

TDWG Life Sciences Identifiers Authority Setup Guide

Using the LSID .NET Software Library

Date:

20 December 2007

Author:

Kevin Richards (Landcare Research, New Zealand)
Richard Pyle (Bishop Museum / Pacific Basin Information Node (NBII), Hawaii, U.S.A.)
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 .NET software library**. Readers must first follow the instructions on the companion document titled "[TDWG LSID Authority Setup Guide – Programming Language Independent Steps](#)" for guidance on the part of the process that 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

Introduction.....	4
Preparation – Platform Independent Steps	5
LSID Setup	6
Download Software.....	6
IIS Configuration for the LSID Authority.....	8
IIS Configuration for the LSID Authority.....	19
Configuring the LSID Implementation.....	20

Introduction

This document provides guidance on how to set up an LSID authority using the **LSID .NET 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 is independent of programming language used.

For anyone wishing to use this guide to implement these services, the following minimum assumptions are made:

- A version of Microsoft Server is installed on a server machine connected to the internet, on which the person installing these services has Administrative privileges.
- A registered domain name is available for use as the Authority Identification part of the LSIDs, and an SRV record can be added to the DNS directing LSID calls through that domain to the server with the LSID services running
- Microsoft Internet Information Services version 5.0 or greater is installed on the server.
- Microsoft .NET framework 1.1 is installed on the server.

Both of the LSID ([LSID-DotNET](#)) and TAPIR ([TapirDotNET](#)) services were developed to operate using version 1.1 of the Microsoft® .NET Framework. Having version 2.0 or later of the .NET Framework installed on the server is fine, but version 1.1 must also be installed. Information on checking to see if the .NET Framework 1.1 is installed on a server, and how to install it if it is not already installed, is available at the [MSDN website](#).

Preparation – Platform Independent Steps

Before 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.

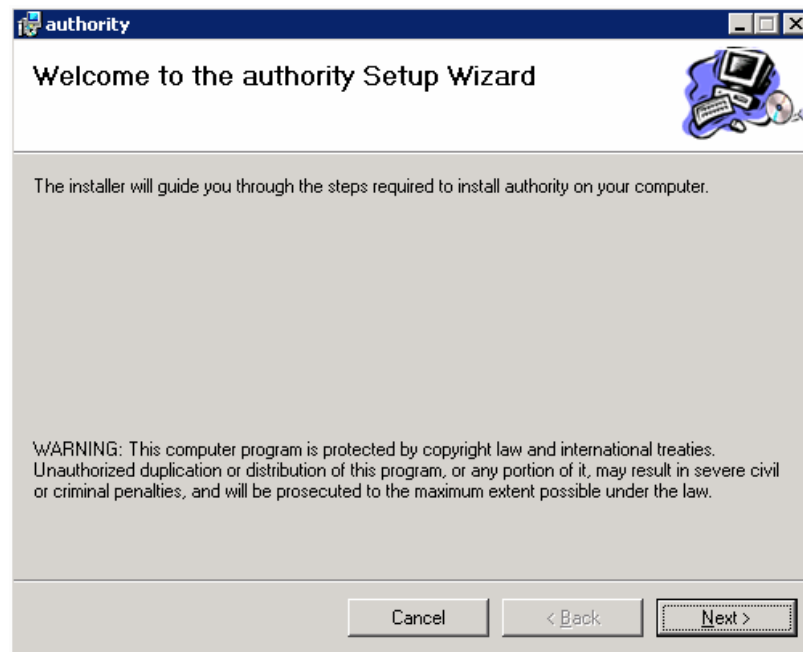
For more information on how to make the decisions above or to set up DNS for work with your LSID authority, please see the [TDWG LSID Authority Setup Guide – Programming Language Independent Steps](#). If you are setting up an LSID authority for biodiversity information, please follow requirements and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

LSID Setup

Download Software

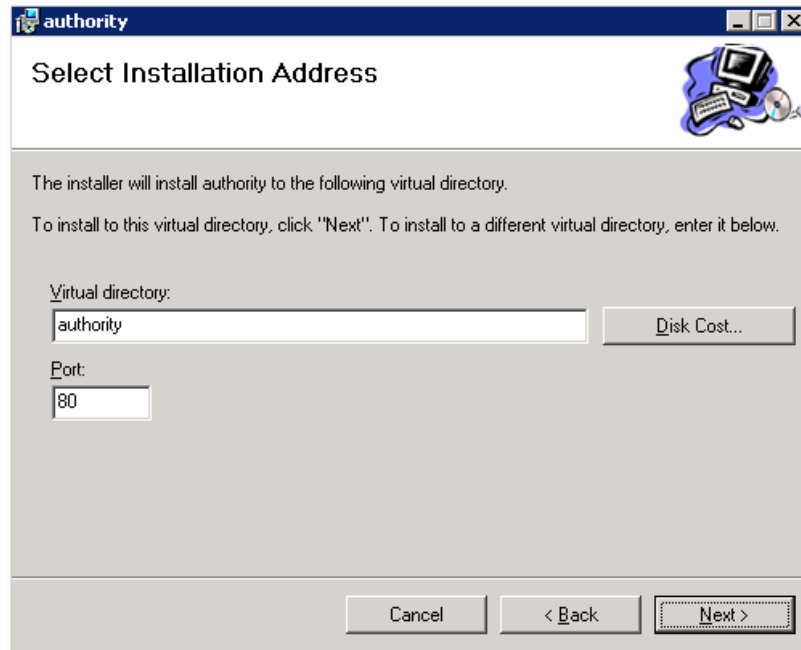
The first step in setting up an LSID service is to download the LSID-DotNET software package (<http://sourceforge.net/projects/lsid-dotnet/>) to the hard drive of the server on which the service will be installed. As of this writing, the current version of the LSID-DotNET package is v1.0.1, dated October 5, 2006.

