QAD Enterprise Applications

SQL Administration Guide

QAD Data Warehouse Designer

78-0918-3.4
Data Warehouse Designer Version 3.4
February 2011

BIPortal_AG_v34/bkp

QAD Inc.

100 Innovation Place

Santa Barbara, California 93108

Phone (805) 566-6000

http://www.qad.com

Contents

What's New?

The following table summarizes significant differences between this document and the previous version.

Date/Version	Description	Reference
September 2011/QAD BI 3.5	Rebranded for QAD BI 3.5	--

CHAPTER 1

Introduction

Sql Admin is a simple database SQL query tool. It provides a means of connecting to a database and issuing a SQL query and displaying the results. One of its main advantages is the ability to access multiple different database platforms. It has been used against Oracle, Informix, Ingres, Sql Server, Sybase and Progress databases. To achieve this database independence it makes use of ODBC data sources.

There are many and better query tools available for accessing specific databases, but Sql Admin provides a good option where multiple database platforms co-exist and provide inputs to a data warehouse environment.

Note: This tool is designed as a data warehouse assistant. It may not accurately reflect what is in the database and should not be used to handle production data. For example *all columns are truncated at 256 character width* for ease of display.

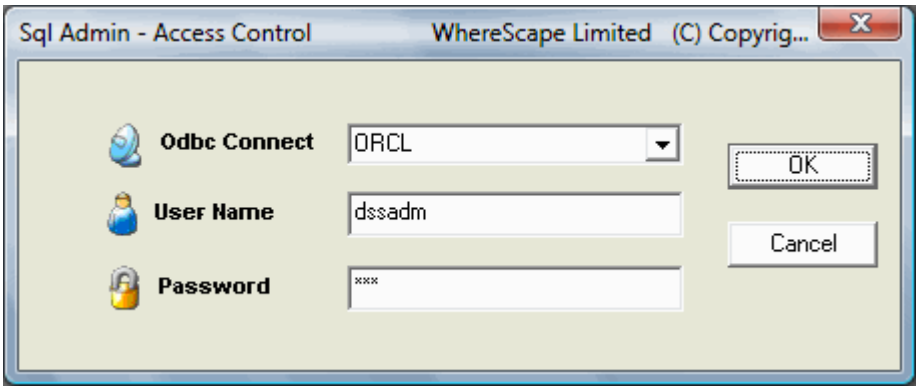
CHAPTER 2

Getting Connected

Sql Admin uses ODBC to connect to the database, so a valid ODBC data source must first be defined. Refer to the Installation and Administration Guide for assistance in the setting up of ODBC data sources.

Connecting to the database

Select the Sql Admin entry from the QAD program library in the standard Windows start/programs menu. The Sql Admin window will be displayed but with all toolbar buttons grayed out except for the 'New' window button. Either click this button or select the File/New menu option to initiate a database connection and logon. A dialog as follows will be displayed:



Select the desired ODBC source from the 'ODBC Connect' drop down list. Enter a valid database username and password for the specified connection. Once the form is completed click the OK button to logon to the database.

Connection failure

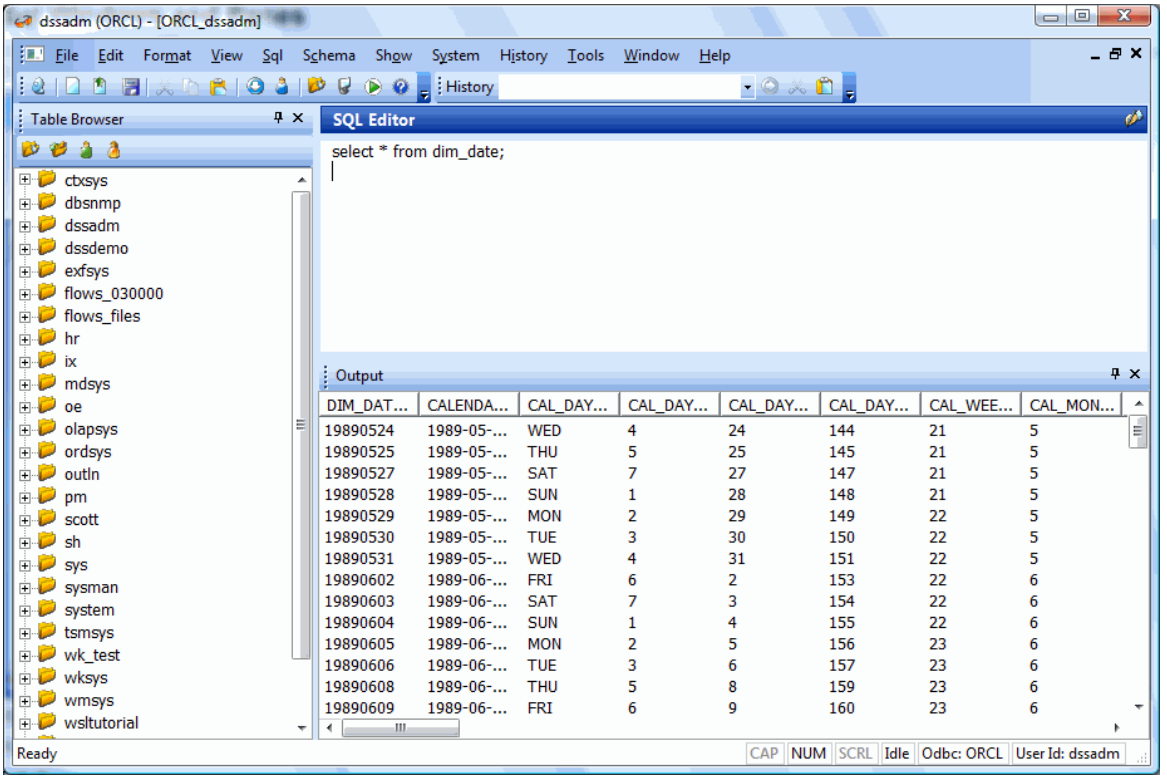
The connection and/or login can be rejected for any number of reasons including an invalid ODBC installation, an invalid database client, or simply an incorrect password. In any event an error is returned from either ODBC, the database client or the database providing the reason for the connection failure. Unfortunately, in some cases, these errors can be misleading or irrelevant. In such a case the best procedure is to work through the various components.

1. Try and establish a connection to the database using the database vendors tool set. (e.g. sql*plus for Oracle).
2. Check the network access to the database. (e.g. tnsping for Oracle).
3. If the database vendor has supplied tools for accessing the database through its client software, then test the connection with these tools.

CHAPTER 3

Sql Windows and Panes

Sql Admin can have multiple active connections to either the same or different databases. Each connection creates a new window. Although multiple windows can therefore exist, they all have the same format and panes as shown in the example below:



The database connection for the current window and the username are shown both at the top of the window and on the status line.

