SYNAMETRICS TECHNOLOGIES

A division of IndusSoft Technologies, Inc.

WinSQL for JDBC Users' Guide

WinSQL for JDBC User's Guide

© Synametrics Technologies, a division of IndusSoft Technologies Inc.
27 Sand Hill Ct. Jamesburg, NJ 08831
Phone 732-605-7088 • Fax 732-656-9745

Table of Contents

Chapter 1

Introduction

WinSQL is a must-have utility for all programmers and DBAs.

Welcome to WinSQL for JDBC. This guide details the features and options of WinSQL, a must-have utility for programmers and DBAs who deal with databases in their day-to-day life.

This guide is intended for engineers and developers who work with database and are familiar with the Structured Query Language (SQL). Please note that this guide assumes that you have working knowledge of SQL and therefore, does not attempt to show how SQL works.

About WinSQL for JDBC

WinSQL for JDBC is a 100% Java based application that uses JDBC to connect to different data sources. Its key features are:

- Multi-platform support
- Works with any JDBC compliant source
- Provides an intuitive graphical user interface for submitting SQL queries and dealing with database objects

You do not have to be an expert in SQL to use WinSQL; however, knowledge of SQL is helpful. You can view database objects such as tables, views, indexes, stored procedures, triggers and more – all though the provided catalog tab.

Use WinSQL to

- Submit SQL queries to any JDBC compliant back-end data source
- Graphically view database objects

- Compare schemas of two databases
- Manage your most common database tasks from one central application
- No need to buy separate products for different databases – WinSQL for JDBC works with a number of databases including Oracle, Sybase, DB2, Informix, MS SQL Server, Pervasive SQL, MS Access, MySQL, PostgreSQL, and others.

System Requirements

- JDBC 2.0 Compliant JDBC Driver
- 10 MB Hard Disk
- 128 MB Ram, 256 MB recommended
- Any operating systems that supports Java, including MS Windows, Sun Solaris, Linux, Apple Mac, and other flavors of UNIX.

Installation and Setup

There are two editions of WinSQL: Lite and Professional and runs on multiple platforms. The following tables list the name of the installation file that is appropriate for each individual platform.

WinSQL Professional

Table 1

File Name	Platform	Description
WinSQLProLinuxNoVM.bin	Linux	Installer for Linux with JRE. You are required to install JRE 1.4.1_01

		or above on the client machine.
WinSQLProLinuxVM.bin	Linux	Installer for Linux and includes the necessary JRE.
WinSQLProWindowsNoVM.zip	Windows	Installer for Microsoft Windows. Does not include JRE. JRE 1.4.1_01 is required. This is a zipped file that holds install.exe
WinSQLProWinSQLVM.zip	Windows	Installer for Microsoft Windows, which includes appropriate JRE. This is a zipped file that holds install.exe
WinSQLProSolorisNoVM.bin	Solaris	Installer for Sun Solaris OS. It does not include JRE. Version 1.4.1_01 is required.
WinSQLProOtherNoVM.jar	Other	Installer for other platforms on which Java is supported

WinSQL Lite

Table 2

File Name	Platform	Description
WinSQLLiteLinuxNoVM.bin	Linux	Installer for Linux with JRE. You are required to install JRE 1.4.1_01 or above on the client machine.
WinSQLLiteLinuxVM.bin	Linux	Installer for Linux and includes the necessary JRE.
WinSQLLiteWindowsNoVM.zip	Windows	Installer for Microsoft Windows. Does not include JRE. JRE 1.4.1_01

		is required. This is a zipped file that holds install.exe
WinSQLLiteWinSQLVM.zip	Windows	Installer for Microsoft Windows, which includes appropriate JRE. This is a zipped file that holds install.exe
WinSQLLiteSolorisNoVM.bin	Solaris	Installer for Sun Solaris OS. It does not include JRE. Version 1.4.1_01 is required.
WinSQLLiteOtherNoVM.jar	Other	Installer for other platforms on which Java is supported

Installation Instructions

Table 3

Platform	Instructions
Windows	• After downloading, extract install.exe from the zipped file • Double click install.exe
Linux	• After downloading open a shell and, cd to the directory where you downloaded the installer • At the prompt type: sh ./install.bin
Solaris	• After downloading open a shell and, cd to the directory where you downloaded the installer • At the prompt type: sh ./install.bin

Other Platforms	• After downloading, type: `java -jar  WinSQLProOtherNoVM.jar`

Memory Setting

By default every Java program is set to use a maximum of 64 MB memory. However, this value can be changed by passing a –Xmx parameter as command line argument to the JVM. The installer routine for WinSQL changes this value to 512 MB, which means that WinSQL for JDBC can use up to 512 MB of memory. If you run a query that returns many records (over 100,000), it is possible that you might have to increase this limit. Edit the WinSQLPro4JDBC.lax file to increase this limit. Search for

```
lax.nl.java.option.additional=-Xmx512m
```

and change –Xmx512m to a higher value, such as –Xmx1024m to increase this limit to 1 GB. The maximum memory a Java process can use is 2GB on Windows/Linux/X86 and 4GB on Solaris.

Chapter 2

Getting Started

When you start WinSQL for JDBC a login screen is displayed. This screen displays your previously created connection profiles or if no profile exists displays the JDBC Connection parameter screen.

JDBC CLASS COMBO BOX DISPLAYS DRIVER CLASSES FOUND IN THE CLASSPATH

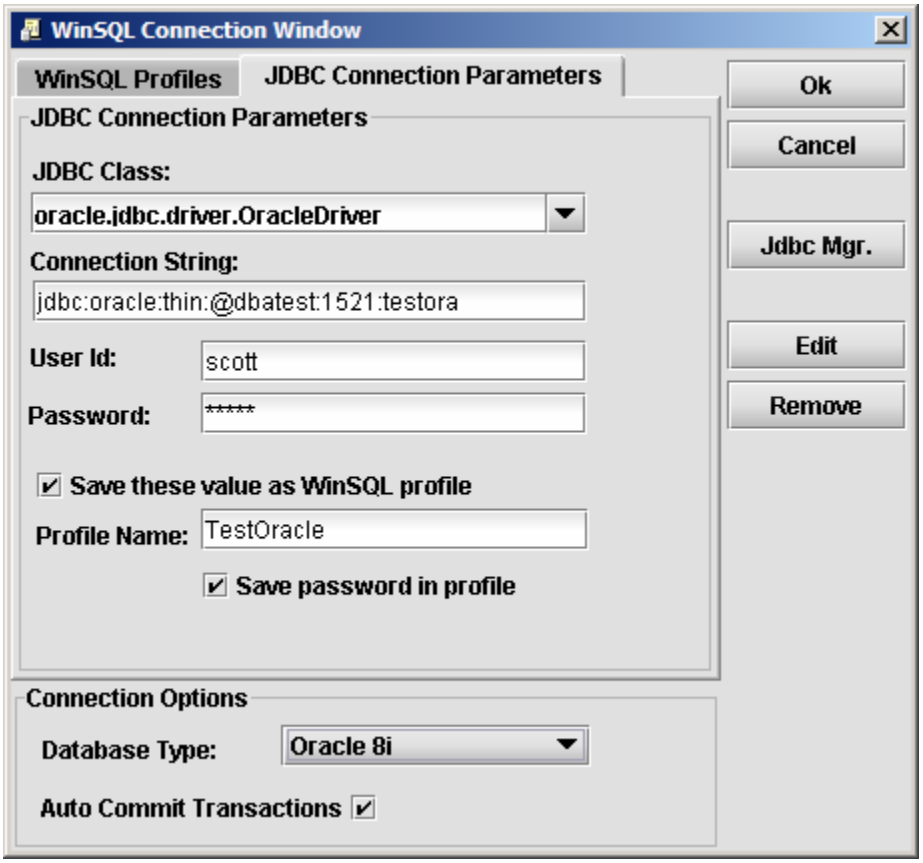

Figure 2-1

INVOKE JDBC MANAGER TO MODIFY CLASSPATH

The first time you open WinSQL for JDBC you will only see "sun.jdbc.odbc.JdbcOdbcDriver" as the JDBC class in the JDBC Class combo box. This combo box displays the name of Drivers that are found in CLASSPATH.