The downloaded file of this version is named “LSID.NET.1.0.1.zip”, and unzips to five folders (/docs, /lib, /samples, /setup, and /source). In the /setup folder is a file called “LSIDAuthoritySetup.msi”. Execute this file to begin installation of the LSID Authority

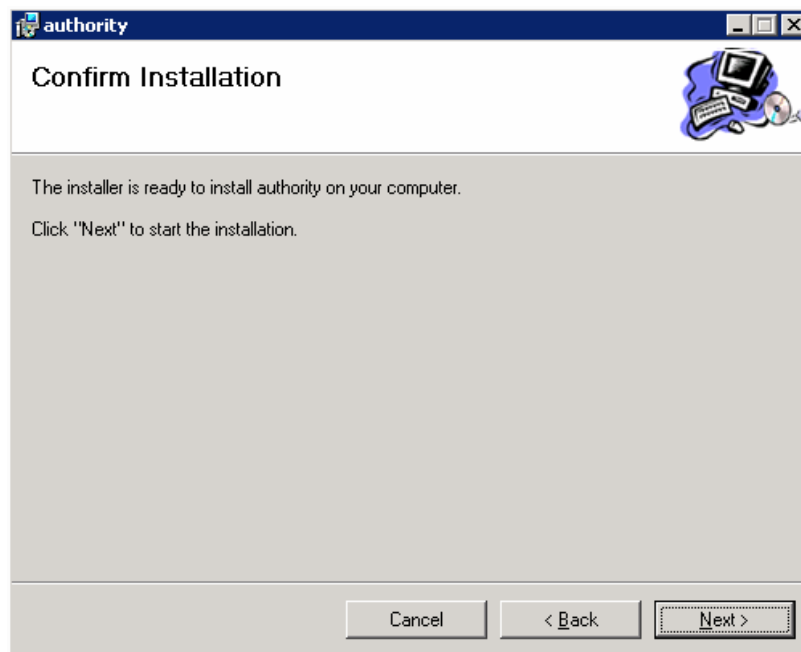


service, under IIS.

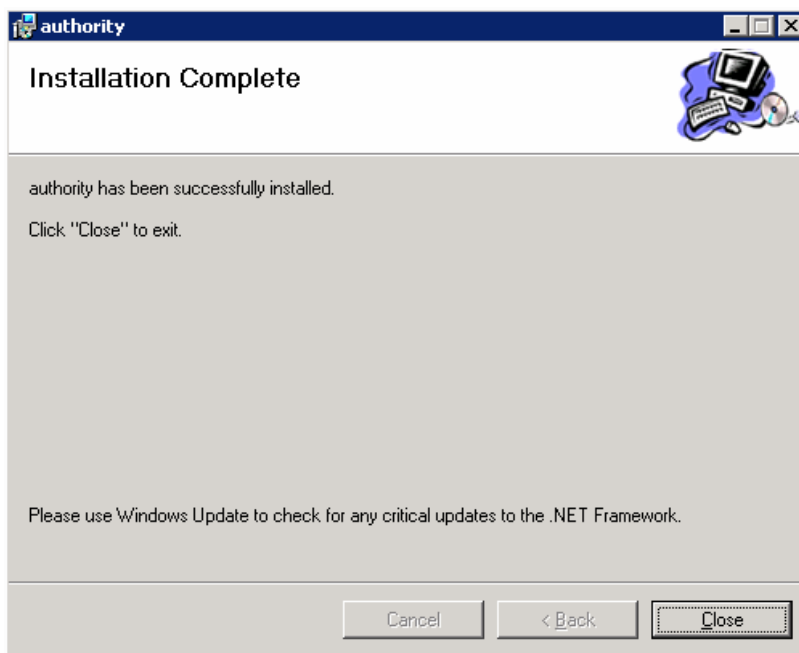
The first page of the installer program is just a welcome screen. Click “Next >” to continue.



The second page establishes the name and port number of the virtual directory to be created within IIS for the LSID Authority services. The default values are standard, and should be used for normal LSID service installations (as they were for the ZooBank LSID installation) . Click “Next >” to continue.



The third page of the installation is simply a confirmation. Click “Next >” to continue.

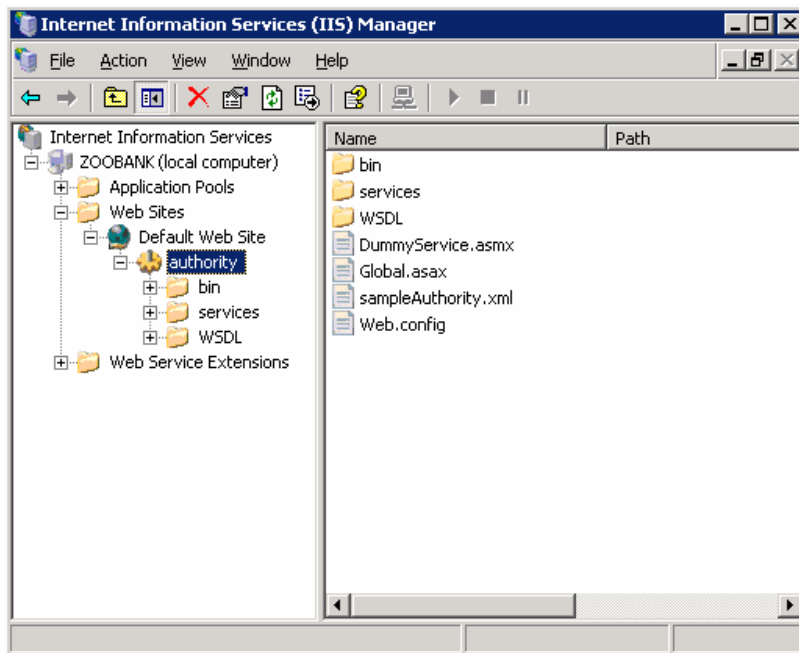


After a brief moment configuring IIS, the final page is displayed confirming the completion of the LSID Authority installation and configuration. Click “Close” to close this dialog box. Note that the LSIDAuthoritySetup.msi program can also be used to remove or repair an existing LSID Authority installation.