The three panes are described briefly below. A more detailed description can be found in the later parts of this manual.

Edit Pane

The edit pane is where a sql command may be entered for execution. Everything appearing in this window is executed when the execute button is pushed. If multiple sql statements exist then they will be executed sequentially from top to bottom. Each sql statement must be terminated by a semi-colon. In the example window above the sql statement 'select * from dim_date;' has been executed.

Table pane

The table pane provides a tree structure display of the users, tables and views etc. found in the database to which the window is connected. Pop-up menus are provided against most of the elements in this pane. Refer to the following chapters for more information. The tree structure is populated and refreshed by selecting the appropriate toolbar icon or one of the menu options provided.

Results pane

The results pane shows the output from either the sql statement in the edit window, a recalled statement or an action on one of the table tree entities. The results are broken into columns which have a maximum width of 256 characters. Care should be taken when displaying large result sets as these results are loaded into the memory of the PC, and this may take some time.

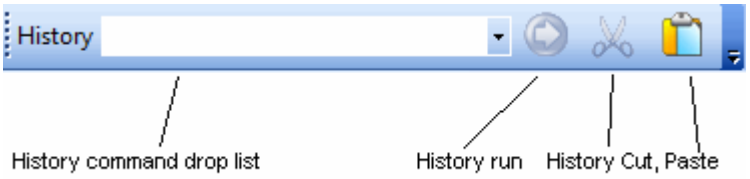
Toolbar

The main toolbar is shown and described below. A second history toolbar is described in the following section.

The new button will create a new connection and window. This connection can be to the same database and user as an existing connection if desired. The script create, open and save buttons all deal with the contents of the edit window. They allow the loading and saving of commands to disk. The procedure compile button will attempt to compile a procedure or function. Any errors will be displayed in the results window. Note, that the compile functionality only works for some databases. The show users button attempts to display the currently logged on database users and the last command they issued. This option only works for some databases and then only if the required privileges have been granted. The results may not be accurate given the dynamic nature of the environment and should be taken as an indication only. The refresh tree view button refreshes the objects in the table pane. This refresh is performed in a background thread. The toggle view button changes between the three pane view and the edit pane filling the whole window. The run button executes the commands in the edit pane.

History Toolbar

The history toolbar allows the selection and execution of commands that have been previously executed.

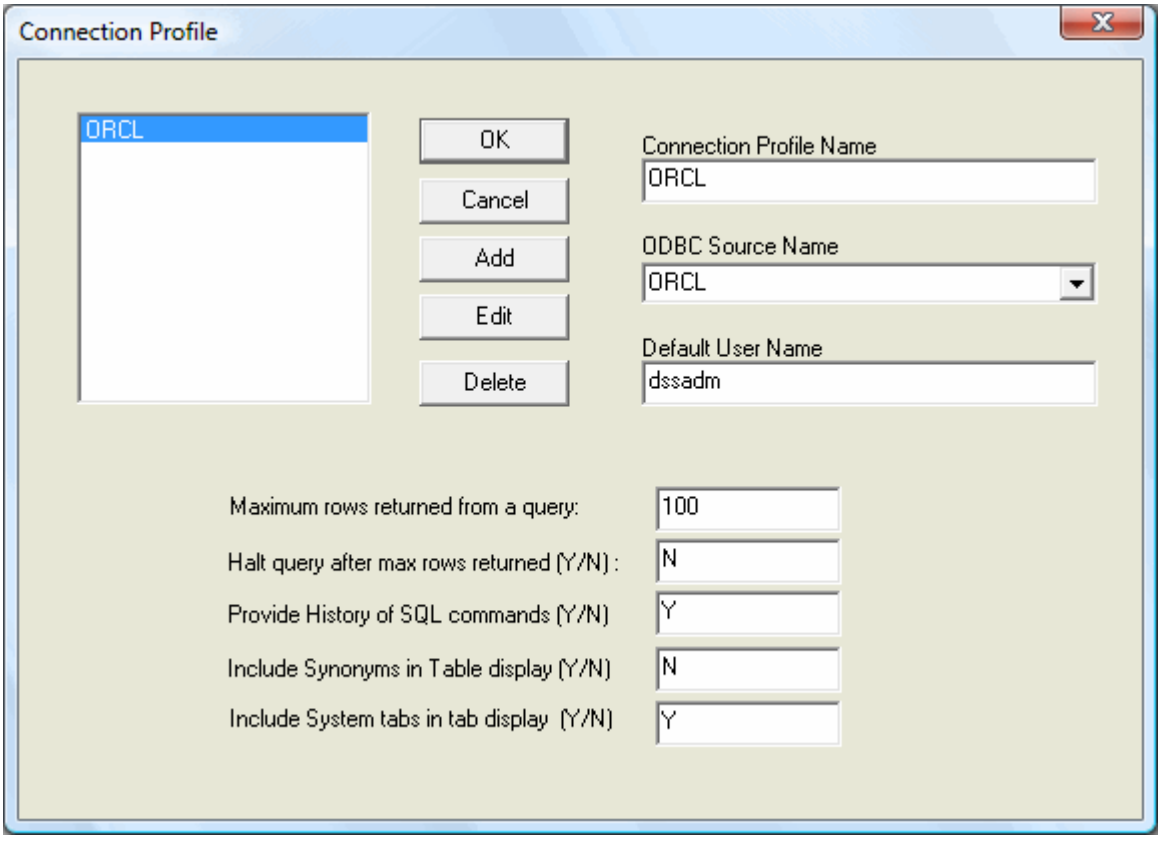


The history command drop down list box provides a list of all the successful commands that have been executed from the edit window. It does not include failed commands, or commands executed from the menus or the table pane. The history run button will run the command currently displayed in the history drop list. The history cut button will place the command shown in the history drop list into the paste buffer. The history paste button will load the contents of the paste buffer into the history drop list.

CHAPTER 4

Parameters and Options

When the new connection button is pressed Sql Admin will display the logon screen. By default this screen has all fields blank. It is possible to set a default connection by selecting the Tools/Options menu option. When this menu option is chosen the following screen will appear.



Listed in the top left hand box are all the currently defined connections. Double clicking on one of the listed connections makes that connection the default connection. A connection can be deleted by selecting the delete button when a connection is highlighted, or edited by selecting the Edit button.

Adding a Connection Profile

To add a new connection profile click the Add button. Enter a name for the connection in the 'Connection Profile Name' field. Often entering the same name as that of the ODBC Data Source is a good idea, but any name can be used. Next select an 'ODBC Source Name' from the drop down list of ODBC Data Sources.