Modifying CLASSPATH for JDBC drivers

You need JDBC drivers to connect to any database. However, every JDBC driver that you wish to use must be added in the CLASSPATH of the application. This is done by JDBC Manager bundled with WinSQL. Click the "JDBC Mgr." button to invoke the JDBC manager.

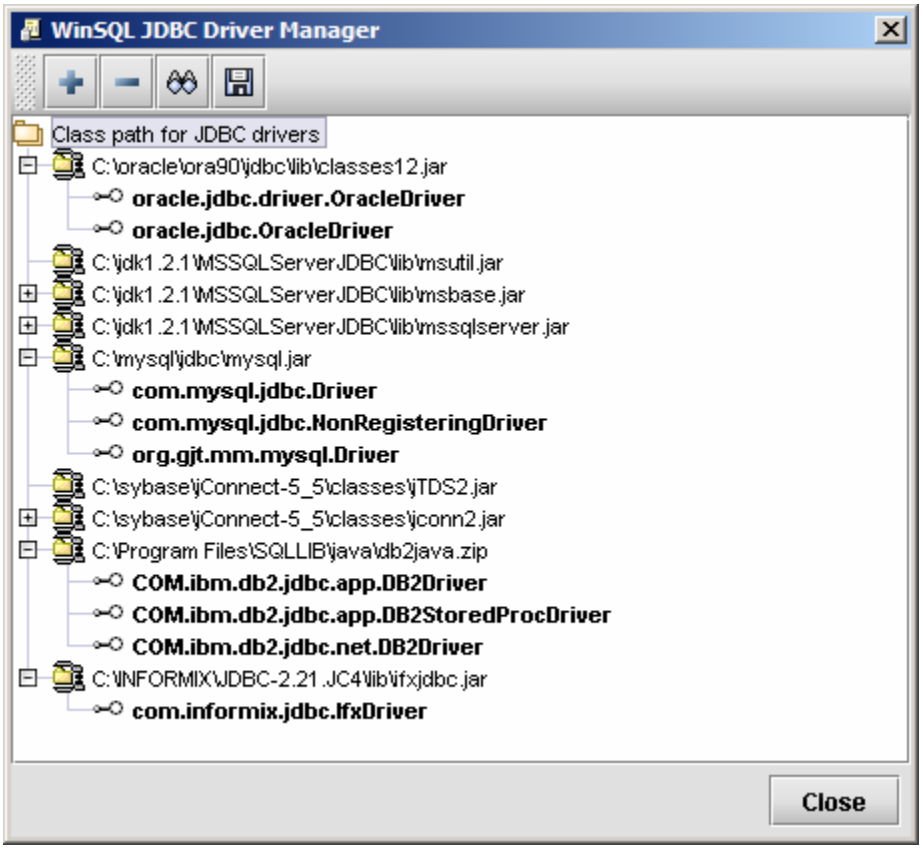

Figure 2-2

All JAR/ZIP files that appear in JDBC Driver Manager get included in the CLASSPATH for WinSQL for JDBC.

Table 4

Buttons	Description

	Click to add a new JAR/ZIP file in the CLASSPATH
	Remove the selected JAR/ZIP file from CLASSPATH
	Click this button to parse the JAR/ZIP file and extract the name of the Driver class. Note that this parsing is done automatically when a JAR file is added and therefore, it is not required that you click this button
	Saves configuration

After you close the JDBC Driver Manager window, the JDBC Driver combo box gets filled with available drivers.

JDBC Driver files for popular data sources

The following table lists name of JAR files for some of the database that you can use with WinSQL for JDBC. For a complete list of drivers visit

http://servlet.java.sun.com/products/jdbc/drivers

THE NAME OF THE JAR FILES MAY CHANGE IN THE FUTURE. PLEASE REFER TO YOUR BACK-END DATABASE DOCUMENTATION FOR THE EXACT NAME

Table 5

Back-end Database	Required JAR files	Source
Oracle	classes12.jar	Found on Oracle CD or can be downloaded from http://otn.oracle.com/software/content.html
IBM DB2	db2java.zip	Found on DB2 CD or can be downloaded from http://ibm.com/software/data/db2
Informix	ifxjdbc.jar	Found on Informix CD or can be downloaded from http://ibm.com/software/data/db2
Sybase	jTDS2.jar jconn2.jar	Found on Sybase CD or can be downloaded from http://www.sybase.com/downloads.

		Search for jConnect for JDBC.
SQL Server	msutil.jar msbase.jar mssqlserver.jar	Can be downloaded from http://www.microsoft.com/sql/
MySQL	mysql.jar	Can be downloaded from MySQL's web site. http://www.mysql.com/products/connector-j/index.html
PostgreSQL	pg73jdbc1.jar	Can be downloaded from http://jdbc.postgresql.org/download.html. Note that "73" in the name of the JAR file refers to the version number, which may change.

Database connection parameter

Once the CLASSPATH is set, you are ready to connect to a database.