IIS Configuration for the LSID Authority

Note: The following description and figures apply to IIS 6, which was the version of IIS used for the installation of ZooBank. These instructions may differ somewhat if the LSID service is installed using IIS 5.

By default, the LSIDAuthoritySetup.msi program establishes a virtual directory and application called “authority” under IIS. This virtual directory will map to a real directory of the same name (“authority”) in the root web directory of the Default Web Site on the server (usually C:\inetpub\wwwroot), as in the illustration below.



It is also possible to setup the LSID authority manually; for example, if you prefer to store the files in a different location outside of the default /Inetpub/wwwroot directory (as was done for the ZooBank LSID service). The steps for manually creating the LSID authority under IIS are as follows:

- 1) Create the directory on the hard disk where the LSID Authority will be located (e.g., "W:\LSIDAuthorityService\WebAuthority"), and create three subdirectories: "bin", "services", and "WSDL".
- 2)
- 3) In the "/source" directory created by unzipping the downloaded "LSID.NET.1.0.1.zip" file, open the subdirectory "AuthorityWebService", and copy the following files from this directory to the corresponding directory or subdirectory created in step 1 above:

```
..\DummyService.asmx
..\Global.asax
..\sampleAuthority.xml
..\Web.config
```

```
..\bin\LSIDFramework.dll
```

```
..\services\default-services.xml
```

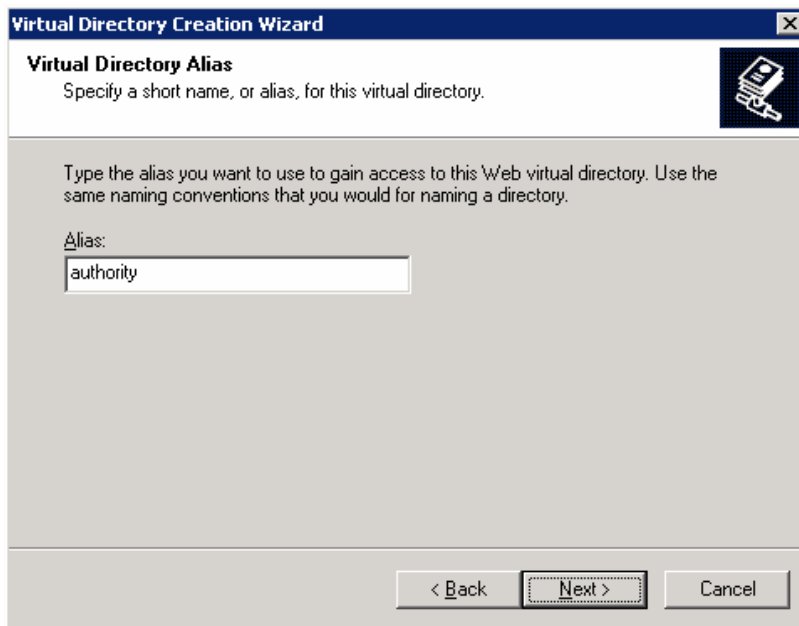
```
..\WSDL\LSIDAuthorityServiceHTTPBindings.wsdl
..\WSDL\LSIDAuthorityServiceSOAPBindings.wsdl
..\WSDL\LSIDDataServiceFileBindings.wsdl
```

..\WSDL\LSIDDataServiceFTPBindings.wsdl
..\WSDL\LSIDDataServiceHTTPBindings.wsdl
..\WSDL\LSIDDataServiceSOAPBindings.wsdl
..\WSDL\LSIDPortTypes.wsdl

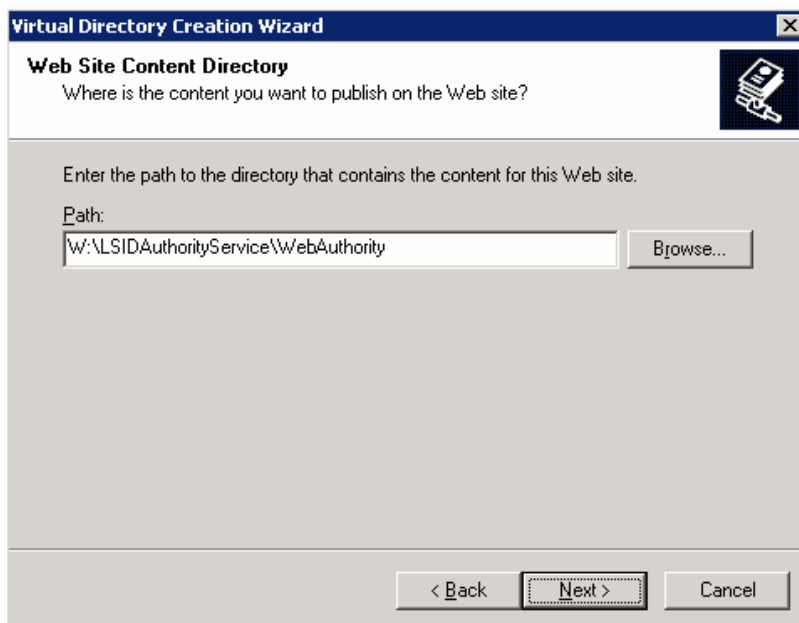
- 4) In the “/lib” directory created by unzipping the downloaded “LSID.NET.1.0.1.zip” file, copy the following additional files from this directory to the “../bin” subdirectory created in step 1 above:

..\bin\AuthorityWebService.dll
..\bin\DNSResolver.dll
..\bin\LSIDClient.dll

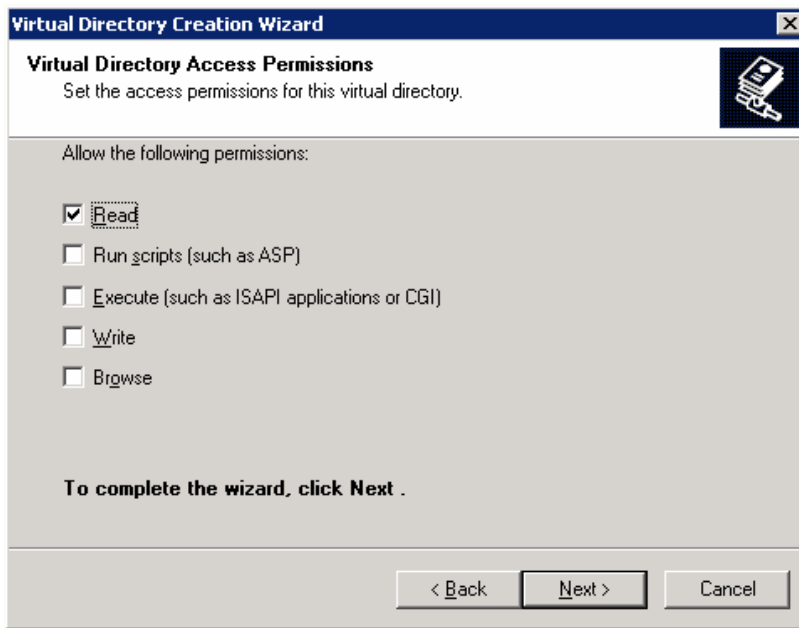
- 5) Launch IIS Manager [Start Menu → Settings → Control Panel → Administrative Tools → Internet Information Services (IIS) Manager]
- 6) In the tree on the left side of the IIS Manager window, expand the contents of the computer by clicking the “+” symbol next to the computer icon [e.g., “ZOOBANK (local computer)” in the illustration above]
- 7) Within the local computer contents, expand the “Web Sites” folder.
- 8) Right-mouse click on the line for “Default Web Site”, and from the pop-up menu select “New”, and then from the pop-up submenu select “Virtual Directory...”
- 9) Select “Next” on the initial “Welcome” dialog, then enter “authority” in the Alias text box on the following dialog page.



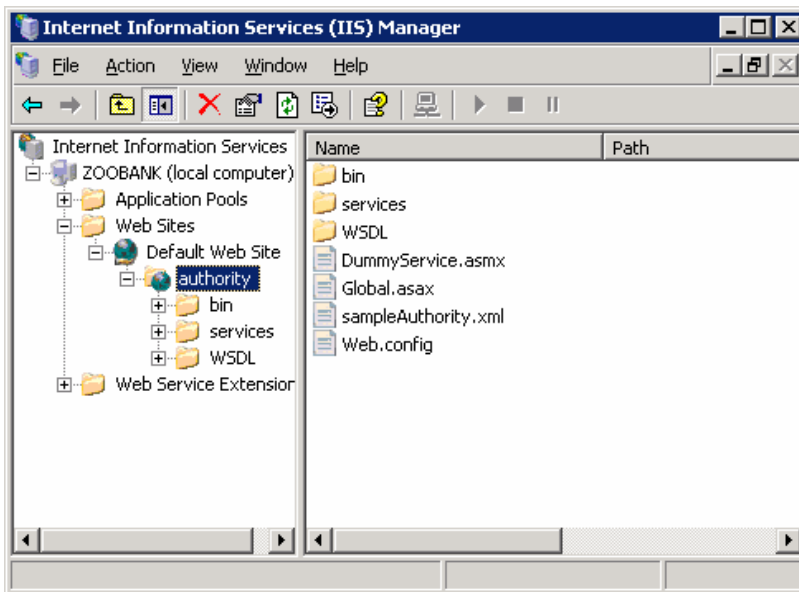
- 10) Click "Next >" and then on the next dialog page enter or browse for the full directory path where the LSID Authority service will be located.



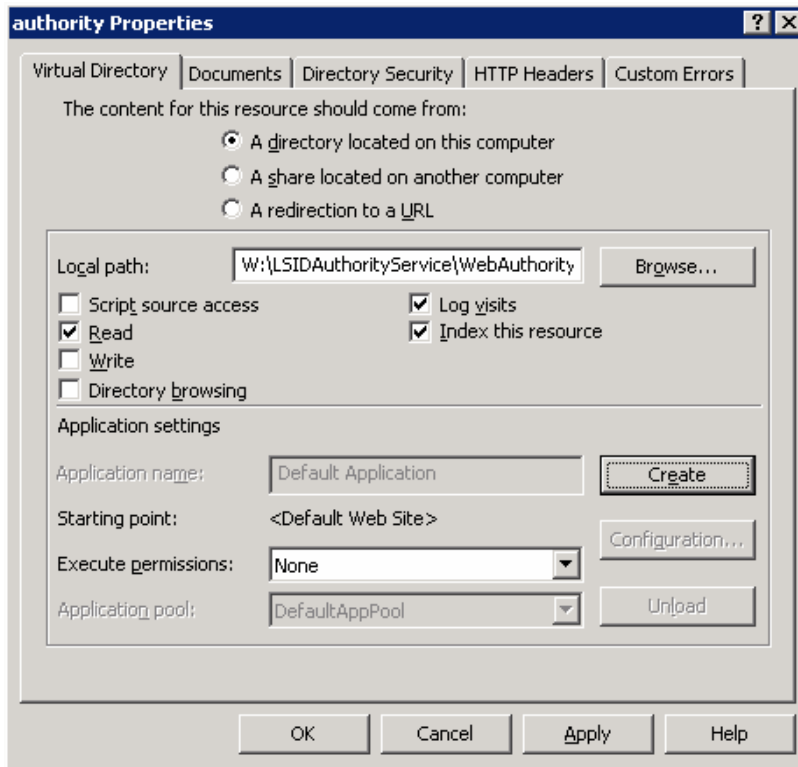
- 11) Click "Next >" and on the next dialog page make sure only "Read" is checked, then click "Next >" again.



