Arc Hydro - HydroID

Version 1.1 Final, July 2005

ESRI 380 New York St., Redlands, CA 92373-8100, USA • TEL 909-793-2853 • FAX 909-793-5953 • E-MAIL info@esri.com • WEB www.esri.com

Copyright © 2005 ESRI
All rights reserved.
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

The information contained in this document is subject to change without notice.

U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

ESRI, ARC/INFO, ArcCAD, ArcIMS, ArcView, *Business*MAP, MapObjects, PC ARC/INFO, SDE, and the ESRI globe logo are trademarks of ESRI, registered in the United States and certain other countries; registration is pending in the European Community. 3D Analyst, ADF, the ARC/INFO logo, AML, *ArcNews*, ArcTIN, the ArcTIN logo, ArcCOGO, the ArcCOGO logo, ArcGrid, the ArcGrid logo, ArcInfo, the ArcInfo logo, ArcInfo Librarian, ArcInfo—Professional GIS, ArcInfo—The World's GIS, ArcAtlas, the ArcAtlas logo, the ArcCAD logo, the ArcCAD WorkBench logo, ArcCatalog, the ArcData logo, the ArcData Online logo, ArcDoc, ArcEdit, the ArcEdit logo, ArcEditor, ArcEurope, the ArcEurope logo, ArcExplorer, the ArcExplorer logo, ArcExpress, the ArcExpress logo, ArcFM, the ArcFM logo, the ArcFM Viewer logo, ArcGIS, the ArcGIS logo, the ArcIMS logo, ArcNetwork, the ArcNetwork logo, ArcLogistics, the ArcLogistics Route logo, ArcMap, ArcObjects, ArcPad, the ArcPad logo, ArcPlot, the ArcPlot logo, ArcPress, the ArcPress logo, the ArcPress for ArcView logo, ArcReader, ArcScan, the ArcScan logo, ArcScene, the ArcScene logo, ArcSchool, ArcSDE, the ArcSDE logo, the ArcSDE CAD Client logo, ArcSdl, ArcStorm, the ArcStorm logo, ArcSurvey, ArcToolbox, ArcTools, the ArcTools logo, ArcUSA, the ArcUSA logo, *ArcUser*, the ArcView logo, the ArcView GIS logo, the ArcView 3D Analyst logo, the ArcView Business Analyst logo, the ArcView Data Publisher logo, the ArcView Image Analysis logo, the ArcView Internet Map Server logo, the ArcView Network Analyst logo, the ArcView Spatial Analyst logo, the ArcView StreetMap logo, the ArcView StreetMap 2000 logo, the ArcView Tracking Analyst logo, ArcVoyager, ArcWorld, the ArcWorld logo, Atlas GIS, the Atlas GIS logo, AtlasWare, Avenue, the Avenue logo, the *Business*MAP logo, the Data Automation Kit logo, Database Integrator, DBI Kit, the Digital Chart of the World logo, the ESRI Data logo, the ESRI Press logo, ESRI—Team GIS, ESRI—The GIS People, FormEdit, Geographic Design System, Geography Matters, GIS by ESRI, GIS Day, GIS for Everyone, GISData Server, *Insite*MAP, MapBeans, MapCafé, the MapCafé logo, the MapObjects logo, the MapObjects Internet Map Server logo, ModelBuilder, MOLE, the MOLE logo, NetEngine, the NetEngine logo, the PC ARC/INFO logo, PC ARCEDIT, PC ARCPLOT, PC ARCSHELL, PC DATA CONVERSION, PC NETWORK, PC OVERLAY, PC STARTER KIT, PC TABLES, the Production Line Tool Set logo, *Route*MAP, the *Route*MAP logo, the *Route*MAP IMS logo, Spatial Database Engine, the SDE logo, SML, StreetEditor, StreetMap, TABLES, The World's Leading Desktop GIS, *Water Writes*, and Your Personal Geographic Information System are trademarks; and ArcData, ArcOpen, ArcQuest, *ArcWatch*, ArcWeb, Rent-a-Tech, Geography Network, the Geography Network logo, www.geographynetwork.com, www.gisday.com, @esri.com, and www.esri.com are service marks of ESRI.

Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

Table of Contents

1.0 Introduction....1
2.0 HydroID Description....1
2.1 The Principles of HydroID Assignment....1
2.1.1 The Last Used HydroID Counter Concept 2
2.1.2 Access to LayerKeyTable and HydroIDTable 2
2.2 HydroID generation....3
2.2.1 Automated HydroID Generation 3
2.2.2 Manual HydroID Generation 3
2.3 Setting Up the HydroID Counter....4
2.3.1 Planning and Executing the HydroID Setup 4
2.4 HydroID Implementation Issues....5
2.5 Globalizing HydroID....5
2.5.1 ADO Connection 5
2.5.2 Creative HydroID "Seeding" 5
2.5.3 Single Project Processing 6
3.0 HydroID Setup....8
3.1 Workspace Connection Setup....8
3.1.1 DLLs Registration 8
3.1.2 INI files 9
3.2 ADO Connection Setup....11
3.2.1 DLLs Registration 11
3.2.2 INI files 11
3.3 Extension Registration....15
3.4 Tables Setup....16
3.4.1 Tables Design 16
3.4.2 Populate the tables 16
Appendix A – Setting up an Access DSN....19
Appendix B – Setting up a SQL Server DSN....21
Appendix C – Configuring an Oracle Net Service Name....23
Appendix D – Setting up an Oracle DSN....28

Table of figures

Figure 2.1 – Relationship between LayerKeyTable and HydroIDTable 2
Figure 2.2 – Example of LayerKeyTable and HydroIDTable setup 4
Figure 3.1 – UniqueIDMgrExt.ini for Workspace connection 9
Figure 3.2 – UniqueIdGenWKS.ini 10
Figure 3.3 – UniqueIDMgrExt.ini for ADO connection 11
Figure 3.4 – UniqueIDGenADO.ini 12
Figure 3.5 – Connection string to Access mdb 13
Figure 3.6 – Connection string to SQL Server 13
Figure 3.7 – DSN Connection 14
Figure 3.8 – Component Category Manager Window 15
Figure 3.9 – Populate HydroIDTable 17
Figure 3.10 – Populate LayerKeyTable 17
Figure 3.11 – Mistyped LayerKey 17
Figure 3.12 – UniqueIDGenWKS Error Message 18
Figure A1 – ODBC Microsoft Access Setup Window 19
Figure A2 – ODBC Microsoft Access Setup: Advanced Options 20
Figure B1 –DSN to SQL Server: DSN/Server Definition 21
Figure B2 – DSN to SQL Server: Login/Password Definition 22
Figure B3 – DSN to SQL Server: Default Database Definition 22
Figure C1 – Net8 Assistant Location 23
Figure C2 – Net8 Assistant Main Window 23
Figure C3 – Net Service Name Wizard: Welcome 24
Figure C4 – Net Service Name Wizard: Protocol 24
Figure C5 – Net Service Name Wizard: Protocol Settings 25
Figure C6 – Net Service Name Wizard: Service 26
Figure C7 – Connection Test 26
Figure C8 – Save Network Configuration 27
Figure D1 – Oracle8 ODBC Driver Setup 28

1.0 Introduction

By definition, HydroID is a unique identifier for features within a single geodatabase that conform to the Arc Hydro data model. Each feature, regardless of the feature class it belongs to has a unique HydroID (unlike the ObjectID that is unique within a feature class, but not unique within a geodatabase). HydroID is defined as a long integer (valid range between +/- 2,147,483,648, although only positive values should be used).

HydroID is used in key relationships in the Arc Hydro data model. "DrainID", "JunctionID", "NextDownID", "FeatureID", "FromNodeID", and "ToNodeID" attributes all contain the HydroID of relevant feature class. All established relationship classes within Arc Hydro use that, and it can also be the basis of establishing additional relationships (either through a relationship class or programmatically).

Arc Hydro tools manage the HydroID directly, that is, the tools will populate the HydroID automatically for any new feature (member of a feature class that follows the Arc Hydro data model) that is created through the tools. The tools also manage the related attributes that contain the HydroID. For example, when generating the hydro network using the "Hydro Network Generation" tool, the "JunctionID" field in the "DrainagePoint" feature class will be correctly populated with the HydroID of the hydro junction that the DrainagePoint points to. Several other functions maintain such relationships programmatically.

2.0 HydroID Description

2.1 The Principles of HydroID Assignment

The implementation of HydroID assignment is based on two key premises:

- Simple implementation for a general user who does not want to micromanage the HydroIDs between the feature classes. The end user does not have to do anything – every feature in a geodatabase will have a unique number, starting with 1 and incrementing by one for every new feature generated in any Arc Hydro compliant feature class.
- Complex implementation for a user who wants control over HydroIDs in each feature class. The user has the ability to specify a starting point for assignment of HydroIDs for each feature class. For example, the user can specify that the HydroIDs for a catchment feature class will start at number 100,000, for the drainage line feature class will start at number 200,000, and for the hydro junction and hydro edges feature classes will start at 300,000.

2.1.1 The Last Used HydroID Counter Concept

The HydroID for a new feature is obtained from a "last used HydroID" counter. This counter contains the last HydroID that was assigned to a new feature. When requested, this counter is incremented by one, and the HydroID number is passed on to the function that assigns it to the newly created feature.