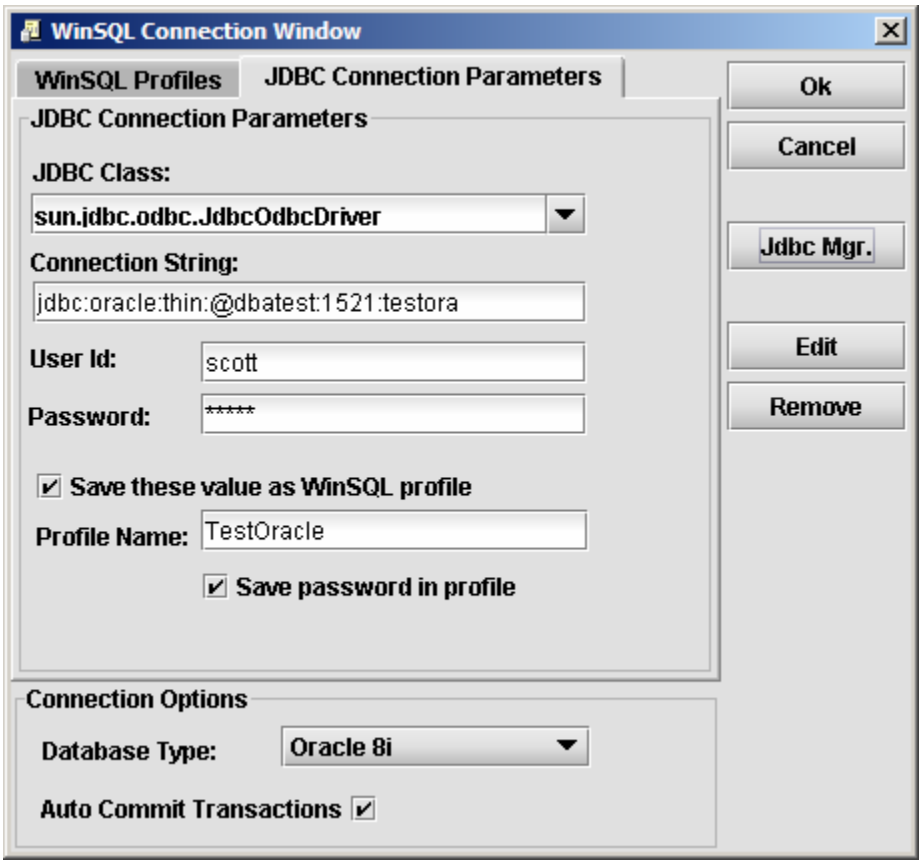

Figure 2-3

Table 6

Field Name	Description
JDBC Class	This is the Java Class name for the JDBC driver. You will only see Classes that are found in the CLASSPATH
Connection string	Connection string required to connect to a database. The exact value depends on the database you are connecting to. Please refer to your database documentation for exact format. For your convenience we have provided the connection string for some database in the table below
User Id	User name required for authentication to the back-end data source. If you don't know what to type here consult your database administrator.
Password	User password for database authentication. If you don't know

	what to type here consult your database administrator.
Profile name	You can optionally save all these setting as a WinSQL profile. Next time you connect simply select appropriate profile. This field holds the name of the profile
Database type	This option is available only in WinSQL Professional and refers to a database plug-in for WinSQL.
Auto commit transaction	If checked all transactions are automatically committed. If you choose to uncheck this option, you must manually COMMIT or rollback all DML queries.

Connection URL

Following table lists type connection string for some databases. Please note that the values may change in the future and therefore, it is important that you refer to your back-end database documentation for an updated format.

Table 7

Database	Connection URL String
IBM DB2	jdbc:db2://<hostname>:<port>/<database name>
Oracle	**Type 4 – Thin Driver** jdbc:oracle:thin:@<hostname>:<port>:<OracleSID> **Type 2 – OCI Driver** jdbc:oracle:oci8:@<hostname>:<port>:<OracleSID> jdbc:oracle:oci7:@<hostname>:<port>:<OracleSID>
Sybase	jdbc:sybase:Tds:<hostname>:<port>
SQL Server	jdbc:microsoft:sqlserver://<hostname>:<port>;DATABASENAME=<DBName>;
Informix	jdbc:informix-sqli://<host>:<port>/<DBName>:informixserver=<Informix Instance Name>

Main Menu

Following table explains the main menu in WinSQL for JDBC

Table 8

Top Level	Sub Level	Description
File	New Connection	Creates a new connection to a data source. WinSQL will prompt you for connection information
	New Query Page	Creates a new Query Set for the existing connection
	Open File in Existing connection	Opens an existing file containing SQL scripts
	Open File in New Connection	Open an existing file containing SQL scripts in a new connection
	Save	Save SQL Script
	Save As	Save SQL Script to a different file
	Disconnect	Closes the connection with the database. Next time you run query, you will be prompted to establish connection again
	Close	Closes the database connection and window
	Exit	Closes all connections, windows and terminates the application
Edit	Undo	Undo previous changes
	Cut	Cut selected text to the clipboard
	Copy	Copies selected text to the clipboard
	Paste	Paste data in the clipboard to the query window
	Select All	Selects entire scripts in the query tab

	Comment block	Puts a comment around the selected text
	Uncomment block	Removes previously commented code
	Indent Code	Indents code to the left
	Unindent Code	Un-indents previously indented code
	Find	Displays the find dialog box
	Find again	Searches again
	Replace	Replaces the data
	Option	Displays configuration dialog for WinSQL for JDBC
View	Query	Switches the view to Query tab
	Results	Switches the view to Results tab
	Catalog	Switches the view to Catalog tab
	Refresh Catalog	Refreshes the catalog
Query	Execute	Executes a query
	Cancel Execution	Stops the currently running query
	Display Results in Text	When selected, query results are displayed in a text box
	Display Results in Grid	When selected, query results are displayed in a grid box. This is the default value.
Tools	Database Diff	Invokes the Database Diff wizard. Only available in the Professional version
	DB Architect	Invokes DB Architect
Help	Register WinSQL	Accepts registration information for WinSQL
	About	Displays version and system information about WinSQL

Menu Shortcuts

Most of the menu items in WinSQL is assigned a shortcut. The mapping of these shortcuts can be dynamically changed by modifying the KeystokesShortcuts.dat file in the config directory. A sample of the file is shown below. The left hand side of the equal sign must correspond to the text that appears on the menu item in WinSQL. A space sign is represented by an underscore.

Example:

```
New_Connection=alt control N
```

This means that ALT-CNTRL N becomes a shortcut for "New Connection" menu item.

```
New_Connection=alt control N
New_Query_Page=control N
Open File In New Connection=alt control O
Open File In Existing Connection=control O
Save=control S
Undo=control Z
Copy=control C
Cut=control X
Paste=control V
Select All=control A
Comment Block=control B
Uncomment Block=shift control B
Indent_Code=shift control I
Unindent_Code=shift control U
Find=control F
Find Again=F3
Replace=control R
Query=alt Q
Results=alt R
Catalog=alt C
Execute=alt X
Refresh_Catalog=F5
```

Running Queries

Once a connection is established, you can write one or more SQL statements and click the play button to submit them.

Important: WinSQL for JDBC does not change the script of the query, which gets submitted as-is to the back-end server. Therefore, any error messages that you get are generated by the RDBMS server and passed through the JDBC driver to WinSQL.

Query Terminator

Query terminator is a configurable string that can be used to separate more than one query. The default value for a query terminator is "go", which can be changed through the configuration window (click "Edit→Options" to access the configuration window)