12) On the final dialog page, click “Finish”, and you should see something like the following in IIS:



13) Next, right-mouse click on the “authority” virtual directory, and select “Properties” from the pop-up menu.



- 14) On the “Virtual Directory” tab of the Properties dialog box, click the “Create” button in the “Application settings” section (lower half) and leave the default name (“authority”) as it is.

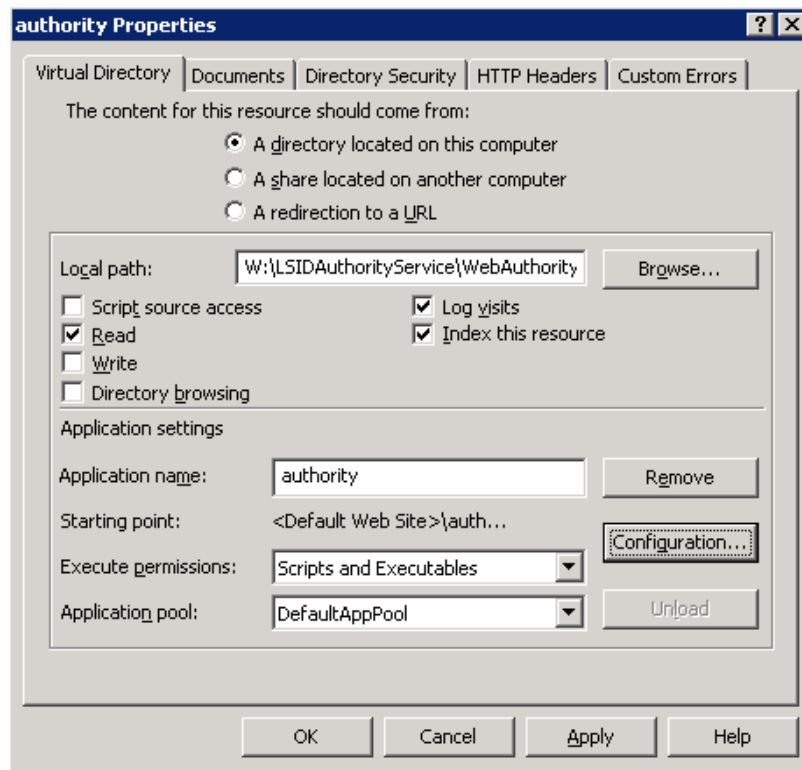
At this point, the manually-created Authority virtual directory and application should be the same as one generated automatically generated by the LSIDAuthoritySetup.msi program described earlier; with two exceptions:

- The automatically generated version will create the Authority folder within the default root directory of the IIS Default Web Site, whereas the manually generated one may be in a different directory as specified in step 1 above; and
- In the “Directory” tab of the IIS Properties dialog for the automatically-generated version, the “Execute permissions” option will be set to “Scripts only”, whereas on the corresponding “Virtual Directory” tab of the manually-created version, this option will be set to “None”.

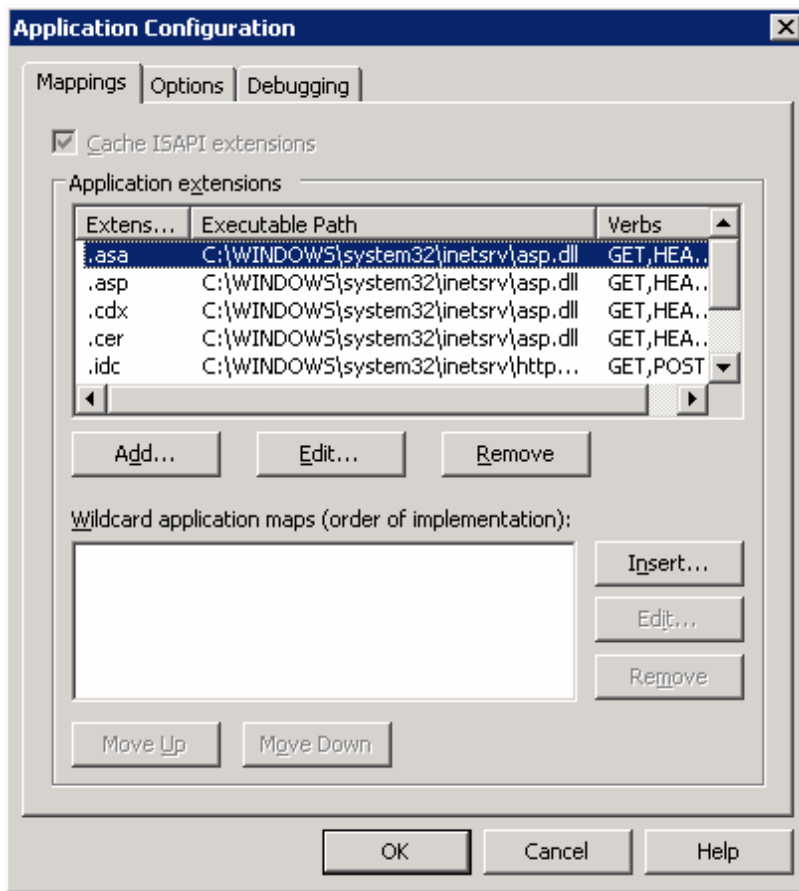
From this point forward, the directions apply regardless of whether the LSID Authority service was created automatically using the LSIDAuthoritySetup.msi program, or manually by the instructions above.

