Benthic Software Golden

Table of Contents

1 Welcome to Golden

Golden6 is a multi-purpose SQL tool from Benthic Software. This help file is designed to help you get the most out of your Benthic Software purchase.

Important Note:

Remember that you may be using a database which is being accessed by others. Speak to your database administrator before using any ad-hoc sql tools. You must understand how data locks, run-away queries, data contention, and other considerations will affect you and other users. It is generally recommended to restrict your queries with proper WHERE clauses. Don't select all of a huge table if you only need to see a subset of records.

Features:

- Full Unicode support.
- Syntax highlighting SQL editor.
- Point and Click SQLBuilder with Oracle keywords, all accessible Oracle Objects, column datatypes, primary key and index information.
- Direct data export to Excel.
- Data export in multiple formats including CSV and HTML.
- Data sorting, formatting and printing.
- Running individual and multiple sql scripts.
- Timing of SQL statements.
- Explain Plan output.
- Transposed data results.
- Editing of simple (single table) result sets.
- Editing of single table in a multi table query.
- Favorites menu with quick access to commonly used queries.
- Support for prompt variables (&,&&).
- Support for calling external script files (using the @ syntax.)
- Support for Bind variables including the REFCURSOR type.
- Support for connect keyword in scripts.
- Support for DESC and EXEC commands.
- Support for the SPOOL command.
- Direct data export to Excel, Open Office Calc, CSV files and XML.
- High speed external delimited Import/Export.
- Many more!

Compatibility:

- Windows NT4, 2000, 2003, XP, Vista, 2008, Windows 7, Windows Terminal Server/Citrix
- 64bit Windows with the 32bit Oracle Client.

Support:

Please email us at support@benthicsoftware.com and we'll be happy to help you!

Trial Version:

The trial version allows unlimited usage for 30 days. If this program meets your needs please purchase a license at www.benthicsoftware.com.

2 Overview

This help document is broken up into several sections. The Welcome and Overview sections describe the program features and information about database drivers. "The User Interface" section lists each menu item and window in the program and explains what each one does. The "How to..." section lists commonly used features and describes how to use them and often gives examples to clarify various points. Lastly the "Tips" section lists all Keyboard shortcuts for the program and Frequently asked questions.

2.1 License and disclaimer

1. License Agreement Golden6 (the "Software")

"Licensee" means the person or company identified below, who is being licensed to use the "Software" identified in the title of this License Agreement.
"Licensor" means Benthic Software.

2. 30 Day Trial License

THIS IS NOT FREE SOFTWARE. Subject to the terms below, Licensee is hereby licensed to use this software for evaluation purposes without charge for a period of 30 days. If you use this software after the 30 day evaluation period you are required to make payment for the software.

3. Limited Nonexclusive License

Licensee acknowledge that you are acquiring only a limited nonexclusive license to use the Software (the "License"). Licensor shall remain the owner of all right, title, and interest in the Software and in any copies of it.

The License permits Licensee to install the Software on more than one computer system, e.g., a desktop computer and a laptop computer normally used by the same person at different times, provided that there is no possibility that the Software will be used on more than one computer system simultaneously; any such simultaneous use requires a separate license for each computer system.

Licensee may make a reasonable number of backup copies of the Software solely for backup purposes.

Licensee agrees not to make, nor to permit, the making of copies of the Software (including its documentation) except as authorized by this License Agreement or otherwise authorized in writing by Licensor.

Licensee agrees not to engage in, nor to permit, the de-compilation, disassembly, or other reverse engineering of the Software.

4. Disclaimer of Warranty

THIS PROGRAM IS SUPPLIED AS IS. THE LICENSOR DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THE WARRANTIES OF MERCHANTABILITY AND OF FITNESS FOR ANY PURPOSE. THE LICENSOR ASSUMES NO LIABILITY FOR DAMAGES, DIRECT OR CONSEQUENTIAL, WHICH MAY RESULT FROM THE

USE
OF THIS PROGRAM. THE LICENSEE MUST ASSUME THE ENTIRE RISK OF USING THE PROGRAM.
ANY LIABILITY OF THE LICENSOR WILL BE LIMITED EXCLUSIVELY TO PRODUCT REPLACEMENT OR REFUND OF PURCHASE PRICE.

To the best of our knowledge Licensee's use of the Software in accordance with its printed documentation will not, in and of itself, infringe any third party's copyright, patent, or other intellectual property right.

No employee, agent, dealer, or distributor of ours is authorized to modify this disclaimer of warranty, nor to make any additional warranties.

If implied warranties may not be disclaimed under applicable law, then ANY IMPLIED WARRANTIES ARE LIMITED IN DURATION TO 30 DAYS AFTER DELIVERY OF THIS COPY OF THE SOFTWARE TO LICENSEE.

Some states do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to Licensee.

This warranty gives Licensee specific legal rights, and Licensee may also have other rights which vary from state to state.

5. No Incidental or Consequential Damages

Independent of the foregoing provisions, IN NO EVENT WILL LICENSOR BE LIABLE TO LICENSEE FOR ANY INCIDENTAL, SPECIAL, PUNITIVE, EXEMPLARY, OR CONSEQUENTIAL DAMAGES ARISING FROM OR CONNECTED WITH THIS AGREEMENT OR LICENSEE'S USE OF THE
SOFTWARE, regardless whether Licensor know or have reason to know of the possibility of such damages.

Some states do not allow exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

6. Licensee's Indemnity to Licensor

Independent of the foregoing provisions, Licensee agrees to defend and indemnify Licensor against, and hold us harmless from, any and all claims, damages, losses, and expenses of any kind arising from or connected with the operation of Licensees business.