If desired a default user name can be entered. If present this will populate the appropriate field in the logon screen.

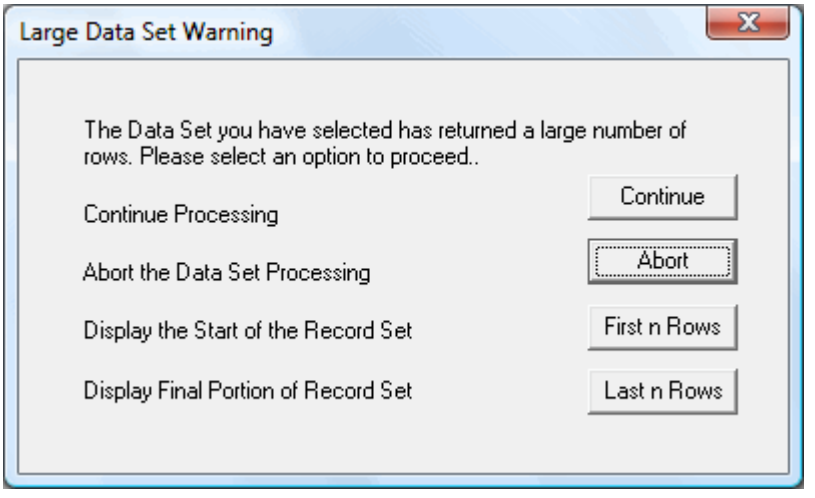
Maximum rows returned from a query

This field defines the maximum number of rows that will be returned from any sql statement or action that results in a database query. The action taken when this maximum number of rows is reached depends on the value in the next option. This maximum number of rows can also be set for the current session, and hence override the default value, by selecting the menu option Sql/Set Max. Rows Returned. The default value is 100 rows, or whatever value has been set for the profile.

Halt query after max. rows returned

If this field is set to 'Y' (yes) then the query is aborted after the maximum number of rows have been returned. A line is placed in the result set to indicate that the query was aborted.

If this field is set to 'N' (no) then a dialog appears after the maximum number of rows have been returned. This dialog requests an action from the user. An example of the dialog appears below.



The default action is to abort the query. In such a case a warning is placed in the row following the data set to indicate that the query was aborted. The Continue button will retrieve the entire result set.

Warning: The data returned from a database query is loaded into the memory of the PC. Returning large data sets may take a large amount of time, and may fail due to memory limitations. Returning more than 1000 rows is not recommended.

The choice provided in this option can be changed by selecting the Sql/Auto Quit on Max Rows menu option. This menu option is a toggle and a tick will appear next to the menu option when the auto quit (Halt) is enabled. The default is to provide the pop-up menu where no profile is being used (i.e. no Auto Quit).

Provide History of SQL commands

If this field is set to 'Y' (yes) the history toolbars are displayed. When set to 'N' (no) the history toolbars are not displayed. The display of these toolbars can be toggled via the View/History Bars menu option. The default is 'Y' where no profile is being used.

Include Synonyms in Table display

The right hand table pane does not normally display synonyms (for those databases that support the concept). By setting this option to 'Y' synonyms are displayed as part of the table tree view. The default is 'N' where no profile is being used.

Include System tables in Table display

System tables and views can be included in the table pane tree display by setting this field to 'Y'. If this field is set to 'N' then system tables and views are not included in the display. The default is 'Y' where no profile is being used.

CHAPTER 5

Sql Statements

This chapter covers the issuing, saving and recalling of sql statements, and sql command files (scripts). It also provides information on how to assemble sql statements from the output of another sql statement. Specific instructions for running database store procedures and statement blocks are included.

In This Chapter

Editing

The edit pane of the Sql Admin window allows for the entering and editing of sql command scripts. A command script can consist of one or more sql statements. A statement is deemed to be everything up to a terminating semi-colon. All statements in the script are executed sequentially when the 'execute/run' button is pushed. The normal Windows select, cut, paste, find and replace operations may be performed in this pane. Each line in the edit pane is terminated by a carriage return/line feed combination.

Comments

Comments may occur within the sql statement. When encountered they are removed prior to passing the statement to the database. Comments are identified by two concurrent dash characters (i.e. --). When encountered all remaining text on the line is ignored.

For Example:

```
select * from dim_date          -- select all columns from the date dimension
where cal_month = 200904        -- for the month of April 2009
;
```

In this example the command executed will be:

select * from dim_date where cal_month = 200204;

Including other script files

Additional script files can be included at execution time by using the '$sql_include ' prefix. When encountered at execution time the contents of these scripts are read and executed as if they had been entered in the edit pane.

For Example:

```
-- Script to create all required tables
--

$sql_include c:\build\groupa.sql
$sql_include c:\build\groupb.sql
$sql_include c:\build\groupc.sql
```

In this example the contents of the three files groupa.sql, groupb.sql and groupc.sql are read at run time and executed.

Saving and Loading Scripts

Four menu options exist to support the maintenance of script files. The edit pane of the Sql Admin window contains the contents of a script file. By default or at initial invocation this script file will be named sqlx where x is the window number. When Sql Admin exits you will always be asked if you wish to save the contents of the script file currently in use. Care should be taken to ensure that you are not overwriting a script that was opened sometime earlier.

Note: The Sql Admin edit pane is always associated with a script file. Therefore if you open a script file and execute it, that script file becomes active and can be written to. To prevent accidental modification of such a script you should open another script file or create a new one if you wish to issue further sql commands.

New Sql Script

A new sql script file can be started by selecting the menu option File/New Sql Script or by pushing the new script button on the toolbar. If the existing script has been modified but not saved a prompt will appear asking if you wish to save the changes to the existing script. A 'save as' file dialog will then appear to allow the selection of a directory and name for the new script. The extension is by default .sql. Once a file location and name have been specified the edit pane will be cleared and the script name placed on the window title bar.

Loading an Existing Sql Script

To load an existing sql script into the edit pane select the menu option File/Load Sql Script or push the load button on the toolbar. An 'open' file dialog will appear that will let you navigate to and select the script you wish to load. Once selected the script is loaded into the edit pane and the script name placed on the window title bar.

Saving a Sql Script

To save the contents of the edit pane to the corresponding script file select the File/Save Sql Script menu option or push the Save button on the toolbar. This action immediately overwrites the contents of the script file with the contents of the edit pane.

Save Sql Script under a New Name

The contents of the edit pane can be saved under a new script file name by selecting the File/Save Sql As.. menu option. A 'save as' file dialog will then appear to allow the specification of a directory and name for the new script. Once a file location and name have been chosen the contents of the edit pane are saved to the script file and the new script name is placed on the window title bar. The original script file is left unchanged.