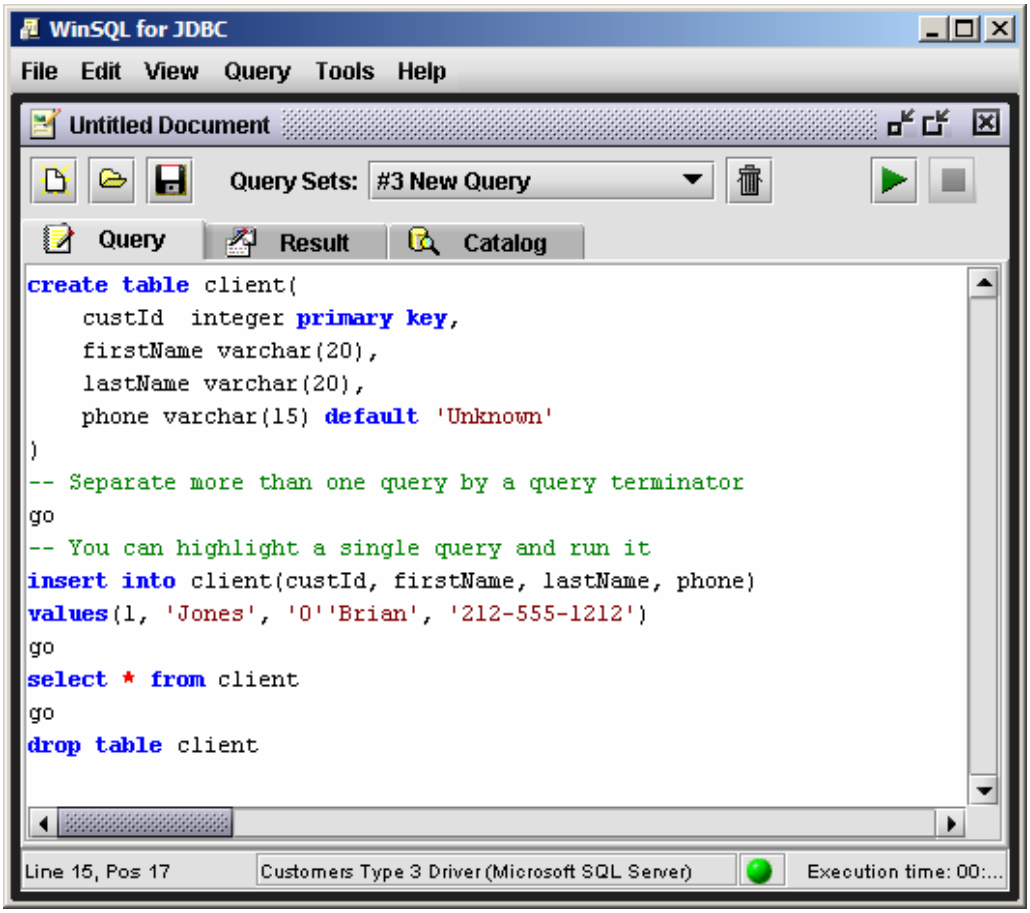

Figure 2-4

Working with Query Sets

A query set is one page of SQL scripts in WinSQL. When working with large scripts it is helpful to arrange queries in multiple pages, also known as Query Sets. You can open as many Query Sets as you want. The number is limited only by available memory on the machine. Click the "New Query Page" button on the screen to create a new Query Set. To switch between different Query Sets, select the appropriate set from the Query Sets combo box. Besides switching the Query Tab, the results will also change.

Displaying results in Grid or Text

WinSQL for JDBC allows you to display results in two format:

- Grid control
- Text control

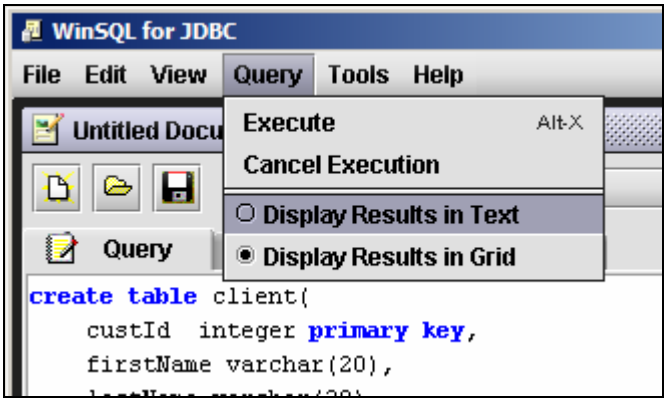

Figure 2-5

Limitations: Maximum characters per field in Text mode is set to 512 characters. If a field has more characters, they will get truncated. This limit does not exist when results are displayed in the Grid

Configuration options

The configuration screen is invoked by selecting the Options menu item under main Edit menu. This screen is provided to set different configuration that a user can change. Details for each option are provided below.

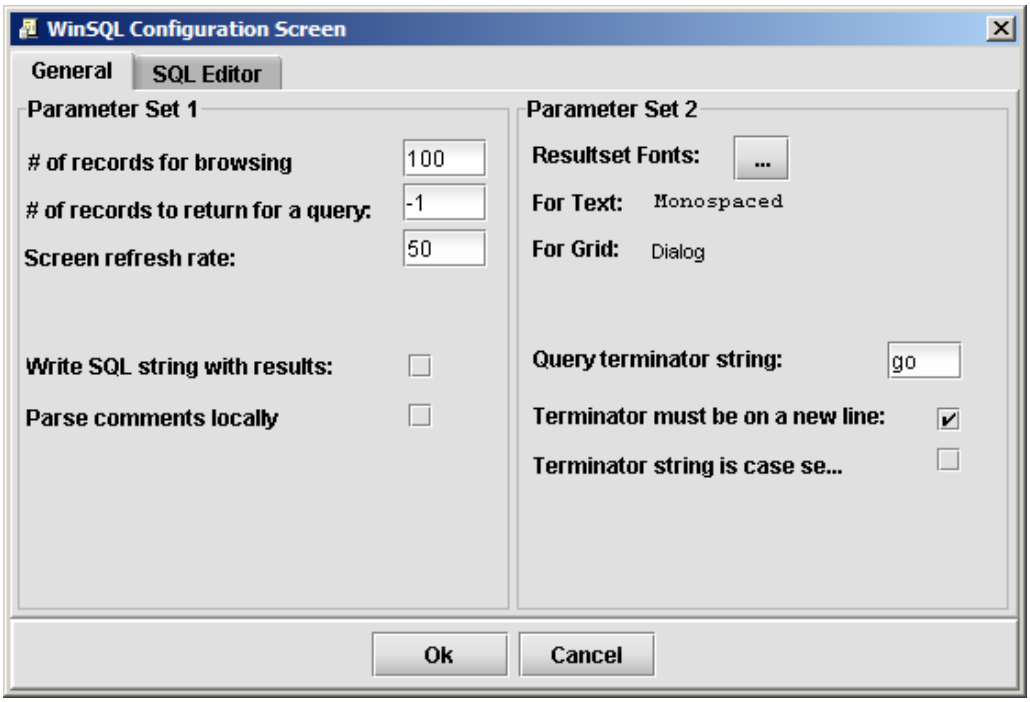

Figure 2-6

of records for browsing

This is a numeric value corresponding to the number of records to display in the Browse Data feature available in the catalog window. The default value is 100.

Caution: If you set this value to a large number, WinSQL will fetch a large result set whenever the Browse Data item is clicked. This can potentially slow down not only your machine but can also have negative affects on your network. We suggest that you apply filter on the records if you rather than displaying all the records in a table.

of records to return from a query

This parameter limits the number of records that are returned from a SELECT statement. A value of -1, which is the default, signifies no limit.

Screen refresh rate

Number of records to fetch before screen is painted. This option is applicable when results are displayed in Text box.

Write SQL string with result set

If checked, the SQL Statement is printed right before the result set. This is useful if you are displaying the result in the Text control.

Parse comments locally

If checked, WinSQL will parse the query script and strip all the comments before submitting it to the backend database.

Query terminator string

This string breaks the script in multiple parts and each part is sent separately to the database. The default value for this parameter is 'GO' and it works in conjunction with "Terminator must be on a new line" and "Terminator string is case sensitive" parameters.

Terminator must be on a new line

If checked, a terminator string is only treated as a terminator if it appears on a new line.

Terminator string is case sensitive

If checked, the terminator string becomes case sensitive.

SQL Editor

This tab is used to specify a different coloring scheme for Syntax Highlighting.

