories of objects to assign LSIDs to .....	7
2. Define the Authority Identification .....	8
3. Define the Namespace Identification for each object category .....	10
4. Define the Object Identification for objects in each namespace .....	11
5. Define an optional Revision Identification for the objects in each namespace .....	11
6. Define the data associated with each LSID .....	12
7. Define the metadata associated with each LSID .....	12
8. Set up the LSID Authority service .....	13
9. Test the LSID Authority .....	14
10. Tag the objects in your information systems with their LSIDs.....	15

## Introduction

Life Science Identifiers (LSIDs) are a type of Globally Unique Identifier (GUID), that have been adopted by Biodiversity Information Standards (TDWG; formerly the Taxonomic Database Working Group) and the Global Biodiversity Information Facility (GBIF) and . Additional information can be accessed at the [LSID Home page](#) but a basic summary of LSIDs follows.



**Figure 1. Structure and Syntax of an LSID.**

Every LSID consists of at least four, and up to five basic parts, each separated by a colon (see Fig. 1).

1. **Network Identifier (NID)**
2. **Authority Identification**
3. **Namespace Identification**
4. **Object Identification**
5. **Revision Identification**

The **Network Identifier (NID)** is a fixed prefix of “**urn:lsid**” for all LSIDs.

The **Authority Identification** part of the LSID is what allows LSIDs to be “self-resolving” - the information within the identifier itself provides a mechanism to find out what the identifier is assigned to. In most cases, this corresponds to the DNS-registered root domain name of the organization issuing the LSID. This document was written with the assumption that this part of the LSID is a domain name that is under the control (in terms of DNS registration) of the same organization that will be issuing the LSIDs.

The **Namespace Identification** part of the LSID is intended to allow the LSID issuer (i.e., the LSID authority) to create different sets or types of LSIDs for different domains of data. There are no established standards for what values of the Namespace Identification should be used for different kinds of datasets (e.g., specimens, images, taxon names, etc.). This part of the LSID is an intermediate level of uniqueness for the LSID itself. This permits a finer resolution than the Authority Identification, and thereby allowing for clustering of sets of Object Identifiers (next LSID part) within a single Authority Identification value.

The **Object Identification** part of the LSID usually corresponds to a locally unique identifier, such as the primary key value for a data table that forms the basis of the LSID service within the context of a given Namespace Identification value.

The **Revision Identification** part of an LSID allows for multiple revisions of a single data object, and thereby allowing unique identifiers to be assigned to each revision. Whether or not this part of

an LSID should be used (and in which circumstances) depends, on whether any “data” (*sensu* LSID specification) are returned for the LSID (vs. just metadata).

No description of LSIDs would be complete without some discussion about the distinction of “data” vs. “metadata”. The LSID specification has very specific definitions and implications concerning these two terms. According to the LSID spec, “data” are immutable; that is, a given LSID must always return the exact same (bit-identical) “data”. For example, if a TIFF image file is the data object identified by an LSID, then changing even one bit of one pixel of that image file would require that a new LSID to be assigned (let alone converting the image file to a JPEG or some other format). The metadata returned by an LSID may be changed.

LSIDs are intended primarily to identify digital objects. These objects are usually a binary computer file (such as an image file, PDF document, video file, text file, etc.). A change in the byte sequence of the file requires a new LSID because it represents a new (and different) digital object. There are many cases where digital objects that are “semantically identical”, or functionally identical from the perspective of a human, may be rendered via different bytes of data. For example, a text file encoded as 8-bit ASCII text will be different, at the level of byte sequence, from the “same” text document encoded as UTF-8 (unicode) text. A digital TIFF image may look identical to a human eye when converted to a high-quality JPEG file, but these files are different at the binary level.

To accommodate objects that differ in byte sequence but are otherwise “semantically identical”, the LSID specification describes LSIDs assigned to what are known as “abstract” or “conceptual” objects. Such LSIDs do not represent (or return) any “data”; only metadata. In the digital image example, the TIFF file and the JPEG file derived from it represent the same “conceptual” image (i.e., a recording of the same set of photos that passed through a camera lens when shutter was released), to which a “conceptual” LSID might be assigned. Among the metadata returned for this “conceptual” LSID would be attributes specific to the conceptual image such as Date, Location, Photographer and links to other LSIDs that identify specific binary renderings of the “same” conceptual image (e.g., RAW, TIFF, and JPEG files).