To provide the flexibility where each feature class can have an independent counter, the counter concept is implemented through two related tables. The first table (LayerKeyTable) contains the relationship between the feature class name and the last used HydroID key, while the second table (HydroIDTable) contains the relationship between the last used HydroID key and the counter itself, Figure 1.



Figure 2.1 – Relationship between LayerKeyTable and HydroIDTable

For example, in Figure 1, layer name "Layer2" points to a layer key "Key B" (LayerKeyTable) and layer key "Key B" points to the HydroID "200035" (HydroIDTable). When a new feature in Layer2 is created, it will get the HydroID of 200,036 and the HydroID value for Key B will be incremented to 200,036. Also note that in the LayerKeyTable, Layer3 is also pointing to Key B. What that means is that all the new features in Layer3 feature class will be drawing from the same pool of HydroIDs as the new features in the Layer2 feature class.

If a feature class name is not specified in the LayerName field of the LayerKeyTable then the HydroID will be drawn from the OTHERS record in the HydroIDTable (e.g. 400,050 in figure 1).

2.1.2 Access to LayerKeyTable and HydroIDTable

Arc Hydro tools can access the LayerKeyTable and HydroIDTable in two ways. First is through the geodatabase's workspace, while the second one is through an ADO connection. Depending on the techniques used to access to the two tables, the LayerKeyTable and HydroIDTable exist either in the geodatabase (workspace access) or in an external database management system (ADO access).

When the two tables are in an external database (ADO connection) they can serve HydroIDs to several geodatabases (thus making the HydroID globally unique for all the databases retrieving the HydroIDs from the same set of LayerKeyTable and HydroIDTable).

With workspace access, the two tables are automatically generated for each geodatabase. For ADO connection, the two tables need to be set by the user in any supported RDBMS. The ADO connection needs also to be set by the user (Details on setting up the workspace and ADO access for HydroID generation are described in Sections 2.2 and 2.3).

2.2 HydroID generation

There are two techniques for HydroID generation:

- Automated – where HydroID is generated using the Arc Hydro tools that create new features in Arc Hydro compliant feature classes, or through the feature addition process using the standard ArcMap editing functions.
- Manual – where HydroID is generated using the "Assign HydroID" tool.

It is also possible to combine the two techniques if necessary.

2.2.1 Automated HydroID Generation

HydroIDs will be generated (and placed in the related ID fields) automatically by any of the Arc Hydro tools that generate new features in any of the Arc Hydro feature classes. The HydroIDs will also be assigned to any new feature belonging to an Arc Hydro feature class (actually any feature class that that has HydroID attribute in it) generated through standard editing process (create new feature) in ArcMap. For the latter, the "UniqueIDMgrExt" ESRI Editor extension has to be installed and configured. (Details on setting up the "UniqueIDMgrExt" extension are described in section 2.4).

When adding Arc Hydro features through the editor, only the HydroID will be assigned to the new feature. None of the related ID attributes will be populated automatically, since the relationship between the new feature and any other feature that might contain the related ID cannot be automatically discerned. The user will have to populate those relationships manually (if necessary).

2.2.2 Manual HydroID Generation

The "Assign HydroID" tool allows explicit assignment of HydroID values to a selected set of features. In general, this method should be only used to populate HydroIDs for those features and feature classes that are being imported into an Arc Hydro project.

None of the related ID attributes will be populated automatically. The HydroIDs for imported features should be assigned as soon as they are added to the project and then their allocation should be done automatically.

For an Arc Hydro project, only the "Arc Hydro" HydroID Method should be used.

2.3 Setting Up the HydroID Counter

When a new geodatabase using the Arc Hydro tools is saved for the first time, the LayerKeyTable and HydroIDTable are automatically created. The LayerKeyTable will have no entries, while the HydroIDTable will have only the OTHERS entry set to value of 0. In this (default) setup, all new features will draw the HydroID from the same source – the OTHERS record.

In order to assign different HydroID for a feature class, the name of that feature class needs to be entered in the LayerKeyTable (with its key) as well adding the specified key and the counter's initial value in the HydroIDTable. For example, assume you want to add two layers, Catchment and DrainageLine, with Catchment HydroID starting at 100,000 and DrainageLine HydroID starting at 200,000. The LayerKeyTable and HydroIDTable will look like in Figure 2.

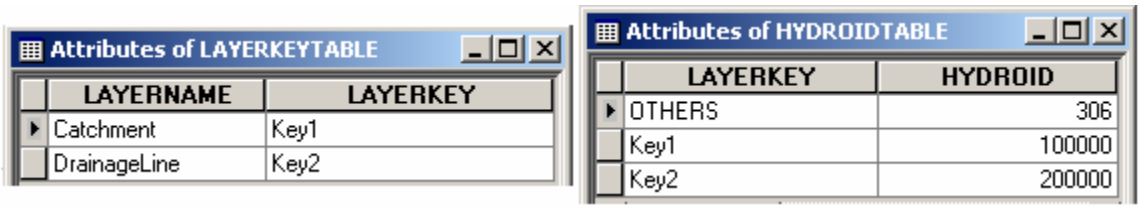


Figure 2.2 – Example of LayerKeyTable and HydroIDTable setup

The LayerKeyTable and HydroIDTable can and should be set as soon as possible. A feature class does not have to exist in a geodatabase before a reference to it can be placed in the LayerKeyTable.

2.3.1 Planning and Executing the HydroID Setup

For best results, when the user wants to fine tune the assignment of HydroIDs, careful planning will ensure the best results. The following steps should be followed:

- Identify as early as possible all the vector feature classes that will be generated/added to the geodatabase (the ones that will have the HydroID field).
  - For each identified feature class, define the name and estimate the maximum number of features that will be contained in it.
- Design the numbering schema.
  - Identify which feature class will point to which key.

- Identify starting number for each key. Make sure that the initial HydroID for each key will not be reached by HydroID for another key after all features in feature classes pointing to that key are assigned.
- Decide whether the HydroID will be shared between different geodatabases (globally unique), and which method to use to ensure that.
  - Setup the ADO connection and tables in an external RDBMS if the ADO approach is chosen.
  - Make the numbering schema unique if workspace approach is used.

- Create the required entries in the LayerKeyTable and HydroIDTable as soon as possible (ideally before any feature classes are added to the geodatabase).

2.4 HydroID Implementation Issues

In principle, the implementation of HydroID generation is fairly simple and straightforward. To ensure proper implementation the following key points must be correctly implemented:

- Names of feature classes must be known in advance.
- The LayerKeyTable and HydroIDTable must be setup before HydroID's can be assigned.
- The HydroID for two keys can get out of sync (pointing to the same number) if incorrect numbering schema is used (e.g. underestimating number of features to be generated).

2.5 Globalizing HydroID

In some instances, it is beneficial to have HydroID that is unique across different geodatabases (global HydroID), that is, HydroID becomes globally unique, and not only unique within a single geodatabase. There are two ways of achieving this.

2.5.1 ADO Connection

When machines that will be creating different geodatabase that need to have the same source of HydroIDs are on the same network, the simplest solution is to use the ADO approach for accessing the LayerKeyTable and HydroIDTable in a commonly accessible RDBMS. Once set up, the ADO connection will point to the same tables for all Arc Hydro projects.

2.5.2 Creative HydroID "Seeding"

If using an ADO connection is not an option (e.g. if processing will be done on machines that do not have access to a shared RDBMS), the HydroID seeds can be initiated within

each geodatabase in such a way that they will be unique across all the geodatabases taken into consideration.

Consider the earlier example in which there were two feature classes - Catchment and DrainageLine, with Catchment HydroID starting at 100,000 and DrainageLine HydroID starting at 200,000. Assume that catchment analysis has to be performed at three different larger areas. Initially, each area will be processed on its own. After the initial study, the results from the three areas will be combined into a single feature class.
If the Catchment HydroID starts at 100,000 and DrainageLine HydroID starts at 200,000 for each of the geodatabases, when combining the features from the three geodatabases into a single feature class, the HydroID will not be unique in the combined feature class. To avoid this issue, the following approach can be used:

- Assign a unique reference ID for each of the three large areas, for example, 1, 2, and 3.
- Use those reference IDs to offset the HydroID counter for each feature class. For example, offset the initial HydroID for each feature class by reference ID times 10,000,000. Thus, the Catchment HydroID for area 1 would start at 10,100,000, for area 2 at 20,100,000, and for area 3 at 30,100,000. Similar adjustment can be made for DrainageLine or any other feature class.
- When assigning the starting seeds for HydroID, attention has to be made to the total number of features for all the features in all the geodatabases getting the HydroID from the same seed (to exclude the possibility of HydroID "overflow").
- Similar approach must be taken for the OTHERS key in the HydroIDTable (to ensure that none of the features, for which a specific counter has not been set, will have the matching HydroIDs).

With this kind of numbering scheme, all the features in different geodatabases will have a unique HydroID and no problems will be encountered when merging the data into a single geodatabase.

2.5.3 Single Project Processing

The final strategy is to process all the data in a single project/geodatabase. If possible, all the initial data can be processed through a single geodatabase. This does not mean that the entire base data need to be in a single feature class, but rather that the processing is performed in a single project/geodatabase.