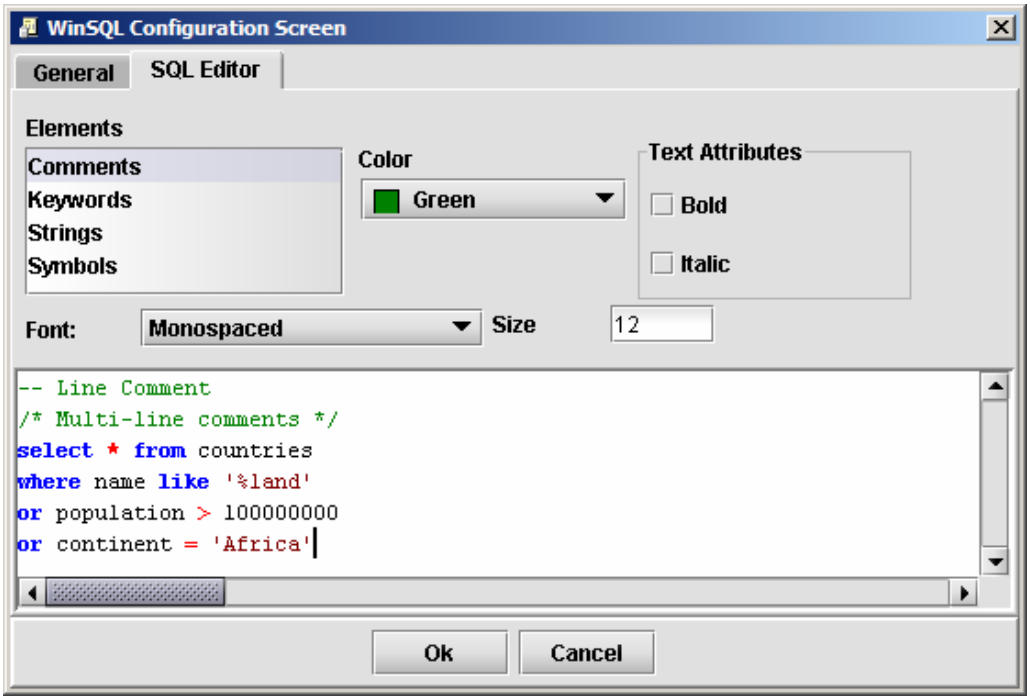

Figure 2-7

Viewing Data Catalog

Viewing database catalog can be very handy when writing SQL statements. WinSQL displays this information in a hierarchical format.

Important: Catalog information depends heavily on the capability JDBC driver. WinSQL queries these drivers and finds out what information is available.

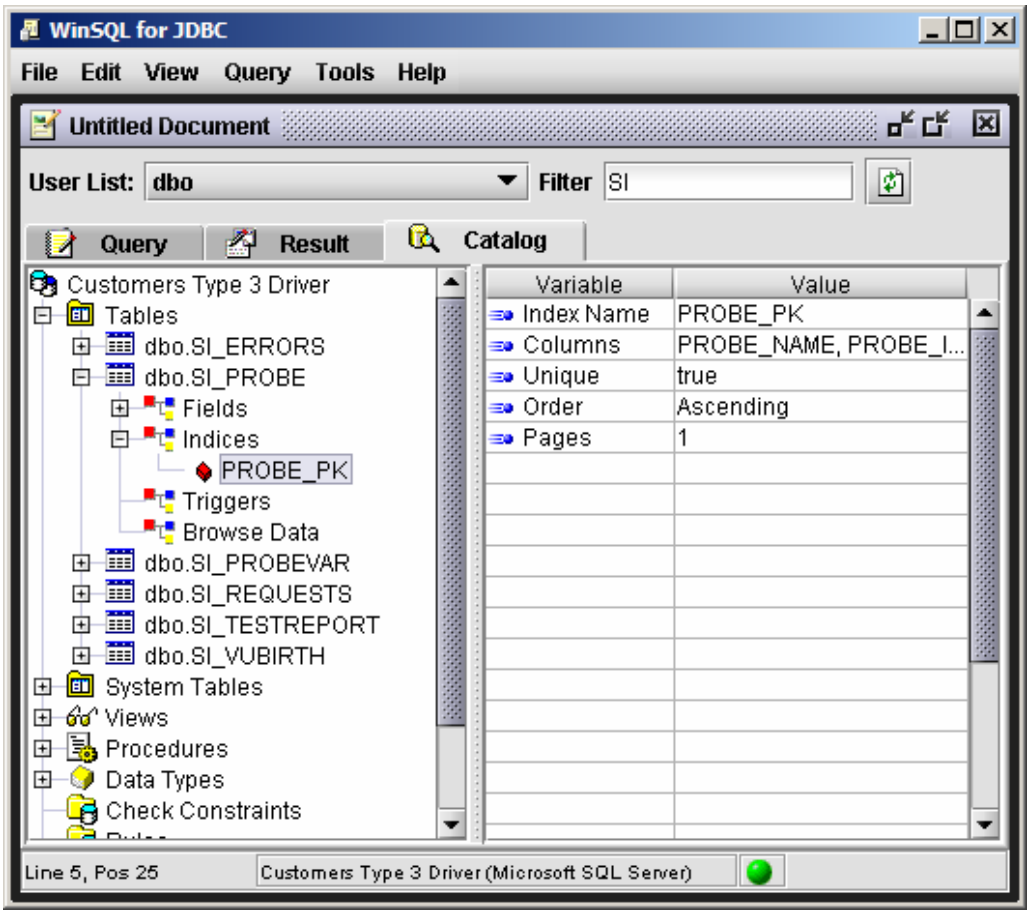

Figure 2-8

The catalog screen displays the following information

TIP: DOUBLE CLICK THE HEADER COLUMN IN BROWSE DATA TO SORT

Tables / System tables

→ Fields

→ Index

→ Browse data – Ability to browse records in the table.

→ Triggers (Professional only and if a database plug-in is available)

Views

→ Fields

→ Browse data – Ability to browse records in the view.

→ View Scripts (Professional only and if a database plug-in is available)

Stored Procedures

→ Fields

→ SQL Scripts (Professional only and if a database plug-in is available)

Providing filters

In some situations where the number of tables is very high, it may take a long time to display all the tables and their fields. Catalog window provides two filtering parameters that can restrict the total number of objects displayed. These filters are:

- **By user name** – Limits the catalog to display objects owned by the specified user or schema
- **By object name** – Limits the catalog to display objects where the name starts with the specified value

Chapter 3

Database Diff

Database Diff feature in WinSQL Professional provides you a mechanism to compare schemas of two similar databases. For instance you can use WinSQL to compare the schema of a database used in a development environment with a similar database in production environment. This feature is available in WinSQL Professional only.

To start the comparison, select "Database Diff…" from the Tools menu. The following screen will show up.