- 15) Open the IIS Properties dialog box for the authority directory/virtual directory by right-mouse clicking on the “authority” directory/virtual directory in IIS, and selecting “P_roperties” from the pop-up menu .

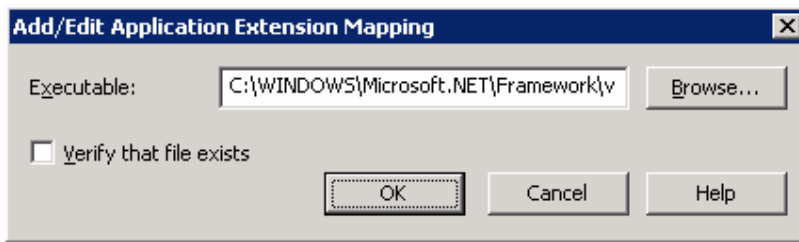
- 16) On the “Directory” or “Virtual Directory” tab of the IIS Properties dialog, select “Scripts and Executables” for the “Execute permissions” option drop-down list.



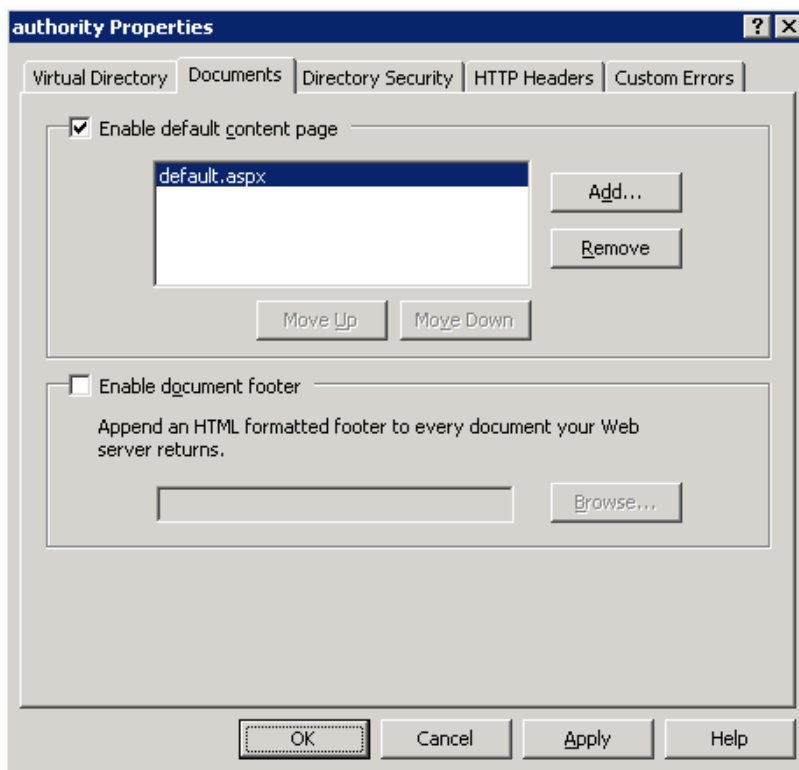
- 17) The LSID service involves calls to files without extensions, and calls to those files need to be passed to the .NET Framework. To ensure that all HTTP requests reach the server, a wildcard mapping must be added to the web directory. In order to do this, click on the “Configuration...” button on the authority Properties Dialog to open the Application Configuration Dialog box.



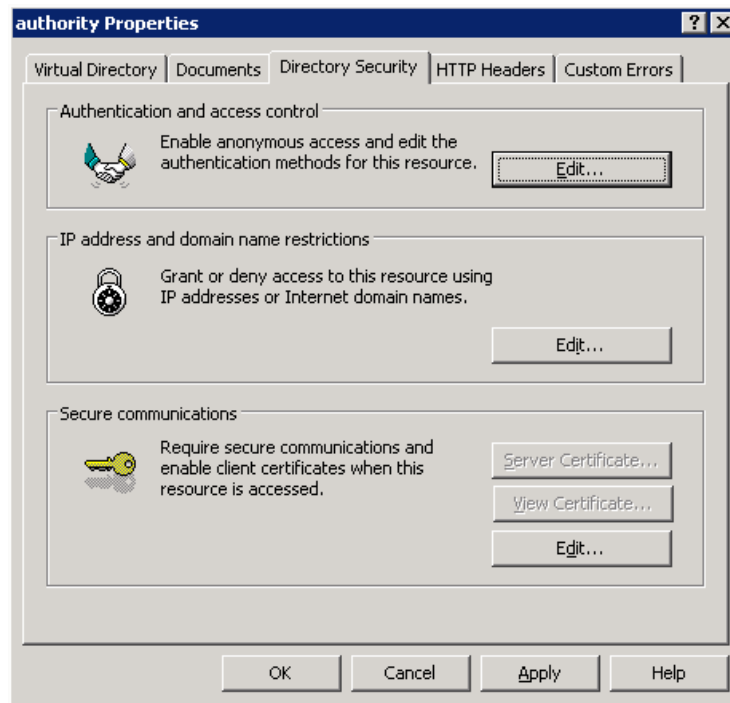
- 18) On the "Mappings" tab, in the lower half of the dialog box in the section for "Wildcard application maps", click on the "Insert..." button, and the "Add/Edit Application Extension Mapping" dialog box will appear.
- 19) Next to the "Executable" text box, click the "Browse" button and navigate to the path where the aspnet_isapi.dll file is located (should be something like "C:\Windows\Microsoft.NET\Framework\v1.1.4322\aspnet_isapi.dll").
- 20) In the "Extension" text box, enter ".*" (without the quotes), Make sure the "All verbs" and "Script engine" options are both selected, and un-check (deselect) the box for "Verify that file exists".