Consider the following scenario. Catchments for a large area are to be generated from DEMs and their derived grids. Due to the large size of the DEMs to cover the entire area, they have been broken down into several non-overlapping areas. These DEMs have been preprocessed and stored separately in order to manage their size. Each DEM has flow

direction and flow accumulation derived grids, as well as corresponding synthetic stream and stream link grids.

At this point, it is possible to start processing the catchments in a unified manner. The following steps should be taken.

- Create a new ArcMap project and make sure the Arc Hydro toolbar is active.
- Make sure all grids have a unique name. For example, flow direction grid for area 1 is called fdg1, while for area 2 is fdg2. These have to be feature class names (not entries in the legend in the ArcMap), so if renaming is necessary, it needs to be done in ArcCatalog using the rename function (right-click on the feature class to be renamed and choose "Rename").
- Add all the required grid data to the map (include grids from separate areas).
- Perform any other required grid processing (if necessary). Make sure that the grids being created have unique names (as specified in the second step).
- Save the project. After first Arc Hydro tool operation, the LayerKeyTable and HydroIDTable will be created automatically if not already present. For the moment, only the OTHERS field will exist, and will have the initial HydroID set to 0.
    - If necessary or desired, set up the HydroID numbering schema. If none is defined, all the features will share the same HydroID counter (as pointed to by OTHERS record in the HydroIDTable).
        - Make sure that each feature class that has HydroID field has unique name (e.g. catchment1 for the first area, catchment2 for the second area, etc.).
- Start vector processing (where HydroIDs are generated). Since all the new features are in the same geodatabase, their HydroIDs will be unique and assigned following the HydroID schema set up in the previous step (if any was set).

3.0 HydroID Setup

The steps involved in the manual set up of the HydroID application depend on the way the 2 tables LayerKeyTable and HydroIDTable are to be accessed. These tables need to be in the same location, and they may be accessed s on the generation application he tables needed to generate the HydroIDs have to be at the same location. This location may be accessed in one of the two following ways:

- **Workspace connection**: the feature layers and the two tables must be stored in the same personal or ArcSDE geodatabase. The advantage is that there are no worries about connection. However, only geodatabase features can be edited (i.e. no shapefile or coverage). Moreover, the tables need to be created in ArcCatalog.

- **ADO connection**: the feature layers and the two tables may be in different locations. The tables do not need to be created in ArcCatalog (no ObjectID field needed). Geodatabase elements, shapefiles and coverages can be edited.

One type of connection only can be used at any given time. The files needed for the setup depend on the type of connection used. The next 2 sections describe the setup for each case.

3.1 Workspace Connection Setup

The 5 following files (3 dlls and 2 ini files) are needed to set up the HydroID application with a workspace connection:

- UniqueIDLib.dll
- UniqueIDMgrExt.dll
- UniqueIDMgrExt.ini
- UniqueIDGenWKS.dll
- UniqueIDGenWKS.ini

3.1.1 DLLs Registration

The dlls files required by the HydroID application need to be registered in the following way:

- Locate regsvr32.exe, which should be under c:\winnt\system32.
- Create a shortcut for this file.
- Drag and drop the files to register on the shortcut.

The following dlls need to be installed and registered:

- UniqueIDLib.dll
- UniqueIDMgrExt.dll
- UniqueIDGenWKS.dll

3.1.2 INI files

Two ini files need to be set up:

- UniqueIDMgrExt.ini
- UniqueIDGenWKS.ini

These files must be located in the same directory as the dlls.

UniqueIDMgrExt.ini

This file defines the type of connection (ADO or Workspace) used to access the tables LayerKeyTable and HydroIDTable. To set up a Workspace connection, the parameter UniqueIdServerName needs to be set to UniqueIdGenWKS.UniqueIdGen. The UnidqueIDMgrExt will then be using the unique id generator that accesses the database and UniqueId tables via an IWorkspace interface.

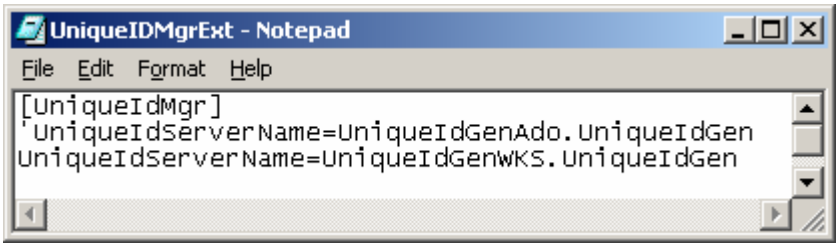


Figure 3.1 – UniqueIDMgrExt.ini for Workspace connection

UniqueIDGenWKS.ini

The UniqueIDGenWKS.ini file sets up the configuration for the UniqueIDGenWKS.dll. It contains the definition of the two tables, HydroIDTable and LayerKeyTable. There is no connection information: the application looks for the tables in the workspace where the feature being edited is located.

The information required is:

For the UniqueidTable (HydroIdtable)

- UniqueIdTable: Name of the UniqueIdTable
- UniqueIdField: Field containing the unique Ids
- LayerKeyField: Field containing the layer keys

For the LayerKeyTable

- LayerKeyTable: Name of the LayerKeyTable
- LayerNameField: Field containing the names of the layers
- KeyNameField: Field containing the layer keys
- KeyOther: Key used for all the layers which are not explicitly defined in the LayerNameField

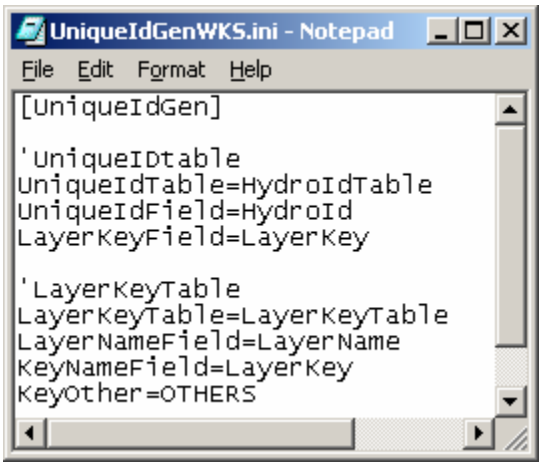


Figure 3.2 – UniqueIdGenWKS.ini

3.2 ADO Connection Setup

The 5 following files (3 dlls and 2 ini files) are needed to setup the HydroID application with an ADO connection:

- UniqueIDLib.dll
- UniqueIDMgrExt.dll
- UniqueIDMgrExt.ini
- UniqueIDGenAdo.dll
- UniqueIDGenAdo.ini

3.2.1 DLLs Registration

The dlls files required by the HydroID application need to be registered in the following way:

- Locate regsvr32.exe, which should be under c:\winnt\system32.
- Create a shortcut for this file.
- Drag and drop the files to register on the shortcut.

The following dlls need to be installed and registered:

- UniqueIDLib.dll
- UniqueIDMgrExt.dll
- UniqueIDGenAdo.dll

3.2.2 INI files

Two ini files need to be set up:

- UniqueIDMgrExt.ini
- UniqueIDGenAdo.ini

These files must be located in the same directory as the dlls.

UniqueIDMgrExt.ini

This file defines the type of connection (ADO or Workspace) used to access LayerKeyTable and HydroIDTable. To set up an ADO connection, the parameter UniqueIdServerName needs to be set at UniqueIdGenAdo.UniqueIdGen. The UniqueIDMgrExt will then be using the unique id generator that accesses the database and UniqueId tables via ADO.

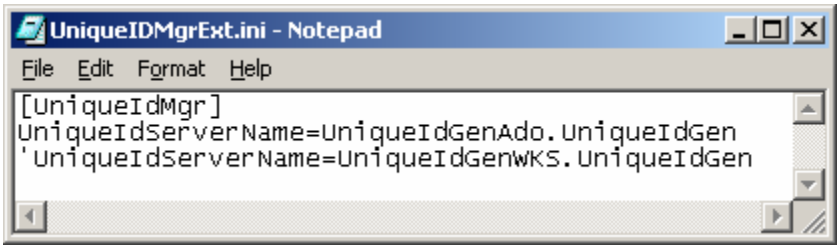


Figure 3.3 – UniqueIDMgrExt.ini for ADO connection

UniqueIDGenAdo.ini

This UniqueIDGenAdo.ini file is used when the connection to the two tables is done with ADO. It sets up the configuration for the UniqueIDGenAdo.dll. HydroIDTable and LayerKeyTable, and the feature layers, may have a different location. In addition to defining the structure of these two tables, the ini also contains the connection information needed to access them. This may be any ADO supported connection, in particular Access, SQL Server or Oracle database connections.

Note: the information separated by ";" need to be on the same line. The file displayed below was modified for displayed purposed.