Printing Scripts

The contents of the edit pane (i.e. the sql script) can be printed. To do so ensure that the cursor is located in the edit pane and select the File/Print menu option.

The contents of the table tree and results pane can not be printed directly. See the Handling Results chapter for an explanation of how to acquire printed results.

Executing

There are a number of different methods for executing code in the edit pane. The appropriate method and syntax depends on the type of statement or statements in the edit pane. Refer to the following sections on how to execute using the appropriate method.

Executing Sql Statements

A sql statement or group of statements can be executed by selecting the Sql/Run menu option or by pushing the blue run button on the toolbar. Alternatively the Escape key can be used to execute the statements.

The entire contents of the edit pane are executed when one of the run options is selected. If multiple sql statements exist then they will be executed sequentially. To prevent a statement from running it must be commented out, by placing two dash characters at the front of each line.

A sql statement is deemed to be everything up to a semi-colon. It is not therefore possible to compile a stored procedure or execute a statement block using one of the run options. A procedure would, for example, be considered a number of distinct sql statements.

 TIP: Do not use the run option to compile procedures or execute statement blocks. See the following sections on how to handle these entities.

Executing Procedures

A database stored procedure can be executed from the edit pane. The syntax required is however not intuitive. Stored procedures will be considered either Functions (returning a value) or Procedures (where no value is returned). Note that Procedures may return values in their parameter list.

Functions

The syntax for calling a database stored procedure function is:

{? = call *function*(*p1*,*p2*,*p3*)};

Where '*function*' is the name of the database stored procedure. '*p1*', '*p2*' and '*p3*' are the parameters of the stored procedure.

Procedures

The syntax for calling a database stored procedure is:

{call *procedure*(*p1*,*p2*,*p3*)};

Where '*procedure*' is the name of the database stored procedure. '*p1*', '*p2*' and '*p3*' are the parameters of the stored procedure.

Parameters

The parameters can be either numeric, character strings, or returned values. A number would normally be entered directly. A character string enclosed in single quotes and a returned value is represented by a question mark (?).

Examples:

A procedure called get_address has four parameters. The first parameter is a customer number, the second a name, the third a returned address, and the fourth a returned city name. The call may look as follows:

{call get_address(1,'Joe Bloggs',?,?)};

A function called get_customer_id has two parameters. The first parameter is a name and the second an address. The call may look as follows:

{? = call get_customer_id('Joe Bloggs','The Hermitage')};

TIP: Some ODBC drivers do not support this functionality. For the Oracle database the 'Microsoft for Oracle' ODBC driver does support the calling of stored procedures. Other ODBC drivers can corrupt the client layer when such commands are issued, resulting in an application crash.

Compiling Procedures

Database stored procedures can be compiled when loaded into the edit pane. This operation is only valid for some databases. To compile a stored procedure select the Sql/Compile PL/SQL menu option or push the blue 'compile a procedure or function' button on the toolbar.

If the stored procedure compiled successfully a message to that effect will appear in the results pane. If the compile failed then an attempt will be made to locate and display the errors in the results pane. The lines in the procedure that generated errors and the error message are displayed.

Executing Sql Blocks

A statement block can be executed when located in the edit pane by selecting the Sql/Execute Statement Block menu option. If the statement block completes successfully then a message to this affect will be displayed in the results pane. If an error occurs then the error will be displayed in the results pane.

An example of an Oracle statement block is as follows:

```
DECLARE

  v_step     integer;

BEGIN

  v_step := 10;
  insert into dim_date (dim_date_key) values (99999999);
  commit;

END;
```

Executing Historical Statements

To execute a historical statement select the statement from the drop down history list and the press the blue 'Run shown history command' button.

The example above shows that the statement 'select * from dim_date;' will be executed when the blue run button is pressed.

The results of the statement are displayed in the results pane in the normal manner.

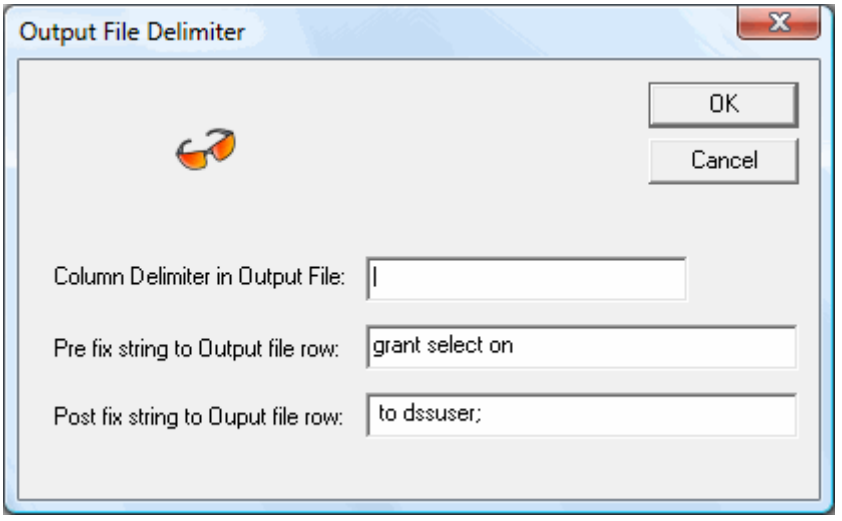
Building Statements from Results

Sql Admin can be used to build up sql statements from the results of other sql statements. For example we may wish to grant access to all of our tables to a new user. We could do this by issuing a statement to list all the table names and use Sql Admin to build up the statements to perform the grants.

Example:

We are connected to an Oracle database and wish to grant access to all our tables to a new user called dssuser. This is achieved by performing the following steps.

1. Enter the sql statement 'select table_name from user_tables;' into the edit pane and run the statement. A list of tables will be shown in the results pane, and the statement will be placed in our history list.
2. Delete the select statement. Ctrl/A in the Edit pane to select all the text and press the Delete key to delete it. The edit pane is now empty.
3. Select the menu option Sql/Set File Delimiter. A dialog will appear. Enter 'grant select on ' in the pre fix edit box, and ' to dssuser;' in the post fix box. Note the space at the end of the first statement and at the beginning of the second statement. The dialog box should look as follows:



4. Choose the menu option Sql/Out -> Edit + List. This will place a tick next to this menu option to indicate that it is the current results output method. When selected this will place the results of the sql statement in the results (list) pane and in the edit pane. The edit pane will have each row prefixed and suffixed by the string entered in step 3. If multiple columns where returned they would be separated by the delimiter field as shown in the dialog example above.
5. Using the history command drop list select the sql statement entered at step 1. Push the blue 'Run shown history command' button to the right of the history command drop list. The edit pane should now be populated with a series of sql statements such as 'grant select on table to dssuser;'.
6. Choose the menu option Sql -> Out List Only. This will ensure that all future results only go to the results pane and not the edit pane.
7. Run the commands in the edit pane by pressing the 'Run' button.