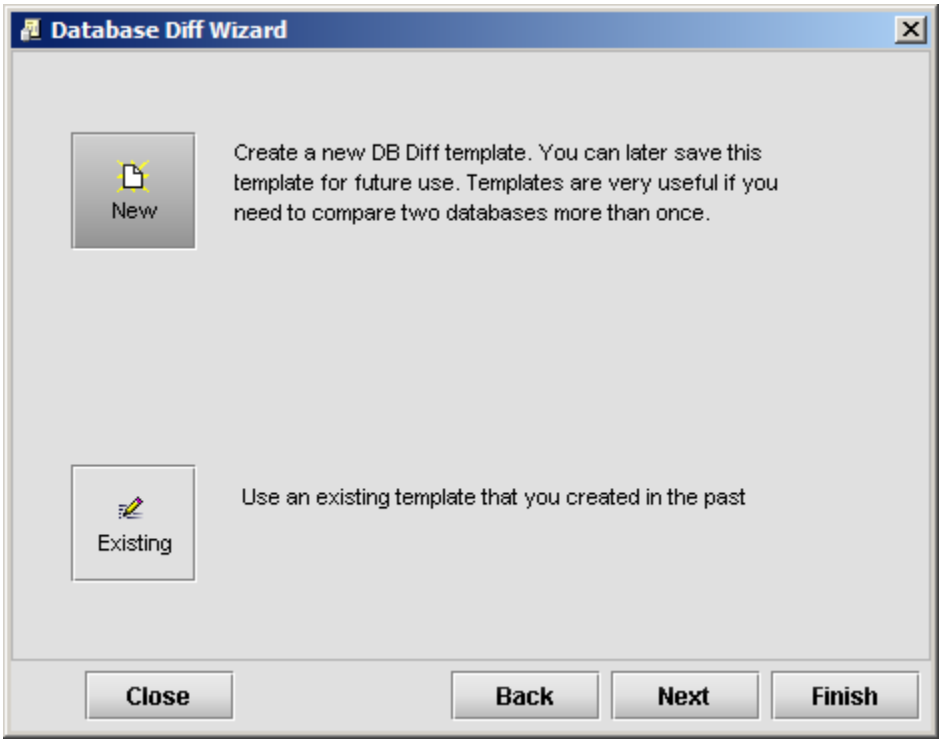

Figure 3-1

If this is the first time you are invoking the wizard select "New" template button. If you have a previously saved template available, you may select the "Existing" button.

→ Click next to proceed.

Here you need to establish connections with the two databases that you are trying to compare. Once the connection is established, the connection button will read "Disconnect" Refer to the screen shot below.

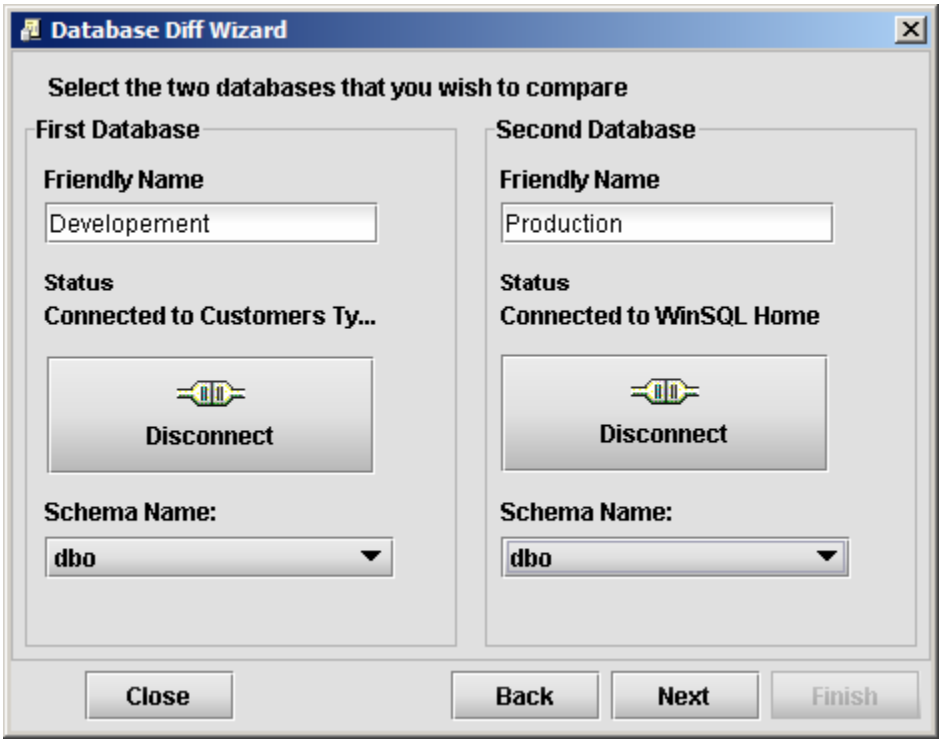

Figure 3-2

→ Click next to proceed

Understanding how objects are compared

When comparing two databases, WinSQL picks the first database and treats it as a pivot database, meaning it will examine all the objects in this database and will try to match it will the non-pivot database.

If "Run comparison with respect to both databases" is selected on the third step, WinSQL will run the comparison twice – once for each database as pivot.

Primary Objects and Detail Options

Important: You must select an appropriate database plug-in to fully utilize the Database Diff functionality

This list box displays all the objects that can be compared. On the right hand side you see individual options for the object selected on the left. For instance when you select Stored Procedures on the left, you see "Parameters", "Parameter type", "Procedure script", "Show DDL if doesn't match" on the right.

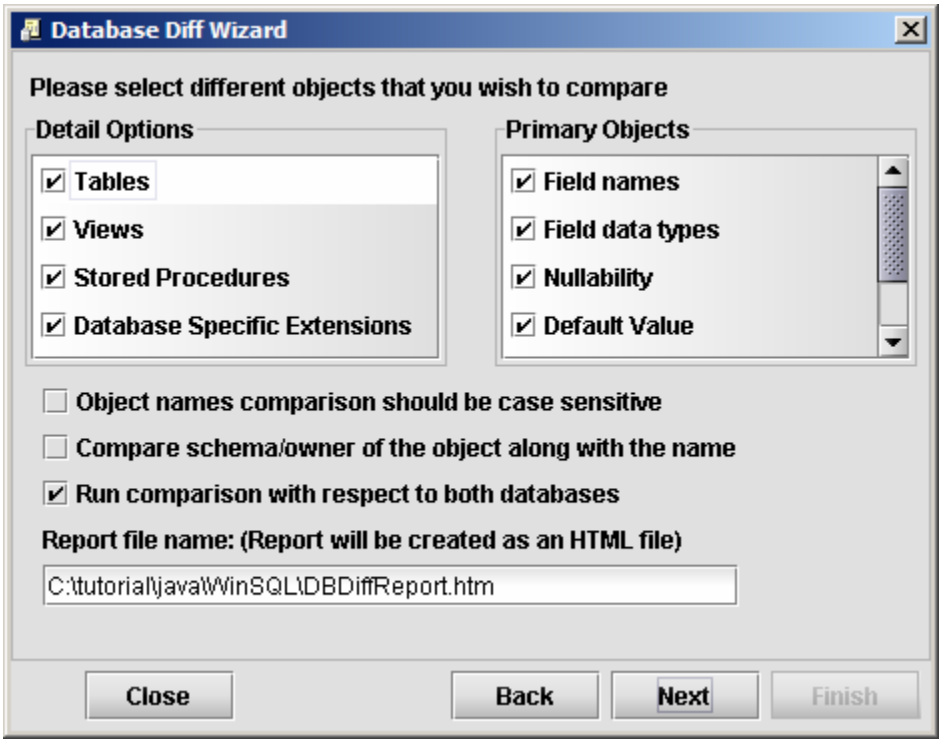

Figure 3-3

Generated Report

The output report is generated as an HTML file. The next steps accepts input parameter that modifies the look and feel of the generated report