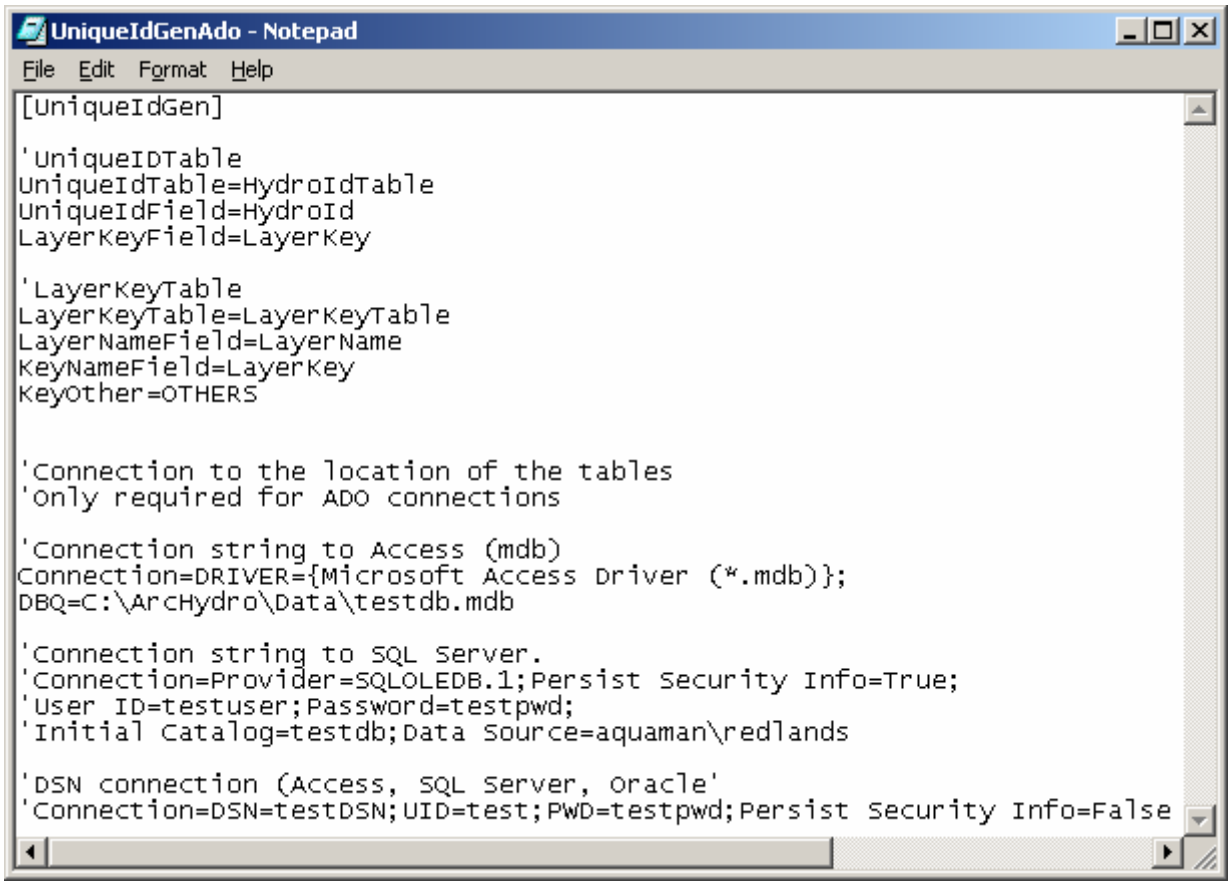


Figure 3.4 – UniqueIDGenADO.ini

The information needed to establish the connection to the workspace, where UniqueIDTable (HydroIDTable) and LayerKerTable are located, is detailed in the remaining of this section.

- **Connection string to Access mdb**

A connection to an Access database (mdb file) can be defined by using a connection string. The full path to the mdb file is set as the DBQ parameter.

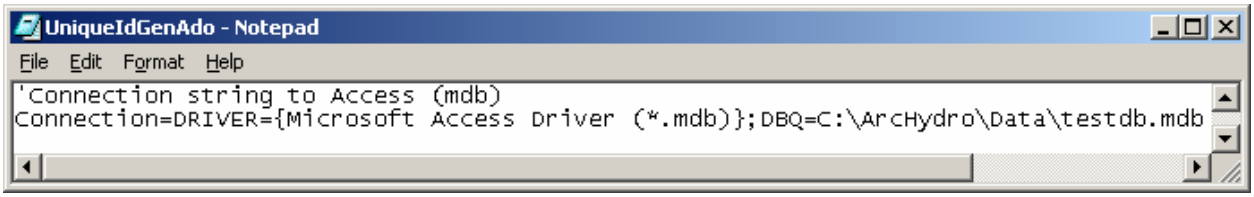


Figure 3.5 – Connection string to Access mdb

- **Connection String to SQL Server**

The connection information needed to connect to SQL Server is the following (note: all the parameters separated by ; should follow each other on the same line).

- User ID
- Password
- Initial Catalog: database
- Data source: SQL Server name. For SQL Server 7, same as the server name. For SQL Server 2000, name of the SQL Server instance.

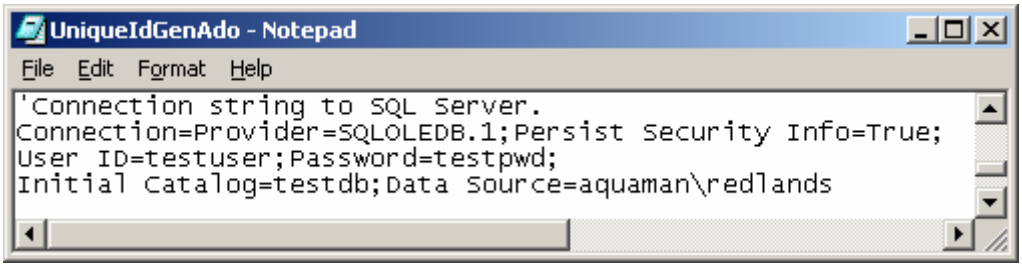


Figure 3.6 – Connection string to SQL Server

- **DSN Connection (Access, Oracle, SQL Server…)**

The connection information required to connect via a DSN is the Data Source Name set up on the client (refer to the appendices for instructions on how to set up a DSN for Access, SQL Server and Oracle). The user name (UID) and password (PWD) are required only if needed to connect to the database.

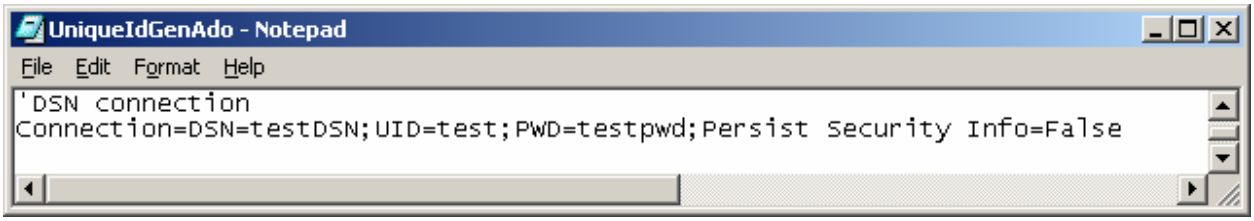


Figure 3.7 – DSN Connection

3.3 Extension Registration

The HydroID application is made available as an edit event in ArcMap by registering the file UniqueIDMgrExt.dll with the ESRI Editor Extensions.

To manually register the DLL as an ArcMap extension:

- Locate categories.exe in the folder where ArcGIS is installed (typically Arcgis\arcexe81\Bin).
- Run categories.exe. The Component Category Manager opens displaying a list of categories.
- Scroll down to ESRI Editor Extensions and highlight this category by clicking it once with the left mouse button.
- Click Add Object…, and then navigate to and open UniqueIDMgrExt.dll.
- Add UniqueIdMgr in the Add Objects windows.

UniqueIDMgrExt.UniqueIdMgr is added to the list of ESRI Editor Extension.

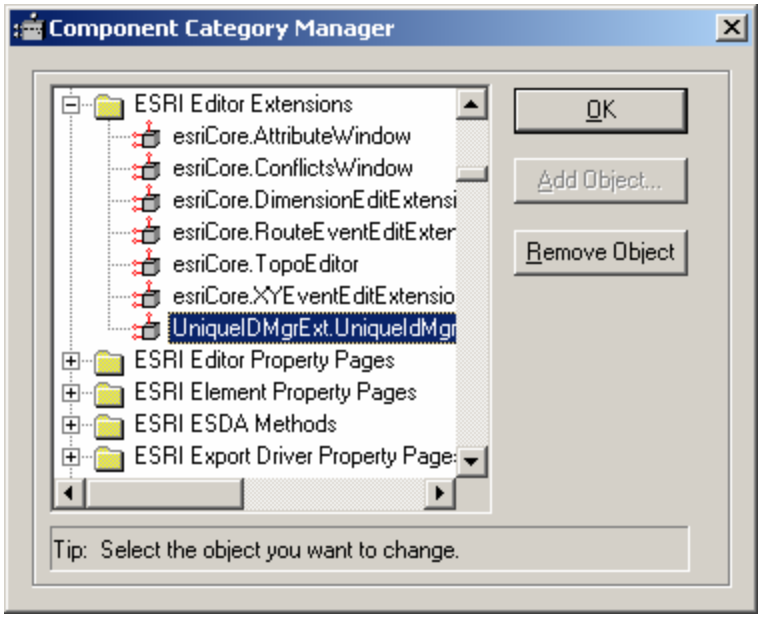


Figure 3.8 – Component Category Manager Window

3.4 Tables Setup

3.4.1 Tables Design

The two tables used by the application have the following structures:

HydroIDTable

Field	Type	Description
LayerKey	String	Layer key
HydroID	Long	Last used hydroid value.

LayerKeyTable

Field	Type	Description
LayerName	String	Layer name
LayerKey	String	Layer key related to the layer key in the HydroIDTable

The tables must have the same location. When using a workspace connection, they must be in the same workspace as the layers they are managing.