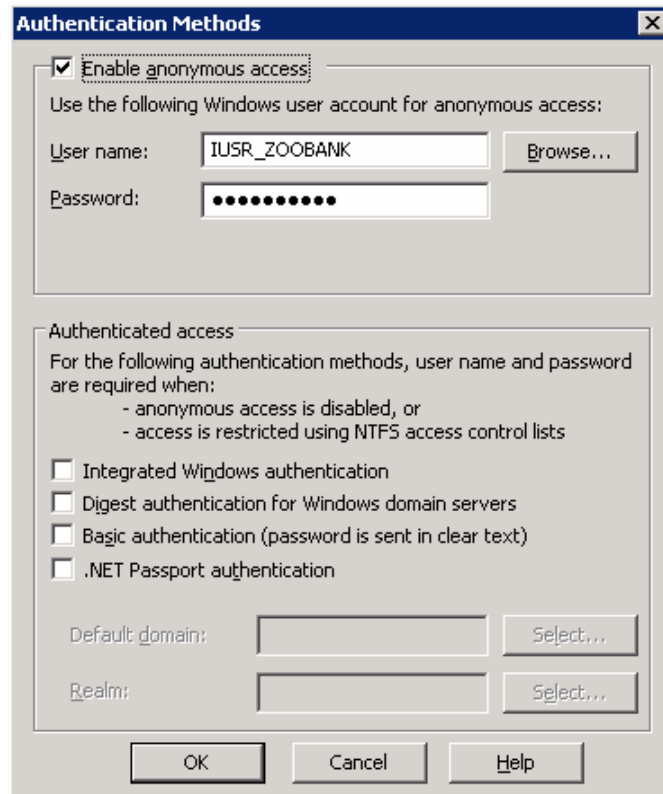
- 21) Click the “OK” button, and also click the “OK” button on the Application Configuration dialog box.
- 22) Back on the IIS Properties Dialog for the authority application, select the “Documents” tab.
- 23) Make sure that “Enable default content page” is selected, and if there are any file names other than “default.aspx” listed in the box, select them one at a time and click the “Remove” button to remove them. If “default.aspx” is not already listed in the box, then click the “Add...” button and enter “default.aspx” in the “Default content page” text box and click the “OK” button.



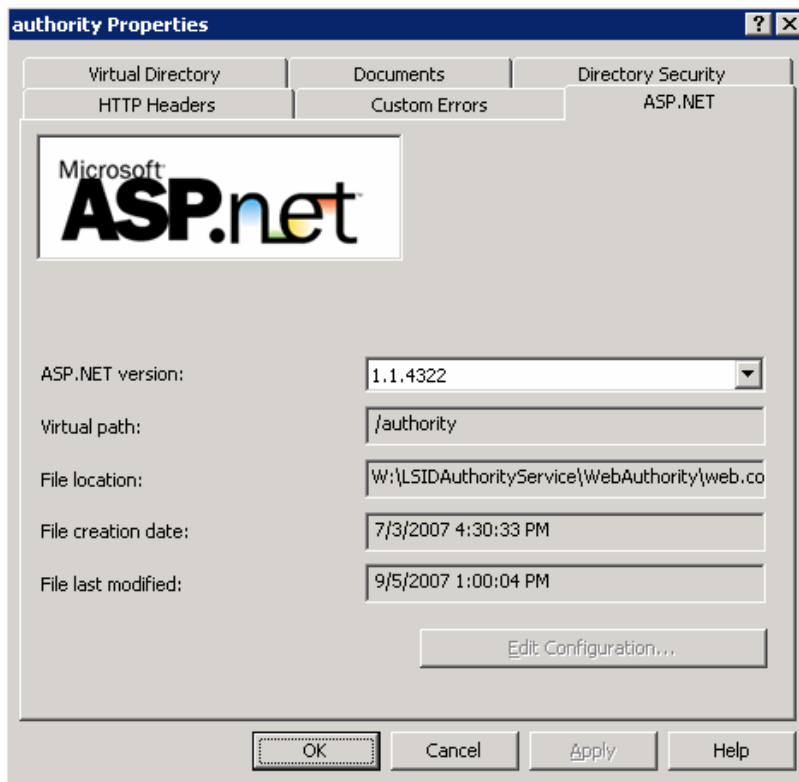
- 24) Back on the IIS Properties Dialog for the authority application, select the “Directory Security” tab.



25) In the “Authentication and access control” section, click the “Edit...” button.



26) Make sure that **only** the “Enable anonymous access” option is checked, and that no other checkbox on the dialog is selected, then click the “OK” button.



- 27) If more than one version of IIS is installed on the server, there will be an additional tab on the IIS Properties dialog box for “ASP.NET”. For the LSID authority, select this tab and make sure that the “ASP.NET version” option is selected to 1.1.4322, as this is the version of the .NET Framework that the LSID-DotNET code is written for.
- 28) The IIS Settings for the LSID Authority are now complete. Click the “OK” button to close the IIS Properties Dialog. The server is now ready to handle LSID requests.