Style Sheet

The output report uses a Style sheet for format the HTML. You can either use the built-in style sheet or provide one through an external file. One sample style sheet is provided with WinSQL and gets installed during setup

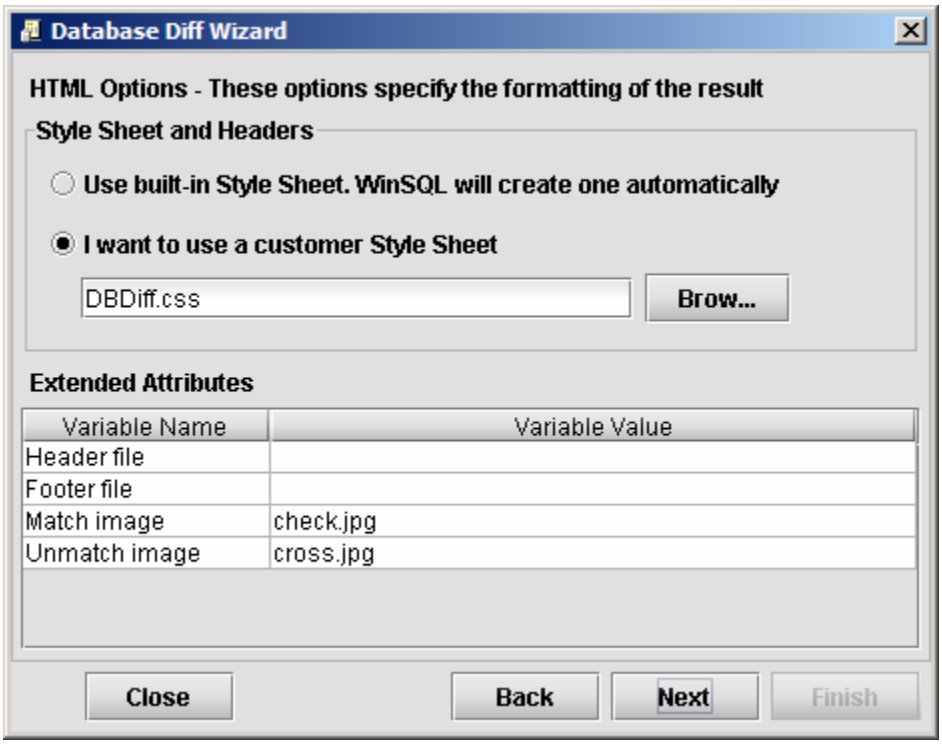

Figure 3-4

Header/Footer file – If this option is specified WinSQL will use the text contained in the specified file as header and/or footer for the report. This is a convenient way to customize the look and feel of the generated report

Match/Unmatch image – This is the URL for the images in the report

→Click next to proceed to the summary page.

The summary page displays a summary of all the options you specified in the previous step. If you are satisfied will all the settings, click next to proceed to the next step.

Figure 3-5

Saving template

Like many other wizard in WinSQL, you have an option of saving all these steps as a template. This is a convenient way of running similar tasks again at a later time.

Running Report

Once satisfied with all the options, click the "Finish" button to view the report

Chapter 4

Database Specific Plug-ins

Plug-in enhances the capability of WinSQL. JDBC is a generic API and therefore, does not support database specific functions. Most of this functionality is achieved by querying system tables (also known as system catalog) in the back-end database. Plug-in provides the script for these queries, which are in plain text and in XML format.

XML is becoming the de facto standard for configuration files and to store small documents. Many programmers and system administrators are already familiar with its semantics and syntax. Therefore, WinSQL uses XML to create plug-ins, which makes its editing very easy.

XML file format

The following shows a complete example of a plug-in for Oracle 8i

```
<?xml version="1.0" encoding="iso-8859-1" ?>


<WinSQLDatabaseType dbName="Oracle 8" dbAlias="Oracle 8i"
                    version="3.8">
  <view supported="yes">
    <viewDDL supported="yes">
      <query>
        <sqlString>
        <![CDATA[
        select text from ALL_VIEWS
        where OWNER = ? and VIEW_NAME = ?
         ]]>
        </sqlString>
        <paramList>
          <param position="1" datatype="VARCHAR2"
                 value="WinSQL_SCHEMA" tag="1"/>
          <param position="2" datatype="VARCHAR2"
                 value="WinSQL_OBJNAME" tag="1"/>
        </paramList>
      </query>
    </viewDDL>
  </view>
  <!-- The storedProcedure element tells WinSQL how to process a
      Stored procedure -->
  <storedProcedures supported="yes">
    <storedProcDDL supported="yes">
      <query>
        <sqlString><![CDATA[ select  TEXT From SYS.ALL_SOURCE
        Where TYPE = 'PROCEDURE' And OWNER = ? And NAME = ?
        order by line
         ]]></sqlString>
```

```
      <paramList>
        <param position="1" datatype="VARCHAR2"
               value="WinSQL_SCHEMA" tag="1"/>
        <param position="2" datatype="VARCHAR2"
               value="WinSQL_OBJNAME" tag="1"/>
      </paramList>
    </query>
  </storedProcDDL>
</storedProcedures>

<triggers supported="yes">

  <triggerDDL supported="yes">
    <query>
      <sqlString><![CDATA[
      select TRIGGER_NAME, 'E', 'CREATE TRIGGER ' || DESCRIPTION,
             TRIGGER_BODY
      from ALL_TRIGGERS
      where OWNER = ? and TABLE_NAME = ?

       ]]></sqlString>
      <paramList>
        <param position="1" datatype="VARCHAR2"
               value="WinSQL_SCHEMA"  tag="1"/>
        <param position="2" datatype="VARCHAR2"
               value="WinSQL_OBJNAME"  tag="1"/>
      </paramList>
    </query>
  </triggerDDL>
</triggers>

<extendedNodes>
  <extension level="1" name="Sequences" displayType="WinSQL_TreeView"
             hasChildren="yes" useInDBDiff="no">
    <query>
      <sqlString><![CDATA[
      select SEQUENCE_OWNER || '.' || SEQUENCE_NAME,
             sequence_name, sequence_owner
      from SYS.ALL_SEQUENCES where SEQUENCE_OWNER like ?
      order by SEQUENCE_OWNER, SEQUENCE_NAME
       ]]></sqlString>
      <!-- Level 1 extensions can ONLY have WinSQL_SCHEMA as a
           parameter  -->
      <paramList>
        <param position="1" datatype="VARCHAR2"
               value="WinSQL_SCHEMA"  tag="1"/>
      </paramList>
    </query>
    <extension level="2" name="Sequence Detail"
               displayType="WinSQL_ListView"
               hasChildren="no" useInDBDiff="no">
      <query>
        <sqlString><![CDATA[
        select * from ALL_SEQUENCES
        where SEQUENCE_OWNER = ? and SEQUENCE_NAME = ?
         ]]></sqlString>
        <paramList>
          <param position="1" datatype="VARCHAR2"
                 value="WinSQL_PARENTVALUE"  tag="3"/>
          <param position="1" datatype="VARCHAR2"
                 value="WinSQL_PARENTVALUE"  tag="2"/>
        </paramList>
      </query>
    </extension>
  </extension>

  <extension level="1" name="Last Error" displayType="WinSQL_TreeView"
```

```
                hasChildren="yes" useInDBDiff="no">
      <query>
        <sqlString><![CDATA[
        select distinct NAME from SYS.ALL_ERRORS
         ]]></sqlString>
        <!-- Level 1 extensions can ONLY have WinSQL_SCHEMA as a
              parameter  -->
        <paramList>  </paramList>
      </query>
      <extension level="2" name="Error Detail"
                 displayType="WinSQL_GridView"
                 hasChildren="no" useInDBDiff="no">
        <query>
          <sqlString><![CDATA[
          select * from ALL_ERRORS where NAME = ? order by SEQUENCE
           ]]></sqlString>
          <paramList>
            <param position="1" datatype="VARCHAR2"
                   value="WinSQL_PARENTVALUE"  tag="1"/>
          </paramList>
        </query>
      </extension>
    </extension>
  </extendedNodes>

</WinSQLDatabaseType>
```

Plug-in Elements

Element name: <view>

This node gets the SQL query that was used to generate the view.