Note: for a workspace connection, these tables **MUST** be created in ArcCatalog.

3.4.2 Populate the tables

Once the tables have been created, they need to be populated with the chosen layers/keys/values.

- Open ArcMap and add the HydroIDTable and the LayerKeyTable to the Table of Contents. Open these tables.
- Start an editing session by selecting Editor>Start Editing.
- Enter in the HydroIDTable the different keys with their associated HydroID value. Note that this value is the last used hydroid. The next hydroid value to use is this value, incremented by 1. Enter the default layer key, which is set in the ini file (e.g. KeyOther=Others).

Note: There is no control on the unicity of the HydroID values in the database. You must make sure that the different keys will not overlap by choosing intervals, which are big enough.

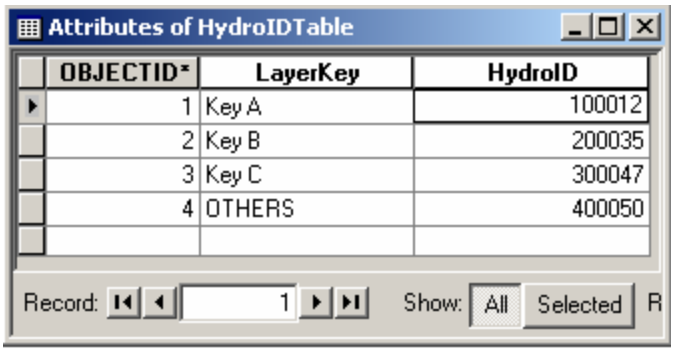


Figure 3.9 – Populate HydroIDTable

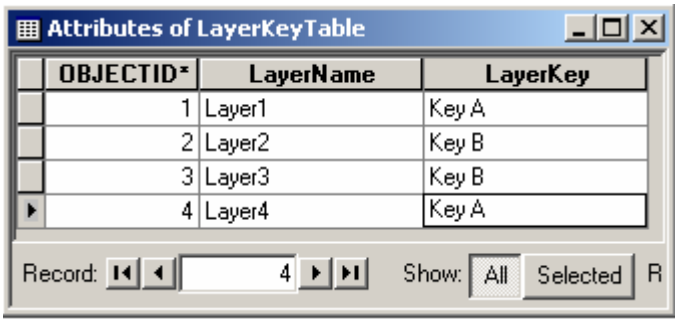


Figure 3.10 – Populate LayerKeyTable

Note: Make sure that the keys defined are correct and defined in HydroIDTable. If not you will get an error message when using the tool.

For example, the key for layer1 is mistyped: it should read Key A instead of KeyA.

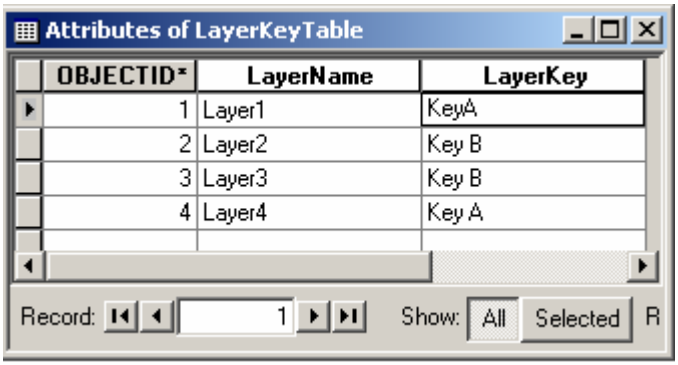


Figure 3.11 – Mistyped LayerKey

When trying to add a new feature in the feature class layer1, the following error message appears. The hydroid field in layer1 is updated with –1, and the hydroid in the HydroIDTable is not incremented.

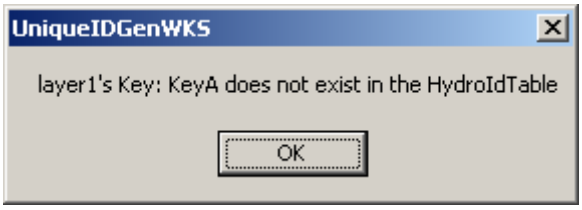


Figure 3.12 – UniqueIDGenWKS Error Message

Appendix A – Setting up an Access DSN

A Data Source Name (DSN) to connect to an Access (.mbd) is created according the following steps:

- Click Start > Settings > Control Panel
- In Windows 2000, click Administrative Tools
- Click Data Sources (ODBC) to open the ODBC Data Source Administrator.
- Click the System DSN tab and click Add.
- Select the driver for Microsoft Access Driver (*.mdb) and click Finish.
- In the ODBC Microsoft Access Setup window, enter the data source name and select the mdb database to connect to.



Figure A1 – ODBC Microsoft Access Setup Window

- To set a login/password to access the database (if required), click Advanced… and enter the login and password in the Set Advanced Options window.

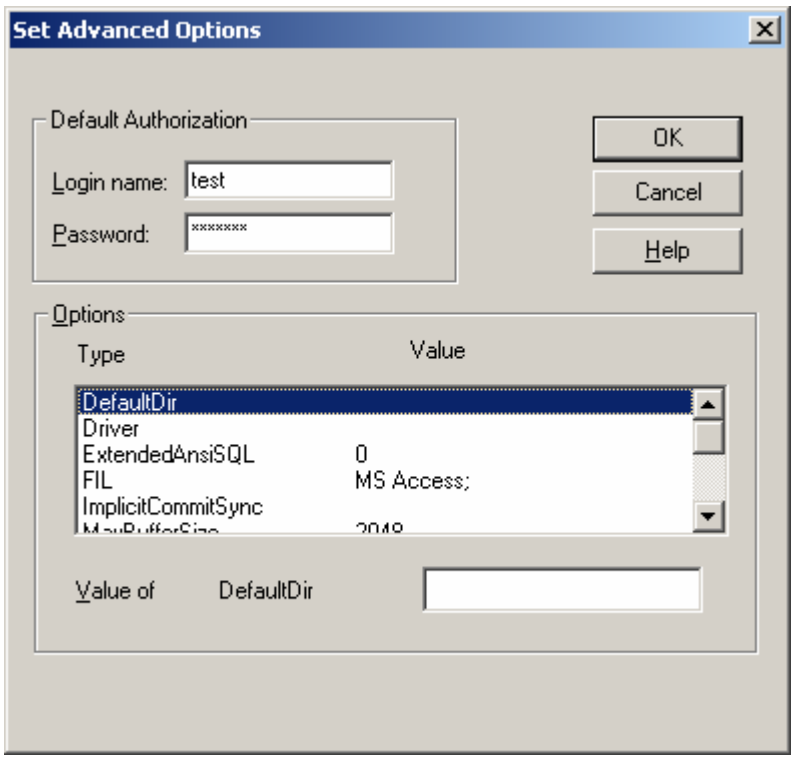


Figure A2 – ODBC Microsoft Access Setup: Advanced Options

Appendix B – Setting up a SQL Server DSN

- Click Start > Settings > Control Panel
- Click Data Sources (ODBC) to open the ODBC Data Source Administrator.
- Click the System DSN tab and click Add.
- Select the driver for which a data source must be created (e.g. SQL Server) and click Finish.
- Enter the Data Source Name and the SQL Server instance. Click Next.

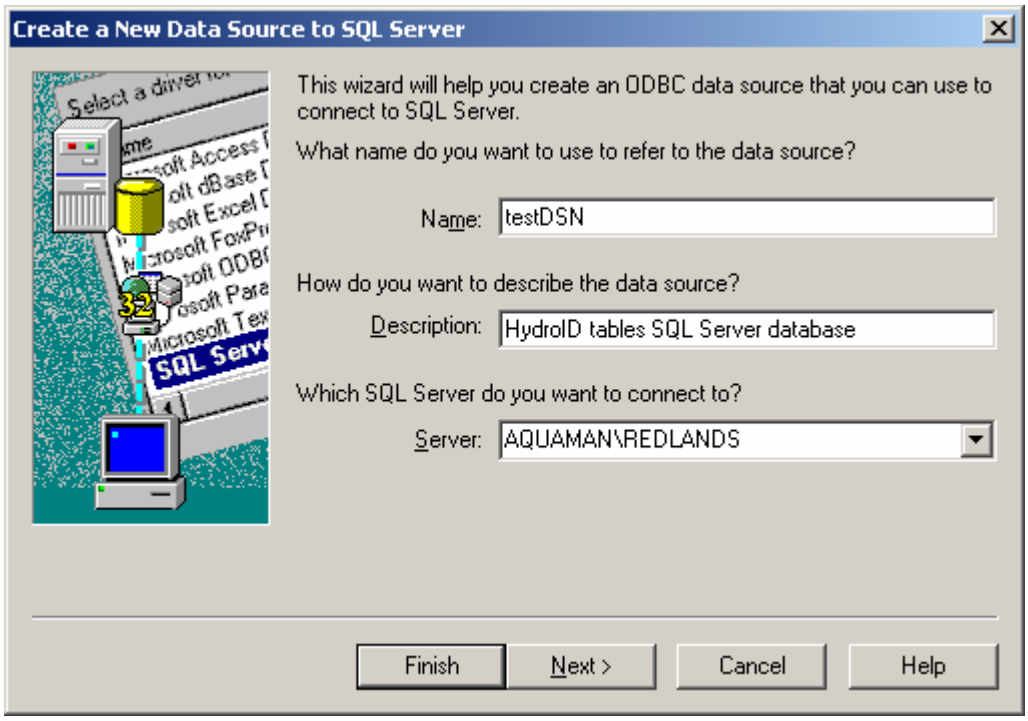


Figure B1 –DSN to SQL Server: DSN/Server Definition

- Enter the login and password to access the SQL Server database.

