Morfik Zeos Package

The Morfik Zeos package contains classes that support direct connections to client-server databases through native connectors, removing the need to use ODBC. This package is based upon the Open source ZeosLib code which you can find at http://sourceforge.net/projects/zeoslib/. The initial package offers native connections for MSSQL and MySQL databases.

Getting Started

To use the Morfik Zeos package simply add the package to the project by utilizing the "Used Packages" command on the project ribbon (see Figure 1 **Used Packages Command**), or simply drag the package file onto the application main client area. Once added there will appear new connection options in the Data Connectors dialogs which are discussed below.

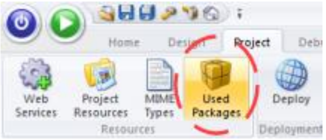
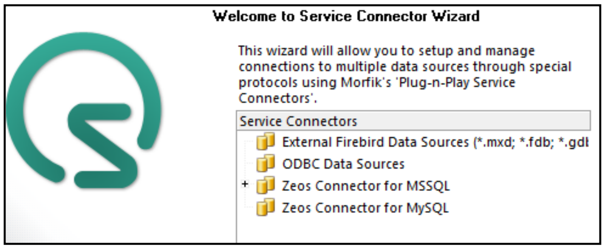


Figure 1. Used Packages Command

Using the New Data Connectors

Once the Zeos package has been added to a project, starting the Data Connectors wizard will add the MSSQL and MySQL options seen in Figure 2 **Data Connector Options**.



Welcome to Service Connector Wizard

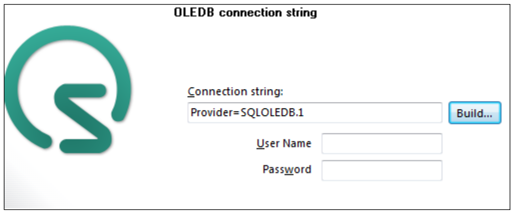
This wizard will allow you to setup and manage connections to multiple data sources through special protocols using Morfik's 'Plug-n-Play Service Connectors'.

Service Connectors
- External Firebird Data Sources (*.mxd; *.fdb; *.gdl
- ODBC Data Sources
- Zeos Connector for MSSQL
- Zeos Connector for MySQL

Figure 2. Data Connector Options

Selecting the Zeos Connector for MSSQL

To create a new connector for MSSQL, highlight the MSSQL option above, select the Add radio button and click the Next button. The build connection dialog will appear as seen in Figure 3 **Build the Connection String**.



OLEDB connection string

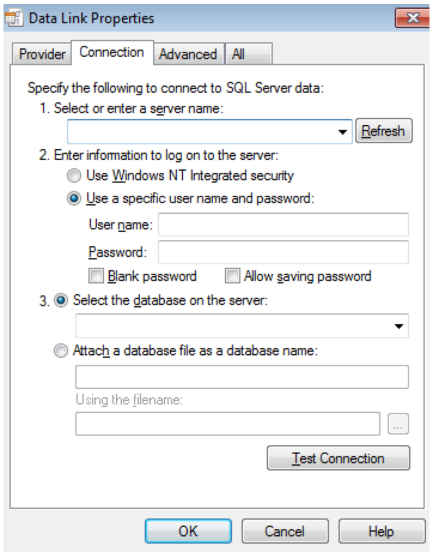
Connection string: Provider=SQLOLEDB.1

User Name

Password

Figure 3. Build the Connection String

Clicking on the Build button brings up a fairly standard dialog box seen in Figure 4 **Build the Connection String Step 2**.



Data Link Properties

Provider | Connection | Advanced | All

Specify the following to connect to SQL Server data:

1. Select or enter a server name:

2. Enter information to log on to the server:

○ Use Windows NT Integrated security

● Use a specific user name and password:

User name:

Password:

☐ Blank password ☐ Allow saving password

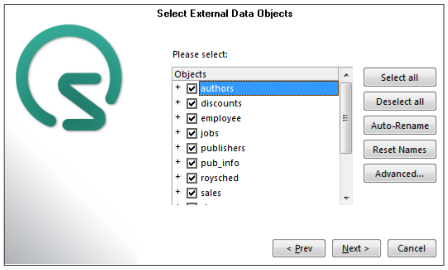
3. ● Select the database on the server:

○ Attach a database file as a database name:

Using the filename:

Figure 4. Build the Connection String Part 2

After completing the dialog box and testing the connection, saving the build string and clicking the Next button, a dialog box containing a list of tables available will be displayed for selection. Check the ones to be available in the application. In Figure 5 **Table Selection** all the available tables in the standard MSSQL Pubs sample database are selected.


Select External Data Objects

Please select:

Objects
- ☑ authors
- ☑ discounts
- ☑ employee
- ☑ jobs
- ☑ publishers
- ☑ pub_info
- ☑ roysched
- ☑ sales

Figure 5. Table Selection

In the next step, provide a name for the connection and proceed to **Finish** the wizard. After completing these steps, the tables will have been added to the project manager and will be available as any other table would be. (See Figure 6 **Tables in the Project Manager**)

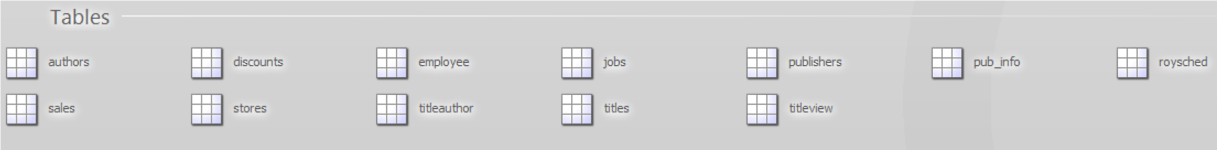


Figure 6. Tables in the Project Manager

Creating a Query

Adding a new visual query is standard except for the option to select the connector to use before proceeding to design the query. The same is true in creating a passthrough query but, for those used to Firebird, field and table names are not case-sensitive so double-quotes are not generally needed.

Using Zeos Connectors at Runtime

When accessing the database in code at runtime, add the **SystemCatalog** to the **Uses** clause and then select the MSSQL service connector by the name given to it when created in the wizard, as seen in the sample code below:

```
Public Overrides Sub Execute

    Dim Connector As TDatabaseServiceConnector

    Dim Connection As TDatabaseServiceConnection

    Dim Recordset As TAnyDBRecordset

    Dim sql As String

    Dim s As TStringList


    sql = "SELECT ""titles"".""title"" as title " + "FROM ((""authors"" INNER JOIN
""titleauthor"" ON (""authors"".""au_id""=""titleauthor"".""au_id"")) " + "INNER JOIN ""titles""
ON (""titleauthor"".""title_id""=""titles"".""title_id"")) " + "WHERE ""authors"".""au_id"" =
:""prm"""
```

```
    Connector = Catalog().GetServiceConnectorByID("Zeos MSSQL1") As TDatabaseServiceConnector

    Connection = Connector.CreateConnection()

    Try

        Recordset = Connection.CreateDBRecordset(sql, "")

        Recordset.Prepare()

        Recordset.ParamByName("prm").AsString = au_id

        Recordset.Open()

        s = New TStringList()

        While Not Recordset.EOF()

            s.Add(Recordset.FieldByName("title").AsString)

            Recordset.[Next]()

        End While

        s.Delimiter = "|"

        au_id = s.DelimitedText

    Finally

        Connection.Free()

    End Try

End Sub
```

That’s it! Now enjoy expanded native database connectivity!