Historical Statements

Each successful sql statement issued in the edit pane is stored in the history list. It is also possible to paste any information in the paste buffer into the history list.

Saving and Loading Historical Statements

The menu options for saving, loading and clearing the history statement drop list can be found under the History menu heading.

Saving History Statements

The contents of the history statement drop list can be saved to disk by selecting either the History/Save History or History/Save History As ... menu options. The first option saves the history statements to the file garth.hst in the QAD program directory. The second option allows the specification of a history file. The default extension is .hst, and it is recommended that this extension name is retained to enable easier location of the history files for subsequent loading.

Loading History Statements

The history statement drop list can be populated from a saved history file. When statements are loaded from a file they are appended to the top of the history list. If only the information in the saved file is required then the history list must be cleared prior to the load with the History/Clear History menu option.

To load history from the default garth.hst file select the History/Load History menu option. To load history from any other saved history file select the History/Load History From... menu option and browse to the required file. As mentioned above the loaded statements are placed at the top of the history list.

Executing Historical Statements

To execute a historical statement select the statement from the drop down history list and the press the blue 'Run shown history command' button.

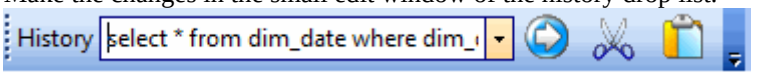
The example above shows that the statement 'select * from dim_date;' will be executed when the blue run button is pressed.

The results of the statement are displayed in the results pane in the normal manner.

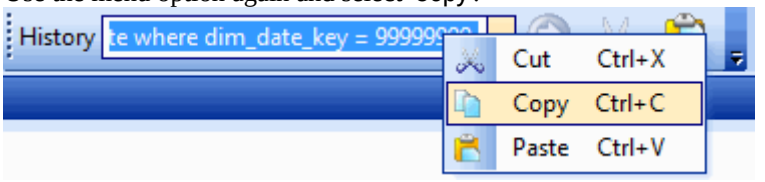
Historical Statement Editing

Although it is not possible to edit a history statement, it is possible to create a new entry in the history list from an existing statement. This can be done without using the edit pane or affecting the contents of the edit pane. The process is as follows:

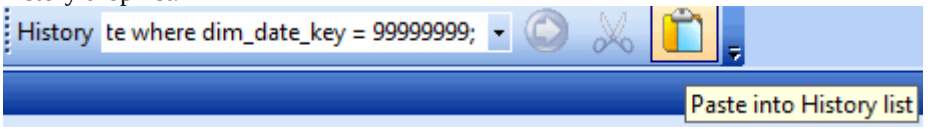
1. Select the history statement that you wish to alter.
2. Make the changes in the small edit window of the history drop list.



3. Use the menu option again and select '**Copy**'.



4. Paste the new statement to the history list by using the '**Paste into History list**' button to the right of the history drop list.



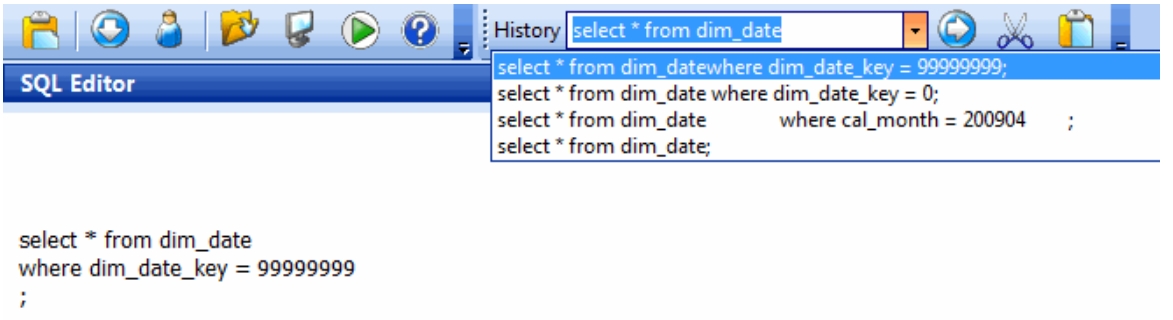
The new statement will now be at the top of the history list and can be executed in the normal manner by selecting the blue history run button.

Historical Cut and Paste

The contents of the edit pane can be manually moved to the history list be selecting the contents, then cutting or copying the contents and pasting them to the history list by pressing the 'Paste into history list' button to the right of the history list.

The currently displayed history list statement can be copied from the history list by selecting the 'Copy from history' button to the right of the history list. In the example below this is the button with the black scissors. The statement can then be pasted into the edit pane or some other location.

Where a statement extends over multiple lines the carriage return/ line feed characters that define the end of the line are stored in the history list. In the example below the two square characters after the first dim_date represent the carriage return and line feed characters.



Multiple line statements can be cut and pasted to and from the history list without changing the statement structure. Comments are, however, removed.

CHAPTER 6

Table Tree

The table tree is displayed in the top right hand pane. This chapter explains how to populate and refresh this tree. It also shows how to use shortcuts and menu options to issue queries relevant to the elements of the tree.


TIP: Double clicking on a table name displays the tables' data in the results pane.

In This Chapter

Table View Legend

The contents of the right hand table pane are a Windows tree structure containing the users, tables, views etc. of the database to which we are connected. The following screen shot shows the legend for the icons that appear in this tree structure.

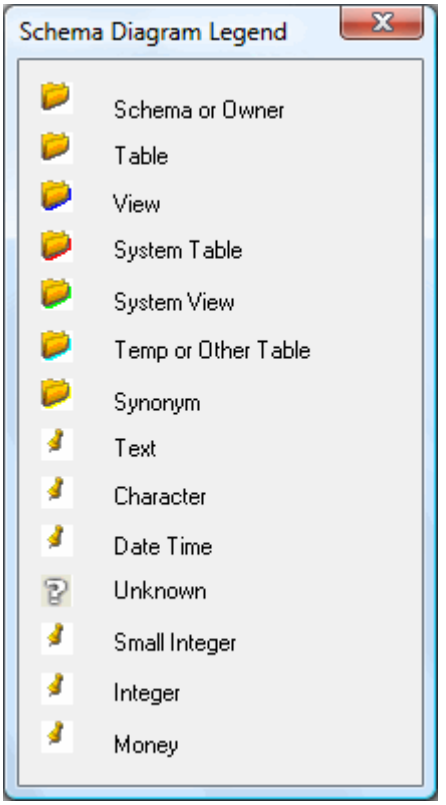


Table Display Options

A number of menu options exist to refresh or display the table tree. These options are covered in the sections below.