Finally, LSID resolver services return metadata as **Resource Description Framework (RDF) documents**. A [RDF Primer](#) is available for those who seek more details. It is also recommended that the following documents be read or scanned before implementing an LSID service

- [LSID “Best Practices”](#)
- [TDWG LSID Applicability Statement](#)
- [CoverPages.org](#) has site.
- The [LSID Specification](#)

A comprehensive range of documents and resources are available through the [TDWG LSID page](#), the [LSID Sourceforge site](#) and a particularly useful article, with many links to other resources, is available on the [CoverPages.org](#) site.

## Step in Setting up an LSID Authority

The process of setting up an LSID Authority is as follows:

1. Identify which categories of objects to assign LSIDs to.
2. Define the **Authority Identification**.
3. Define the **Namespace Identification** for each category identified above.
4. Define the **Object Identification** for objects in each namespace defined above.
5. Define an optional **Revision Identification** for the objects in each namespace.
6. Define the **data** associated with each LSID.
7. Define the **metadata** associated with each LSID.
8. Set up the LSID Authority service.
9. Test the LSID Authority
10. Tag the objects in your information systems with their LSIDs.

## 1. Identify which categories of objects to assign LSIDs to

Data providers would normally assign LSIDs to data objects they are authorities for. Examples of objects in the biodiversity information domain that **should** be assigned LSIDs are-

- scientific names,
- taxonomic concepts,
- species observations,
- specimens,
- collections,
- images of type specimens,
- images, videos, and sound recordings of specimens

Providers are also encouraged to assign LSIDs to other kinds of objects they share with their clients.

**Aggregators** are a special kind of data providers. They add value to existing data objects by collecting and integrating data from distributed, heterogeneous sources. Added value may come from:

- Integrating objects into homogeneous datasets;
- Verifying consistency;
- Georeferencing locality descriptions;
- Checking spelling or
- Resolving ambiguities

Aggregators serve the value-added objects to clients and become the authority for these additions to the original objects. Aggregators **should** assign new LSIDs to derived objects because the process and the additions need to be described by metadata.

Aggregators must use the object original LSID if no modifications are made to that object, i.e., when they act as data indexes or caches.

## 2. Define the Authority Identification

LSIDs provide three “scoping mechanisms” or levels of resolution for identifying objects:

- the **Authority Identification** part,
- the **Namespace Identification** part, and
- the **Object Identification** part.

These layers parallel the three-part identifiers used in the DarwinCore Federation Schema: InstitutionCode, CollectionCode, CatalogNumber. The Authority Identification identifies the LSID issuing authority. The Namespace Identification identifies a cohesive set of objects (like a collection). The Object Identification identifies an object within the Namespace scope.

There has been considerable discussion about the “authority” part of an LSID (see the [TDWG LSID web page](#)). The simplest and most reliable way to select the Authority Identifier part is to use a domain name for which you or your IT systems administrator can access the DNS record. The LSID Authority itself does **NOT** need to reside at the domain name used for the Authority Identification part of the LSID. The easiest way to redirect LSID resolution requests to a different domain name/server is to add an SRV record to the domain name used for the Authority Identification part of the LSID. Instructions for creating the SRV record are located near the bottom of [this page](#).

If the Authority Identifier for the LSIDs is a root domain name (e.g., “mymuseum.org”), and the service is located on port 8080 at mymuseum.org/authority/, then the SRV record should have the following values (using BIND syntax):

```
_lsid._tcp IN SRV 1 0 8080 mymuseum.org.
```

In this case, the service is “\_lsid”, the protocol is “\_tcp”, the priority is “1”, the weight is “0”, the port number through which the service is accessed is “8080”, and the domain name target where the service is located is “mymuseum.org.” Note the trailing period.