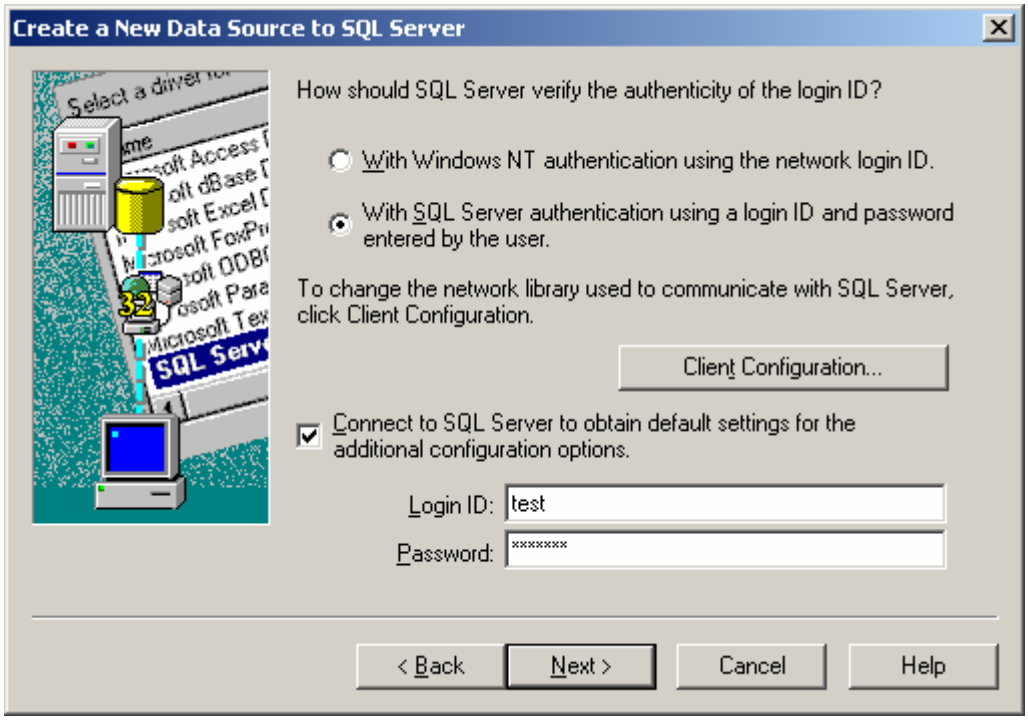


Figure B2 – DSN to SQL Server: Login/Password Definition

- Under Database, click Select and browse for the external database. Click OK.

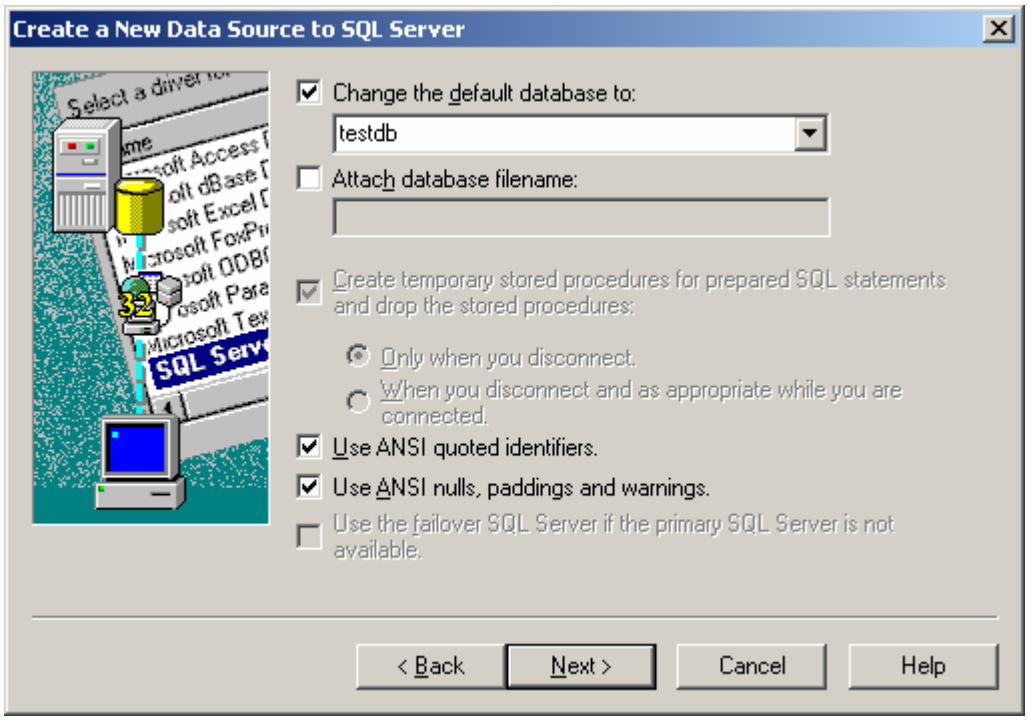


Figure B3 – DSN to SQL Server: Default Database Definition

- Click OK.

Appendix C – Configuring an Oracle Net Service Name

To set up a DSN for Oracle on a client machine, you need first to install the Oracle client and networking components to use the Oracle driver. You need to configure a Net Service Name so that the client can access the Oracle database.

Install the Oracle Client and open the Net8 assistant

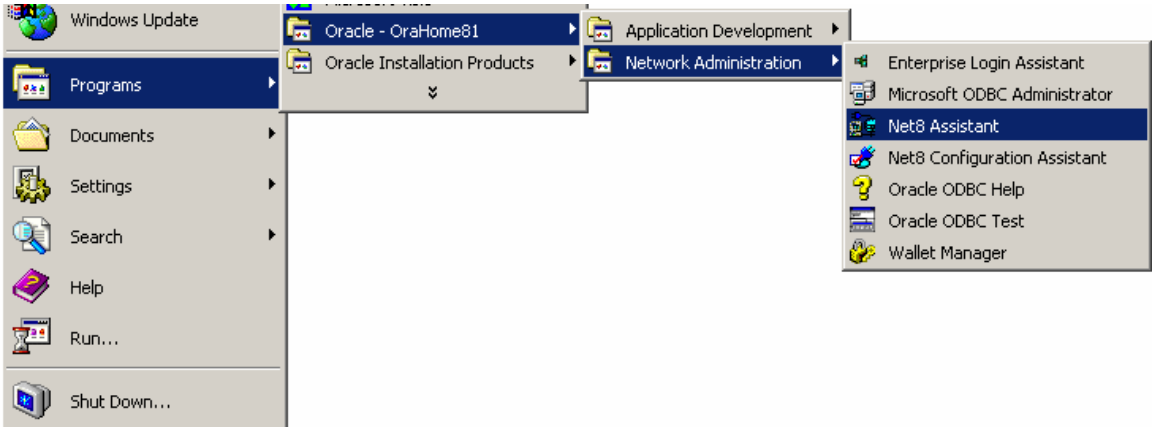


Figure C1 – Net8 Assistant Location

In the Net8 Assistant window under Local, Click Service Naming, click the + (create)

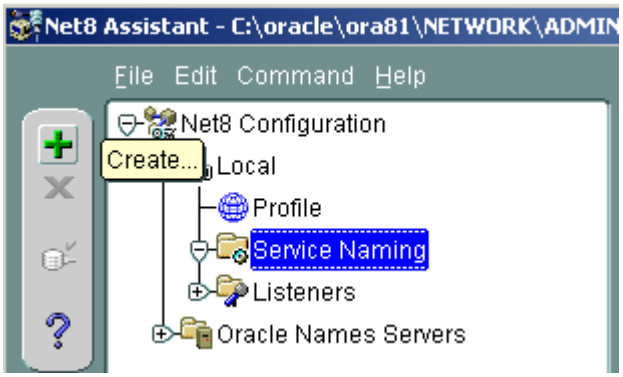


Figure C2 – Net8 Assistant Main Window

Follow the following steps:

Enter Net Service Name that you want to use to connect to the database. This can be any name you choose (e.g. archydro), but you will use this name when setting up the ODBC connection.