Display/Refresh Tables only

The menu option Schema/Refresh Tables only will display or redisplay the table tree structure in the table pane. This option will display all users and tables that are visible including system tables. Note that other tables may exist in the database, which our user cannot see due to insufficient privileges. The toolbar button 'Refresh Tree' performs the same function.

Display/Refresh My Tables only

The menu option Schema/Refresh My Tables only will display or redisplay the table tree structure in the table pane. Only tables and views owned by the current user are displayed.

Display/Refresh Tables and Columns

The menu option Schema/Refresh Tables and Columns will display or redisplay the table tree structure in the table pane. This option will display all users and tables that are visible including system tables. In addition it will also provide all the columns for all the tables and views. This may take some time.

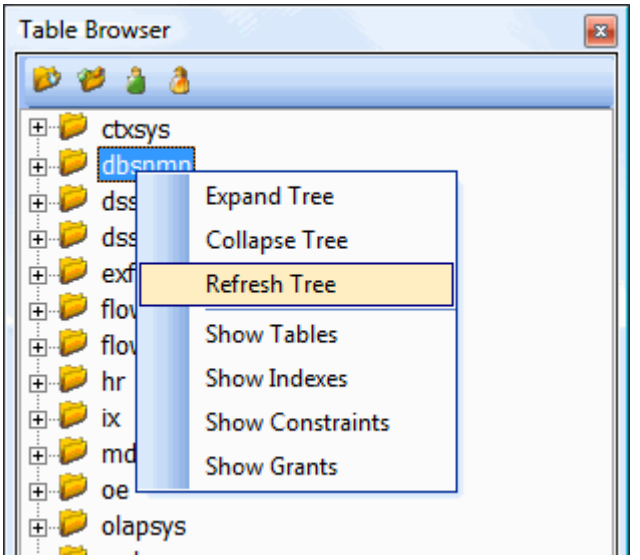
Warning: displaying all columns for all tables may take some time

Table Menu Options

Two pop-up menus are available within the table view pane. The first is available when a user or schema is selected. The second menu is associated with a table or view.

User Menu Options

When positioned over a user in the tree structure and the right mouse is pressed the following menu will appear.

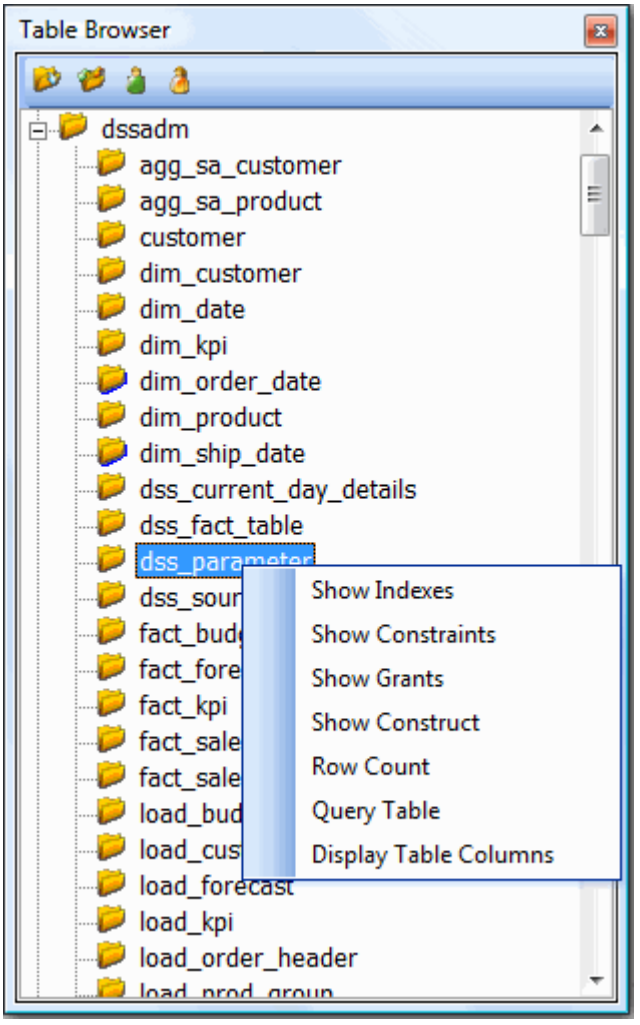


The 'Expand Tree' and 'Collapse Tree' menu options expand or collapse the particular node that is selected. The 'Refresh Tree' menu option refreshes the entire table tree.

The various 'Show' menu options will display in the results pane information pertaining to the selected user. The 'Show Tables' option will list all the tables owned by the user. The 'Show Indexes' will list all the indexes that are owned by the user and the tables to which they are connected. The 'Show Constraints' option will show all constraints owned by the user. Likewise the 'Show Grants' will show all grants that have been established by the selected user.

Table Menu Options

When positioned over a table or view in the tree structure the right mouse menu as shown below is available.



Show Indexes

Shows all indexes (and indexed columns) associated with the selected table in the results pane.

Show Constraints

Shows all constraints associated with the selected table in the results pane.

Show Grants

Shows all grants that have been made against the selected table in the results pane.

Show Construct

Shows a basic construct statement for the selected table in the results pane.

Row Count

Displays the number of rows in the selected table in the results pane.

Query Table

Performs a 'Select * from *table_name*;' statement where *table_name* is the name of the selected table. The results are shown in the results pane. This action can also be achieved by **double clicking the left mouse button** when positioned on the table name.

 TIP: Double clicking the left mouse button on a table name will issue a 'select *' statement against that table.

Note: If a table column is double clicked then the table is selected and ordered by the selected column. (i.e. the statement issued would be 'Select * from *table_name* order by *column_name*'.

Display Table Columns

Expands the tree structure to include all columns of the selected table.

CHAPTER 7

Handling Results

By default when a sql statement is issued the results appear in the results pane. It is, however, possible to redirect these results to the edit pane, a file or some combination of the three main output options.

It is not possible to print the results from the Sql Admin tool. To print the results you must either save the results to file and then print that file or send the results to Windows Excel and print from within Excel. See the following sections on how to perform these functions.

In This Chapter

Results Output to File

To send the output of a sql statement to a file select the menu option Sql/Out -> File Only or the menu option Sql/Out -> File + List. When either menu option is selected a 'Save As' dialog will appear to allow the specification of a directory and file name for the file. The second menu option will put the output to both the file and the results pane. This may be slow if a large amount of data is retrieved.

When either of the above menu options is selected a tick will appear next to the menu option to indicate the current data output method. This method stays in effect until changed by the selection of a different output option. The default option is 'Out -> List Only'.