Children node(s) - viewDDL

Attribute name	Description
supported	Can be “yes” or “no. If views are supported, this value must be “yes”. The value is case sensitive.

Element name: <viewDDL>

Children node(s) – query

Attribute name	Description
supported	Can be “yes” or “no. If querying a system table can return the script for a view, this value should be “yes”. The value is case sensitive.

Element name: <storedProcedure>

Children node(s) - storedProcDDL

Attribute name	Description
supported	Can be “yes” or “no. If stored procedures are supported, this value must be “yes”. The value is case sensitive.

Element name: <storedProcDDL>

Children node(s) - query

Attribute name	Description
supported	Can be “yes” or “no. If querying a system table can return the script for a stored procedure, this value should be “yes”. The value is case sensitive.

Element name: <triggers>

Children node(s) - triggerDDL

Attribute name	Description
supported	Can be “yes” or “no. If triggers are supported, this value must be “yes”. The value is case sensitive.

Element name: <triggerDDL>

This node holds the query node required to get the information regarding the triggers on a database.

Children node(s) - query

Attribute name	Description	Children nodes
supported	Can be “yes” or “no. If querying a system table can return the script for a trigger, this value should be “yes”. The value is case sensitive.	storedProcDDL

Element name: <query>

This element is used in many places within the document. It is used to specify an SQL statement that is run against the database to fetch any data. Below is a snippet from the XML document. This snippet contains <query>, <sqlString> and <paramList> nodes

```
<query>
        <sqlString>
        <![CDATA[
        select text from ALL_VIEWS
        where OWNER = ? and VIEW_NAME = ?
         ]]>
        </sqlString>
        <paramList>
          <param position="1" datatype="VARCHAR2"
                 value="WinSQL_SCHEMA" tag="1"/>
          <param position="2" datatype="VARCHAR2"
                 value="WinSQL_OBJNAME" tag="1"/>
        </paramList>
      </query>
```

Children node(s) - sqlString, paramList

Attribute name	Description
none	

Element name: <sqlString>

Holds the SQL query within a CDATA section. Run time parameters are specified by a ?.

SQL string for views and stored procedures
This query should only return 1 column. If more than one column is returned, the remaining columns will be ignored. Concatenating all the rows will create the value of the script.

SQL string for triggers
In case of triggers, this query should return 3 columns. The first columns should be the name of the trigger. If names are not supported, query must return 'Not Available'. The second column is the type of trigger. It could be Update, Delete or Insert. The third column is the text containing the SQL script

SQL string for extension
There are two kinds of extensions. Level 1 and level 2. In case of level 1, there can be only one parameter, which is the schema or the owner of the object. The number of columns returned by the query depends on the display type of the extension. Refer to displayType node for further details.

Attribute name	Description
none	

Element name: <paramList>

Holds the param node

Children node(s) - param

Attribute name	Description
None	

Element name: <param>

Holds information regarding the parameter of the query.

Attribute name	Description
position	This is a numeric value that represents the parameter number, which starts from 1
dataType	This is the data type of the parameter. It MUST be a valid datatype of the backend database. You can get a list of valid datatypes from the catalog tab of WinSQL. For example if you are changing the plug-in for MS SQL Server and one of the datatype is a CHAR, writing CHARACTER instead of CHAR will generate errors.
value	This signifies the meaning of the parameter. It tells WinSQL where to get the value at run time. Possible entries are: **WinSQL_SCHEMA** – This is owner of the object **WinSQL_OBJNAME** – This is the name of the object **WinSQL_FULLOBJNAME** – This is the schema followed by a period and the name of the object **WinSQL_PARENTVALUE** – This value is only useful in the extension node where level is 2. This is the result from a parent query. The tag attribute specifies the column number.
tag	This attribute is only useful in the extension node where level is 2. This specifies the column number of the parent query.

Extension nodes

Extension nodes provide the SQL statement to fetch database specific feature like sequences in Oracle. There are two kinds of extension nodes:

- Level 1
- Level 2

When a user clicks on the Tree View control on the left of the catalog tab, level 1 query gets executed. Level 2 provides a drill-down query for level 1.

Following code is an extract of the extension node from the example above.

```
<extension level="1" name="Last Error" displayType="WinSQL_TreeView"

          hasChildren="yes" useInDBDiff="no">
   <query>
     <sqlString><![CDATA[
     select distinct NAME from SYS.ALL_ERRORS
      ]]></sqlString>
     <!-- Level 1 extensions can ONLY have WinSQL SCHEMA as a parameter  -->
     <paramList>  </paramList>
   </query>
   <extension level="2" name="Error Detail"

       displayType="WinSQL_GridView" hasChildren="no" useInDBDiff="no">
     <query>
       <sqlString><![CDATA[
       select * from ALL_ERRORS where NAME = ? order by SEQUENCE
        ]]></sqlString>
       <paramList>
         <param position="1" datatype="VARCHAR2"

                value="WinSQL_PARENTVALUE"  tag="1"/>
       </paramList>
     </query>
   </extension>
 </extension>
```

First lets examine the attributes of the <extension> element.

Attribute name	Description
level	This is the level number of the extension. It must be either 1 or 2
name	This is only valid for level 1. It is the text that appears in the tree view control in the catalog tab
displayType	This is the target for the query result. There are four possible values for this attribute: **WinSQL_TreeView** – Use this to display results in tree view control, which appears on the left-hand side of the catalog window. Only the first the column is displayed in the window. However, the query can return more than one column. This is useful if you want to use data from the second column to appear in the WHERE clause of level 2 query. **WinSQL_GridView** – Use this to display results in a grid control, which appears on the right-hand side of the catalog window. The

	query can return as many rows and columns that are necessary. Field name will be displayed as column headers **WinSQL_ListView** – Use this to display results in a list view control, which appears on the right-hand side of the catalog window. The query can only return one row of data. If more rows are returned, their results will be discarded. **WinSQL_TextView** – Use this to display results in a text box. The query can only return 1 column but multiple rows. The result of every row is concatenated with the previous row. This is useful to display SQL queries.
hasChildren	Can be either “yes” or “no”. A yes means that there is a level 2 extension for this node
useInDBDiff	Can be either “yes” or “no”. If yes, it means that this extension will be used DBDiff report.