If the LSID authority service is located at a subdomain of the domain indicated in the Authority Identifier of the LSID (e.g., at lsidauthority.mymuseum.org/authority/), or at a different domain name/server altogether (e.g., at anothermuseum.org/authority/), then the target in the SRV record is changed accordingly:

```
_lsid._tcp IN SRV 1 0 8080 lsidauthority.mymuseum.org.
```

or

```
_lsid._tcp IN SRV 1 0 8080 anothermuseum.org.
```

If the Authority Identifier for the LSID itself is a subdomain name as in “urn:lsid:lsidauthority.mymuseum.org:[etc.]”, the SRV record would need to include a “name” component that would look like this:

```
_lsid._tcp.lsidauthority IN SRV 1 0 8080 lsidauthority.mymuseum.org.
```

Note that the “target” of the SRV record (where the LSID authority is actually located on the internet) is independent of the “name” part of the SRV record (i.e., whether a subdomain is included as part of the Authority Identifier of the LSID itself), in terms of setting up the SRV record. For example:

```
_lsid._tcp.lsid IN SRV 1 0 8080 lsidauthority.mymuseum.org.
```

would be used for the SRV record of the DNS for the domain “mymuseum.org” if the LSIDs were of the form “urn:lsid:lsid.mymuseum.org:[etc.]”, and the authority service for these LSIDs was located at [lsidauthority.mymuseum.org/authority](http://lsidauthority.mymuseum.org/authority).

It is best to use the root domain name as the LSID Authority Identifier. If separate LSID authorities and resolver services are required for an organization, different subdomain prefixes to the Authority Identifier part of the LSIDs will be needed to resolve the different services.

Where one organization (e.g., “host.org”) hosts the LSID authority service on behalf another provider organisation (e.g., “provider.org”), the [TDWG LSID Applicability Statement](#) recommends *against* using an Authority Identifier for the LSIDs of the form “provider.host.org” This approach would permanently tie the LSIDs to host.org. Instead, “provider.org” or a subdomain should be used for the LSID Authority Identifier with direct LSID calls to the target host.org via the SRV record of provider.org as described above.

If the reason for an alternate host was due to lack of administrative access to (or uncertain persistence of) the DNS record for “provider.org”, then use of the central and independent TDWG [LSID Authority Identification](#) service should be considered. This service allows organizations to issue LSIDs with the Authority Identification part of the LSID formatted as “<authority\_name>.lsid.tdwg.org”, where “<authority\_name>” is a unique string assigned by TDWG on behalf of the LSID-issuing organization. TDWG will also provide the related DNS SRV resource record to be used to locate the LSID resolver. To use this service, [Register](#) on the TDWG site and log on to the TDWG Wiki..

Long-term persistence of the self-resolution capabilities of LSIDs needs to be considered when selecting the Authority Identifier of LSIDs. Recommendations are included in section 3 of the [TDWG LSID Applicability Statement](#).

### 3. Define the Namespace Identification for each object category

There is no current consensus about selecting Namespace Identifiers (see Figure 1) for LSIDs. The main purpose of the Namespace Identifier is to enable the LSID issuer to serve LSIDs for more than one “domain” or category of data. Categories include object type, scientific or taxonomic discipline, departments, collections or projects. Different Namespace Identifiers allows for unique LSIDs in cases where the different domains have potentially overlapping Object Identifier values.

Suppose an organization wished to serve LSIDs for both specimens and publications. The organization wants to use its internal identifiers for the Object Identification part of the LSID, but these values may overlap between the specimen and publication datasets (e.g., there was a specimen with ID#12345 in the specimens table, and a publication that also had ID#12345 in the publications table). Using different Namespace Identifiers would address this problem by using:

urn:lsid:myorganization.org:specimen:12345

and

urn:lsid:myorganization.org:publication:12345

The [TDWG LSID Applicability Statement](#) recommends against using the Primary Key (PK) field of a relational database because there are situations where PK values may change. For example, when merging databases. A separate dedicated field should be established to store the Object Identifier value.

Don't read too much into the Namespace Identifier. LSIDs are intended to be “semantically opaque” so the value of the Namespace Identifier should not be relied upon for adding intelligible information about the identified object. For example, it is possible that an LSID system initially established for specimens may eventually be expanded to include observations (or vice versa)..

#### **4. Define the Object Identification for objects in each namespace**

The Object Identifier should be thought of as the stand-alone unique identifier within the context of the Authority and Namespace scope. Although it is tempting to use the Primary Key value of a relational database table for the Object Identifier, this practice is discouraged as described above. The Object Identifier can be alphanumeric text.