The columns in the results are separated by a comma by default. This default setting can be overridden by selecting the Sql/Set File Delimiter menu option and changing the contents of the 'Column Delimiter in Output File' field to the desired delimiter. The delimiter may be multiple characters if desired.

Once the file name has been entered and a suitable delimiter chosen the sql statement can be issued. When all data is retrieved the file would normally be closed by selecting the Sql/Close File menu option. This option closes the file and reverts to the default output method of 'Out -> List Only'.

Results Output to Excel

The results of a query can be moved to the Windows Excel tool. To do so proceed as follows:

1 Issue the required sql statement displaying the results in the results pane.

2. Position the cursor in the results pane and using the right mouse menu select the menu option 'Excel it'. This will create or replace the file garth.xls in the QAD program directory. It will then invoke this file which in turn will invoke Windows Excel if it is associated with the file extension (.xls) and it is available on the PC.
3. Within Excel highlight the first column by clicking on the column header (normally 'A').
4. Again within Excel select the 'Data/Text to Columns...' menu option and answer the questions. Under normal conditions the data is comma delimited.
5. Save the Excel spreadsheet under a new name if you wish to retain it. The sheet will be overwritten the next time this procedure is undertaken if not saved under a new name.

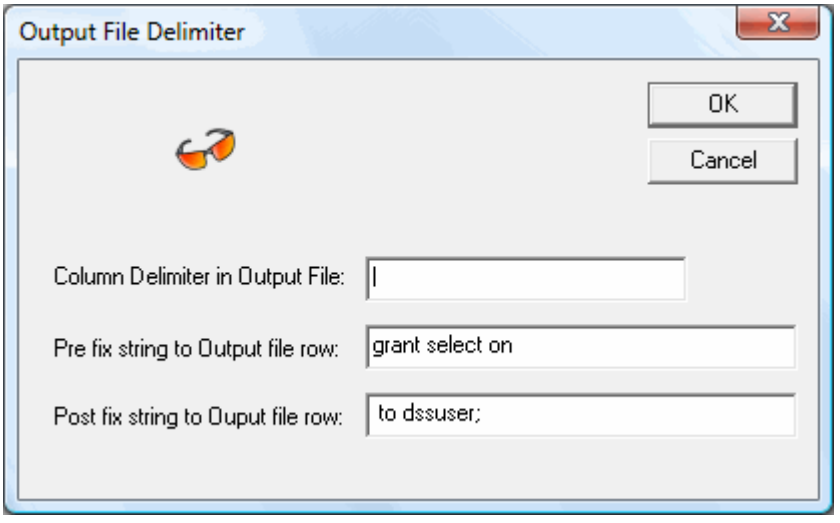
Building Statements from Results

Sql Admin can be used to build up sql statements from the results of other sql statements. For example we may wish to grant access to all of our tables to a new user. We could do this by issuing a statement to list all the table names and use Sql Admin to build up the statements to perform the grants.

Example:

We are connected to an Oracle database and wish to grant access to all our tables to a new user called dssuser. This is achieved by performing the following steps.

1. Enter the sql statement 'select table_name from user_tables;' into the edit pane and run the statement. A list of tables will be shown in the results pane, and the statement will be placed in our history list.
2. Delete the select statement. Ctrl/A in the Edit pane to select all the text and press the Delete key to delete it. The edit pane is now empty.
3. Select the menu option Sql/Set File Delimiter. A dialog will appear. Enter 'grant select on ' in the pre fix edit box, and ' to dssuser;' in the post fix box. Note the space at the end of the first statement and at the beginning of the second statement. The dialog box should look as follows:



4. Choose the menu option Sql/Out -> Edit + List. This will place a tick next to this menu option to indicate that it is the current results output method. When selected this will place the results of the sql statement in the results (list) pane and in the edit pane. The edit pane will have each row prefixed and suffixed by the string entered in step 3. If multiple columns where returned they would be separated by the delimiter field as shown in the dialog example above.
5. Using the history command drop list select the sql statement entered at step 1. Push the blue 'Run shown history command' button to the right of the history command drop list. The edit pane should now be populated with a series of sql statements such as 'grant select on table to dssuser;'.
6. Choose the menu option Sql -> Out List Only. This will ensure that all future results only go to the results pane and not the edit pane.
7. Run the commands in the edit pane by pressing the 'Run' button.

CHAPTER 8

Database Information

A number of menu options are available for looking at database specific information. These menu options only work for some databases. All options work for the Oracle database, and some options work for a number of other databases.

The two main menu headings for database information are 'Show' and 'System'. The 'Show' menu options provide information relative to the current user, where appropriate. The 'System' menu options provide system wide information. In both cases the command may fail or return no information if the current user does not have sufficient privileges.

In This Chapter

Database User Information

The 'Show' menu options provide information relevant to the user. The various options are:

Tablespaces

This option provides a list of all tablespaces that the user has access to. Each tablespace extent is shown together with the currently registered free space. Note that temporary tablespaces do not always show the total amount of available space.

Table Sizes

This option lists all objects belonging to the user that are consuming tablespace. It shows the object name together with the tablespace used and the amount of space allocated.

Db Links

This option lists all database links that are owned by the user.

Sequences

This option lists all the sequences belonging to the user.

Constraints

This options lists all constraints on all objects owned by the user.

User synonyms

This option lists all the synonyms local to the user.

Public synonyms

This option lists all public synonyms visible to the user.

Indexes

This option lists all indexes that are owned by the user.

Grants

This option lists all table grants either given to the user or on tables owned by the user.

Columns

This option lists all columns of all tables owned by the user.

Tables

This option lists all tables owned by the user.

Views

This option lists all views owned by the user.

Users

This option provides information about the user profile of the user.

Database System Information

The 'System' menu options provide information relevant to the database as a whole. The various options are:

User Sessions

This option provides a list of all user sessions running on the database. It also attempts to find the last sql statement issued by the session. Other information is also provided. See the sections on tracing and killing user sessions for more information.

Tablespaces

This option provides a list of all tablespaces in the database. Each tablespace extent is shown together with the currently registered free space. Note that temporary tablespaces do not always show the total amount of available space.

Table Sizes

This option lists all objects that are consuming tablespace. It shows the object name, owner name together with the tablespace used and the amount of space allocated.

Db Links

This option lists all database links.

Sequences

This option lists all the sequences.

Constraints

This option lists all constraints.

Indexes

This option lists all indexes.

Grants

This option lists all table grants.

Columns

This option lists all columns of all tables.

Tables

This option lists all tables.

Views

This option lists all views.

Users

This option provides information on all users defined in the database.

Rollback Stats

This option provides information on all the rollback segments available.

Rollback Users

This option provides information on the current usage of the various rollback segments.