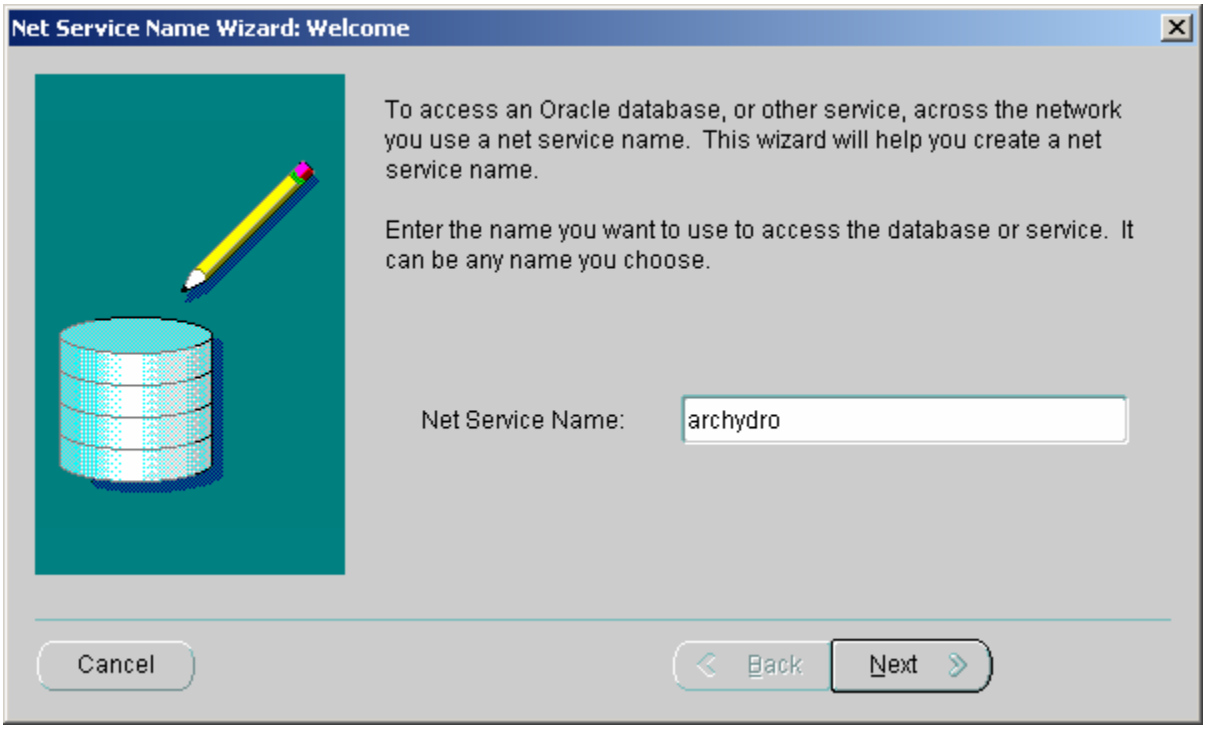


Figure C3 – Net Service Name Wizard: Welcome

Select TCP/IP as the network protocol.

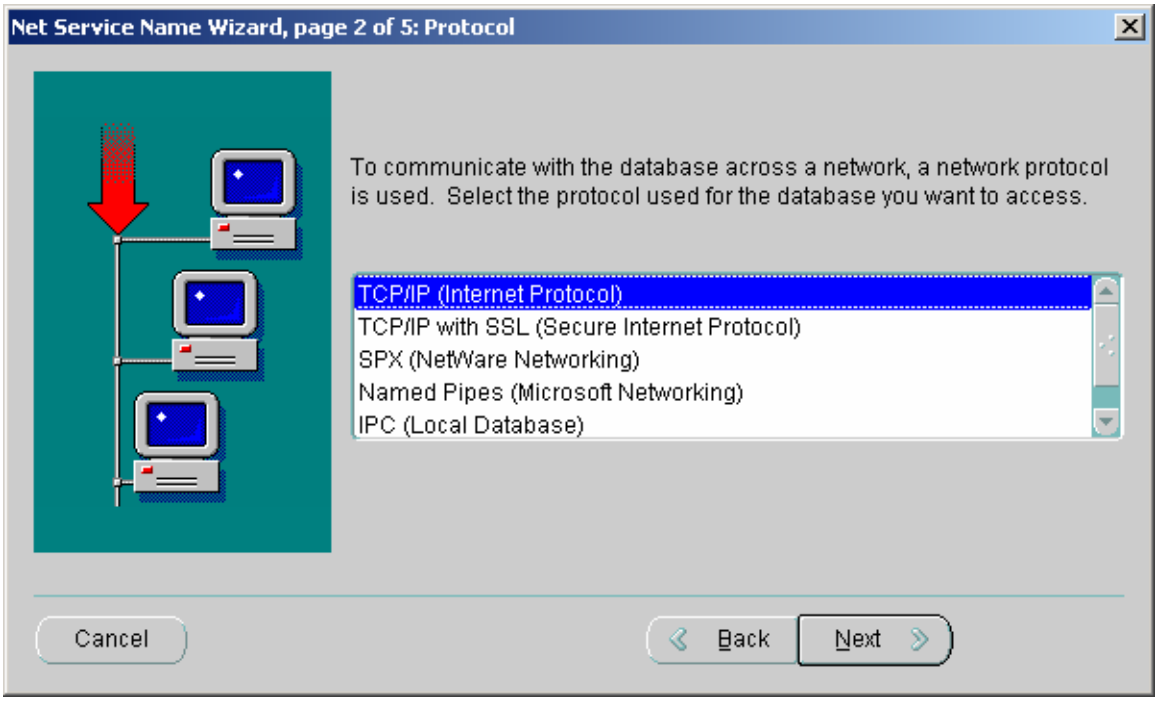


Figure C4 – Net Service Name Wizard: Protocol

Enter the name of the server on which the database is located (e.g. aqualad)

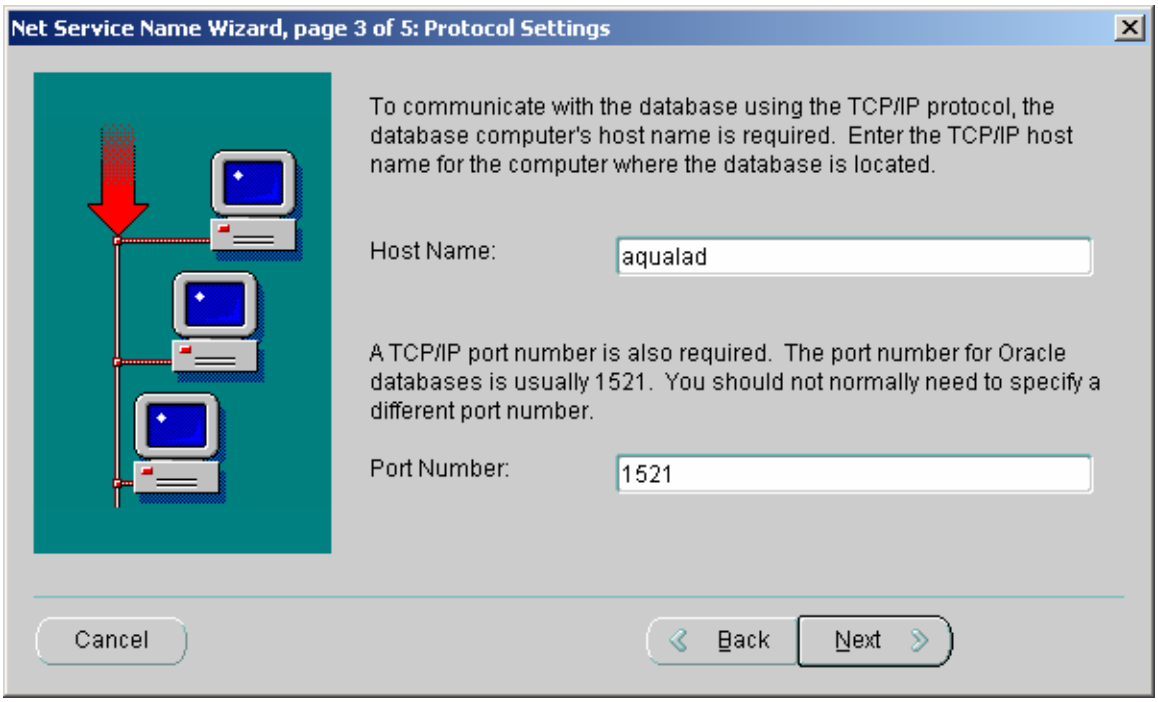


Figure C5 – Net Service Name Wizard: Protocol Settings

Enter the Service Name (= global database name). Note that you may look up this service name in the file tnsnames.ora located on the server in the Oracle Home under OracleHome/Network/admin (e.g. SERVICE_NAME = archydro.esri.com).