#### **5. Define an optional Revision Identification for the objects in each namespace**

According to the [TDWG LSID Applicability Statement](#), LSID Authorities should use the revision identifier to manage object that are revised over time.

The revision identification allows independent management of object revisions. While the object identifier is used to uniquely identify an object within a namespace, the revision identifier is used to distinguish between revisions of an object over time.

Refer to section 6 of the [TDWG LSID Applicability Statement](#) for more information on the LSID revision identification and versioning.

## 6. Define the data associated with each LSID

When setting up an LSID authority, data providers may assign data, i.e., a sequence of bytes to their LSIDs. These data may be a DNS sequence in FASTA format, or an Ecological Markup Language (EML) instance document.

The process of assigning data to LSIDs depends on the LSID software library and the platform being used. Refer to section 8 for more information about associating data with LSIDs for each specific LSID software library.

According to the LSID specification, data associated with an LSID (i.e., the content returned by the `getData` method) must never change. This is a requirement for the `getDataByRange` method to work as expected. That method has two parameters that define the starting point and the length of the subset of the data to be returned. If the data associated with the object changes, subsequent calls to `getDataByRange` may yield different results. This is not permitted in the specification.

Additionally, LSID authorities in the biodiversity information domain should-

- avoid encoding data in formats such as XML that may change the exact sequence of bytes;
- avoid return irrelevant data in LSID `getData` calls; and
- avoid using `getData` just to assert that some attributes of objects are immutable.

Refer to the section 8 of the [TDWG LSID Applicability Statement](#) for further details.

## 7. Define the metadata associated with each LSID

Metadata may change over time. The only requirements for LSID metadata for objects from the biodiversity information domain are that-

- the default metadata response format is RDF serialized as XML,
- HTTP GET is the default binding for LSID `getMetadata` calls, and
- objects are typed using the TDWG ontology or other well-known vocabularies in accordance with the TDWG common architecture.

Refer to the section 9 of the [TDWG LSID Applicability Statement](#) for more details.

## 8. Set up the LSID Authority service

Once you have made the decisions described in the previous sections, an appropriate LSID software library must be chosen. At the time of writing the LSID software were available for Java, Perl and .NET platforms. Specific details about each of these software libraries are available in the following companion documents:

- [LSID Authority Setup Guide using the LSID Java Software Library](#)
- [LSID Authority Setup Guide using the LSID Perl Software Library](#)
- [LSID Authority Setup Guide for the .NET Platform](#)

## 9. Test the LSID Authority

A very useful tool for testing LSID authorities is Rod Page's LSID Tester:

<http://linnaeus.zoology.gla.ac.uk/~rpage/lcid/>

The tool takes an LSID as input and displays the outcomes of each step of the LSID resolution protocol including useful diagnostics information such as-

- LSID validity
- the server pointed at by the DNS SRV record,
- WSDL documents for all endpoints,
- HTTP headers sent by the authority,
- warnings about non-compliance and error handling,
- RDF metadata for the LSID (serialized as XML and as a graph)

The LSID Tester only supports LSID resolution through HTTP. Resolution through SOAP and other protocols are not supported.

To test your resolver, visit the web page above, type in some of the LSIDs from your authority, and check the diagnostic messages output by the tester.

Below are sample LSIDs (valid and invalid) resolved through Rod Page's LSID Tester:

[urn:lsid:col2005.gbif.org:col2005:tc-\(t.1246388\)\(d.\)\(cn.\)\(r.\)](urn:lsid:col2005.gbif.org:col2005:tc-(t.1246388)(d.)(cn.)(r.))

<urn:lsid:ubio.org:namebank:11815>

<urn:lsid:lsid.zoology.gla.ac.uk:predicates:isBasionymOf>

<urn:lsid:tdwg.gbif.org:dharma:10298322400>

## **10. Tag the objects in your information systems with their LSIDs**

Once your LSID authority is set up you should tag the objects with their respective LSIDs wherever they appear in your information systems.

Tagging your objects with LSIDs will let you and your clients attain the benefits -

- Clients (authors or other information systems) may refer to the object unambiguously;
- The LSID alone gives users access to provenance and attribution information;

Refer to chapter 10 of the [TDWG LSID Applicability Statement](#) for more details about how to present LSIDs depending on the media they appear in.