Locks All

This option provides a list of all locks in the database.

Locks (User)

This option provides a list of all non system (user) locks.

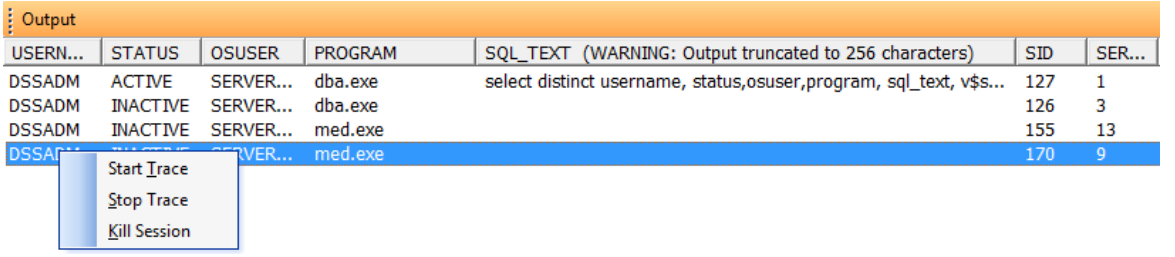
Locks Object

This option provides a list of locks showing the object that is locked.

Killing Sessions

For Oracle databases it is possible to use the Sql Admin tool to kill an active session provided that the user has the required privileges. To perform this function proceed as follows:

1. Select the menu option System/User Sessions or press the 'Show Oracle users (must be privileged)' button on the toolbar. The results pane will now contain a list of sessions similar to the following screen shot.

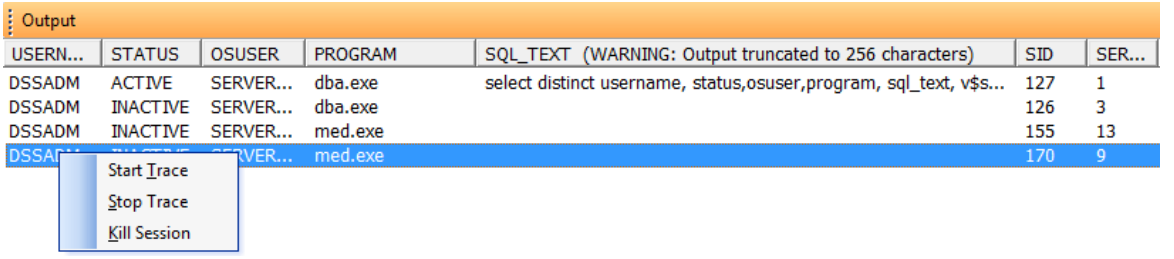


2. Select a username in the left hand column of the results pane and using the right mouse menu select the 'Kill Session' option. A prompt will appear asking if you are sure you wish to kill the selected SID, and when accepted the session will be killed. Note, however, that the session may not die immediately. Refer to the Oracle database manuals for more information.

Tracing Sessions

For Oracle databases it is possible to use the Sql Admin tool to start and stop a trace on an active session. To do so proceed as follows.

1. Select the menu option System/User Sessions or press the 'Show Oracle users (must be privileged)' button on the toolbar. The results pane will now contain a list of sessions similar to the following screen shot.



2. Select a username in the left hand column of the results pane and using the right mouse menu select the 'Start Trace' option. Tracing will start for the session provided the user has the require privileges.
3. Select the username for the same SID and using the right mouse menu select the 'Stop Trace' option.
4. Locate the trace file for the session. Consult the Oracle manuals on how to find the trace files and on how to process and analyzing these files. Normally the trace files are in the directory $ORACLE_BASE/admin/$ORACLE_SID/udump on the database server system.

Sql Admin Batch Mode

SQL Admin in batch mode can be called from a script file. This is useful when a user wants to do an ODBC extract from a table in a database to a file, which is then subsequently loaded into the data warehouse.

Call SQL Admin with Parameters below.

Switch	Parameter	Function	Notes
/A		Interactive Auto flag	Should not be used if batch mode is required. Interactive use only.
/B		Batch mode flag	Set this flag to run in batch mode.
/X		Append to output file flag	Appends data to the specified output file. If not present then the file is created or over-written.
/Z		Output file in Unicode format flag	By default output to a file is in ASCII mode. Set this flag for UNICODE
/O	ODBC Source	Used to specify the ODBC source.	This source must be present on the system where SQL Admin is being run from.
/U	Username	The username for the ODBC source.	
/P	Password	The password for the ODBC source	
/F	File name	Output result file name	If specified SQL Admin will output the results to this file name
/D	Delimiter	The delimiter character(s) to be used to separate the columns in a row.	
/M	Maxrows	Specifies the maximum number of rows to output.	If not specified then the maximum value of 999,999,999 is used.
/R	Replace	Replacement character	Delimiter characters found in the actual data will be replaced with this character(s) if specified.

/S	Sql command to run	The SQL command to be executed.	If multiple commands they smust be separated by a semi-colon and a space. E.g. “select * from a; select * from b”
/C	Sql script file to run	The name of a file containing the SQL Command(s) to run.	If multiple commands they should be separated by a semi-colon.
/V	Prefix string before each field	The pre-fix string	If desired a pre-fix string can be specified. This string is pre-pended to each column in the results.
/W	Postfix string after each field	The post-fix string	If desired a post-fix string can be specified. This string is appended to each column in the results.
/L	Log file to create	Log file name	If specified the log file will be created with the following string in it. nnnn rows outputted

General Usage Notes

The SQL Admin executable is called dba.exe and is located in the QAD program directory.

The switches /A and /B indicate either interactive or batch mode. Only one or the other should be used.

The option /S denotes a command to run. The option /C denotes a script from which to read the command(s) to run. Only one or the other should be used.

Where a switch has a parameter, that parameter can optionally be enclosed in quotes. Some parameters must be enclosed in quotes. For example a pipe delimiter must be entered as /D “|” to prevent the command line interpreter from processing the pipe. Also if you wish to have a space as a delimiter or replacement value it must be in quotes. E.g. /R “ “.

Example:

D:\PMS\Release\Dba.exe /B /O DataWarehouse2005 /Usa /Pwsl /F"D:\Pms\Temp\Wayner.txt" /D"|" /S"Select * from load_product"

Return Values:

- 0 – success or at least we think so.
- 1 – Warning: No data was returned from the command. The output file will be empty.
- 2 – Warning: The number of rows returned was limited to the value set by the M switch. More data was available.
- 3 – The specified script file as specified with the /C switch could not be found or was empty
- 5 – The specified output file as specified with the /F switch could not be created or opened.
- 7 – The ODBC Connection failed.
- 9 – The SQL Statement(s) failed to execute correctly.

Index