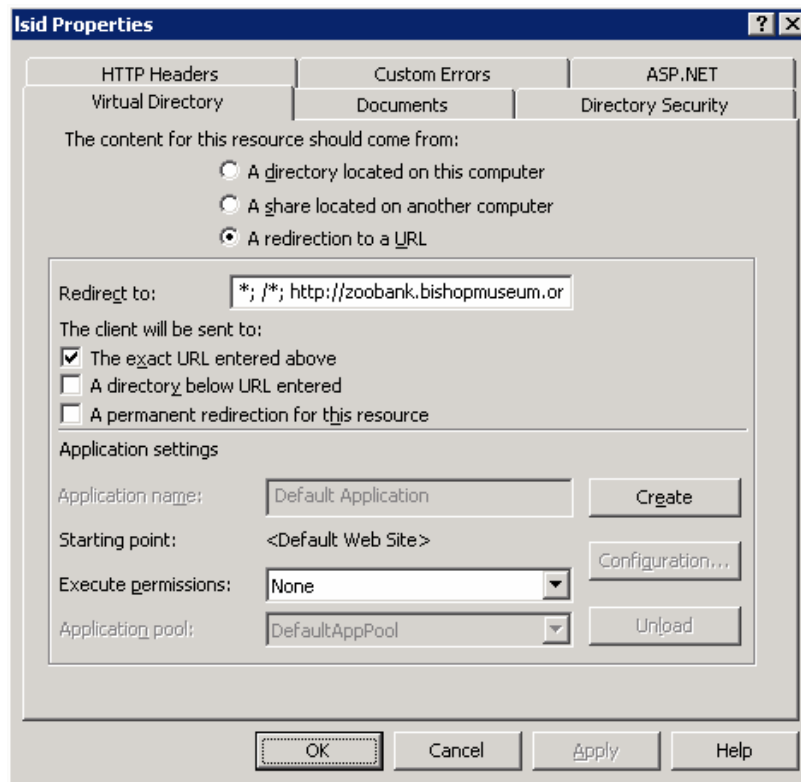
IIS Configuration for the LSID Authority

The Semantic Web does not know how to deal with LSIDs inherently. One solution is to establish a proxy for the LSID as the LSID “wrapped” within an HTTP string. This was done for the ZooBank LSID service in the following way:

- 1) Within IIS, a New Virtual Directory was created within the Default Web Site with the Alias “lsid” to a “dummy” directory (i.e., any existing directory on the server – this will be changed in the next step).
- 2) The IIS Properties dialog was opened for the new “lsid” virtual directory, and the content was changed to “A redirection to a URL”, and following text string was entered into the “Redirect to” text box:

; /; http://zoobank.bishopmuseum.org/authority/metadata/?lsid=\$0

- 3) Under the section “The client will be sent to”, the checkbox for “The exact URL entered above” was selected.



- 4) Click the “OK” button to exit.

Configuring the LSID Implementation

LSID resolution services are configured in an application server using XML documents that reside in the services/ directory inside the authority web application. These 'deployment descriptors' are typically named x-services.xml where x is a descriptive name. Below is the deployment descriptor for the Zoo Bank services.

The map is a collection of patterns that specify sets of LSIDs. Services are collections of components. Each component is an authority service, a data services or a metadata service. Each component has a single map which determines which LSIDs it should handle. It also has a type that specifies how the implementation of the service interface should be provided. The most basic type is "class" and the element body should be the class that implements the interface (LSIDAuthorityService for auth element).

ZooBank classes were written in VB.NET (using the libraries from Source Forge) to handle LSID metadata (meta xml tag in ZooBankNames), assigning (assn tag in ZooBankAssigning). References were also implemented as shown below.

```
<deployment-descriptor xmlns="http://www.ibm.com/LSID/Standard/rsdl">
  <maps>
    <map name="NamesMap">
      <pattern auth="lsid.zoobank.org" ns="taxonname" />
    </map>
    <map name="RefMap">
      <pattern auth="lsid.zoobank.org" ns="reference" />
    </map>
  </maps>
  <services>
    <service name="ZooBankAssigning">
      <components>
        <authentication type="class"
assemblyname="ZooBankResolver.dll">ZooBankResolver.ZooBankLSIDAssigner</authentication>
        <assn type="class"
assemblyname="ZooBankResolver.dll">ZooBankResolver.ZooBankLSIDAssigner</assn>
      </components>
    </service>

    <service name="ZooBankNames">
      <components>

        <auth map="NamesMap" type="class"
assemblyname="ZooBankResolver.dll">ZooBankResolver.ZooBankNamesAuthority</auth>
        <meta map="NamesMap" type="class"
assemblyname="ZooBankResolver.dll">ZooBankResolver.ZooBankNamesAuthority</meta>

      </components>
    </service>

    <service name="ZooBankReferences">
      <components>

        <auth map="RefMap" type="class"
assemblyname="ZooBankResolver.dll">ZooBankResolver.ZooBankReferenceAuthority</auth>
        <meta map="RefMap" type="class"
assemblyname="ZooBankResolver.dll">ZooBankResolver.ZooBankReferenceAuthority</meta>
        <data map="RefMap" type="class"
assemblyname="ZooBankResolver.dll">ZooBankResolver.ZooBankReferenceAuthority</data>

      </components>
    </service>

  </services>
</deployment-descriptor>
```

The VB.NET code implements the `SimpleResolutionService` LSID.NET class and obtains the name metadata from the ZooBank SQL Server database, converts this to TCS RDF and returns this RDF to the requesting client. This is all done by overriding the `SimpleResolutionService` class and implementing several methods of the class.

The conversion of the data from SQL Server into RDF was achieved using .NET libraries written by Kevin Richards, which produce TaxonName RDF, TaxonConcept RDF, and RDF for other related classes. The RDF document handling uses the SemWeb library available from <http://razor.occams.info/code/semweb/>.