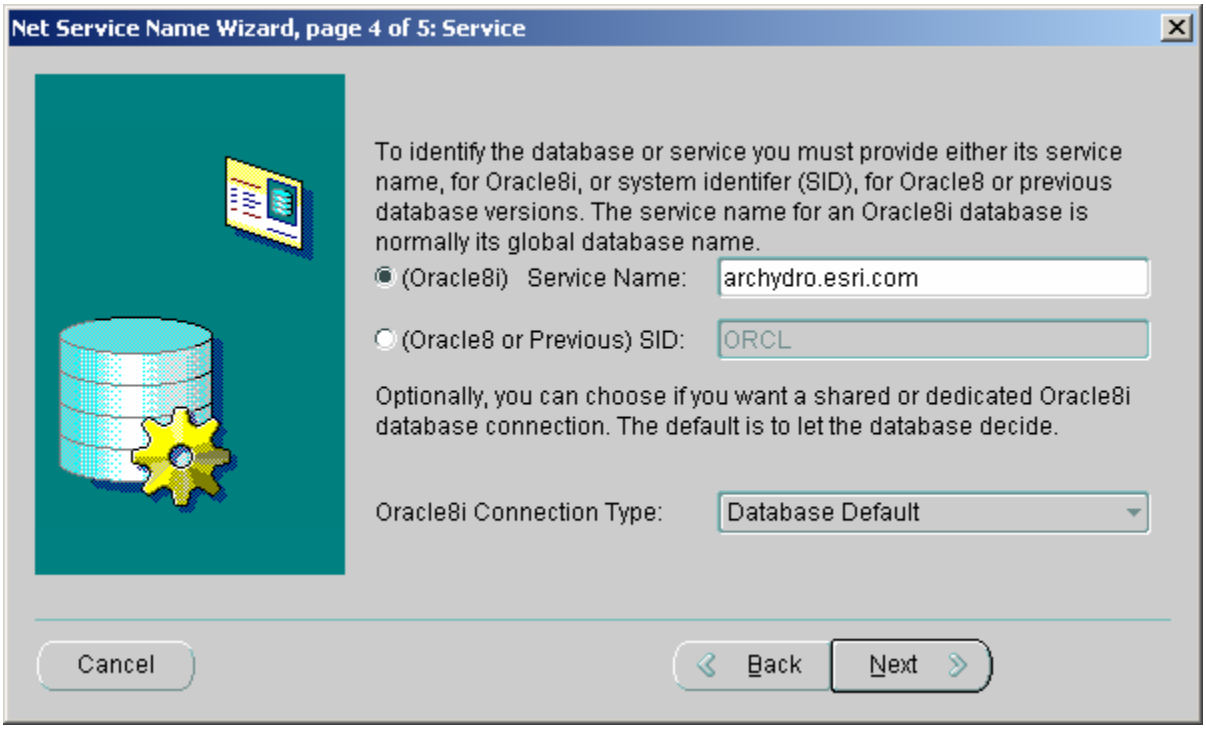


Figure C6 – Net Service Name Wizard: Service

Connection test (Note: you may need to change the login).

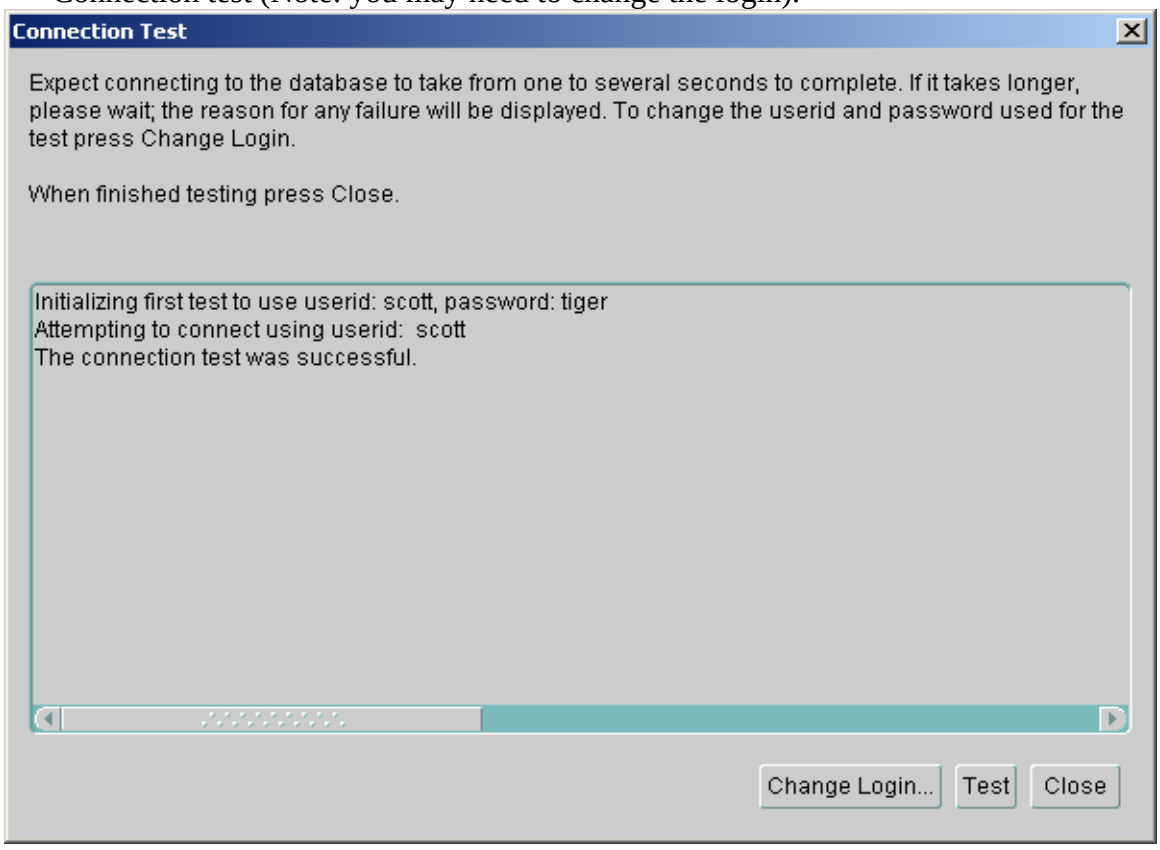


Figure C7 – Connection Test

After testing, click close/Finish.

WARNING: Make sure that you save the network configuration.

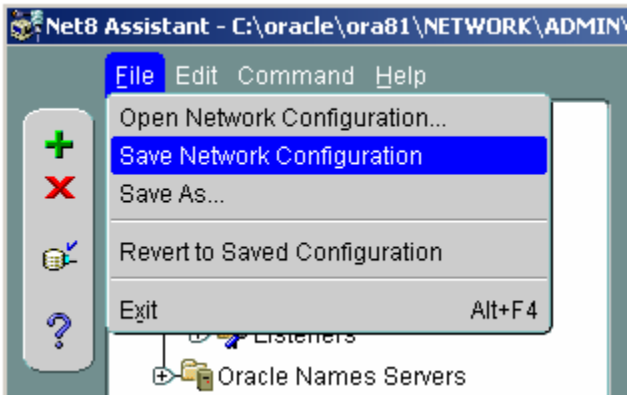


Figure C8 – Save Network Configuration

Appendix D – Setting up an Oracle DSN

They are several ways you can create an Oracle ODBC connection. This section describes how to set up the ODBC by using Data Sources (ODBC) located in the Administrative Tools within the Control Panel in Windows 2000 (Note: in Windows NT, it is located in the Control Panel).

Click Start > Settings > Control Panel

Click Data Sources (ODBC) to open the ODBC Data Source Administrator.

Click the System DSN tab and click Add.

To create an ODBC connection click New…

Select the driver Microsoft ODBC for Oracle and click Finish.

In the Oracle8 ODBC Driver Setup, enter the Data Source Name (same as DSN in the ini file).

Enter the Service Name (Note: this is the Global Database Name in Oracle.).

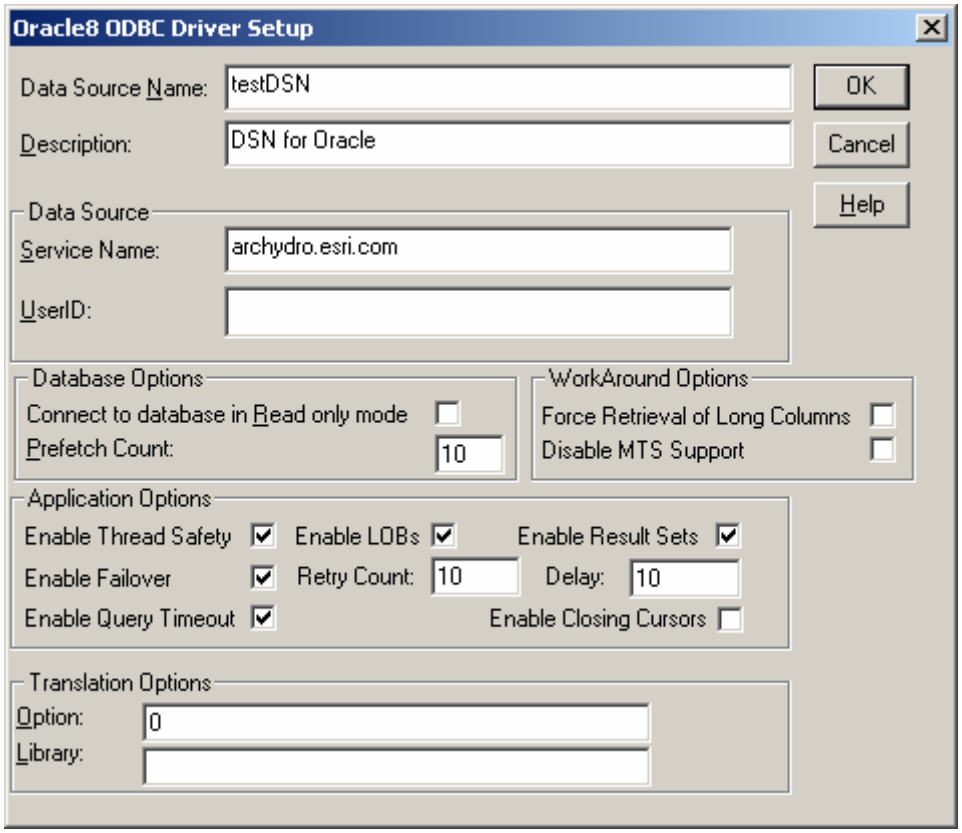


Figure D1 – Oracle8 ODBC Driver Setup