7. Termination

If Licensee materially breaches this License Agreement, Licensor may terminate

Licensee's right to use the Software by notice to Licensee.

Licensee agrees that, upon termination of the License, Licensee will either return to Licensor or destroy all copies of the Software in Licensee's possession.

8. Entire Agreement, etc.

This written License Agreement is the exclusive agreement between Licensee and Licensor concerning the Software and supersedes any and all prior oral or written agreements, negotiations, or other dealings between the parties concerning the Software.

This License Agreement may be modified only by a writing signed by the parties.

In the event of litigation between Licensee and Licensor concerning the Software, the prevailing party in the litigation, and the prevailing party in any ancillary disputes (e.g., discovery disputes) will be entitled to recover attorneys' fees and expenses from the other party.

This License Agreement will be governed by the law of the State of Massachusetts applicable to contracts executed and performed entirely in, and by residents of, that state.

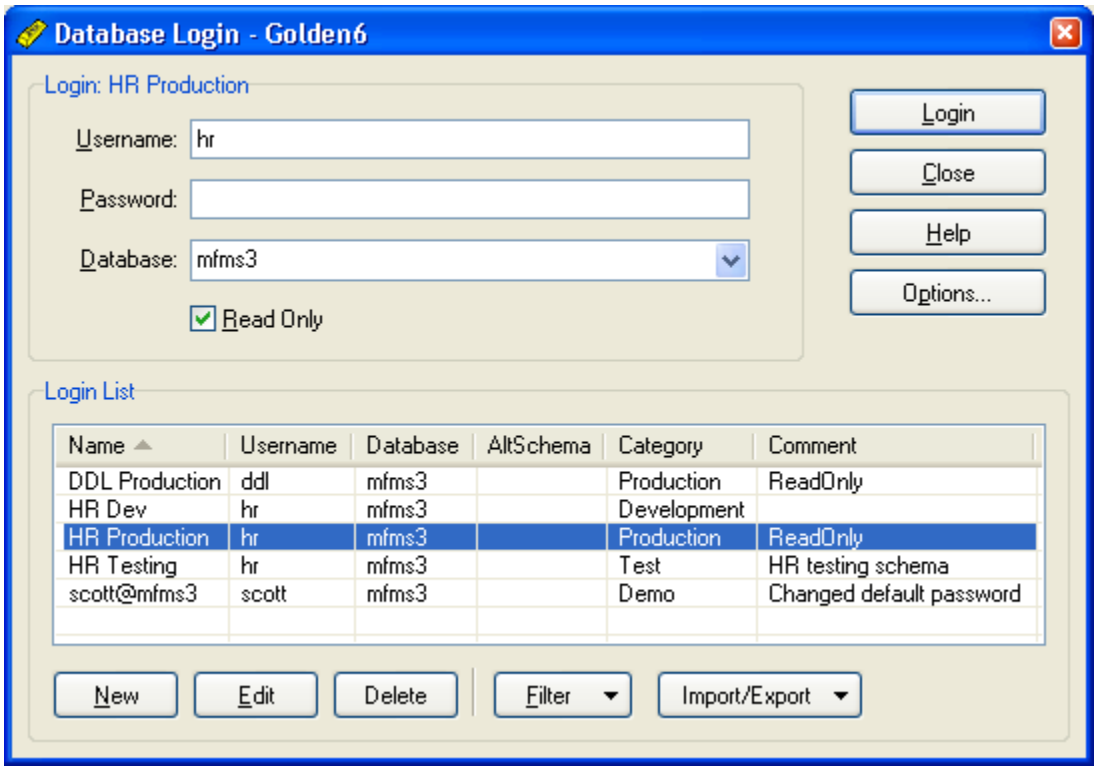
This License Agreement is effective upon the Licensee use of the Software. Licensee agrees that Licensor need not sign this License Agreement in order for it to take effect.

3 The User Interface

This section describes the user interface of Golden. It lists the windows you will encounter and also the menu and toobar buttons.

3.1 Login Window

The Login Window is where you enter the information needed to connect to your database and manage your list of saved logins.



- **Login area.**

  This is the information for the current login. These fields can be filled in by hand or can come from the Login List. If they are from the login list then the login item name will be displayed in the group caption (e.g. Login: HR Production).

  Username:

  This is the username of your database account. Check with your database administrator to find out what to use. To login as SYSDBA or SYSOPER use "username as sysdba" or "username as sysoper" in this field.

  Password:

  This is the password of your database account. Check with your database administrator to find out what to use. There is an option on the Login Options window that allows you to turn on or off password saving. Password saving can also be disabled by an administrator.

  Database:

  Ask your DBA for the connect string necessary for your database. Generally this will be a database alias from your tnsnames.ora file or an Easy Connect string (e.g., //myserver/sid).

  Hint: The database drop down list contains the "database" entries you've used previously as well as the aliases in your tnsnames.ora file if it could be found.

  Read Only checkbox:

  Check to make sure that your login can't edit data. This can be useful for querying production databases where you don't want to modify data accidentally.

  Options Button:

Click to view or change the login options.

- **Login List:**

  This is a list of your saved logins. It will initially be empty or contain login items that have been loaded from an older version of our products.

  Sorting the list:

  Click a column header to sort. Click the same one again for a descending sort.

  Filtering the list:

  Right click an item or select an item and click the "Filter" button to see options. Use the "Clear Filter" option to see all the list items.

  Import/Export:

  Click this button to see options. Choose "Export to file" to write all login list items to a file. There are options to import items from an older product or to import from a previously created export file.

  New:

  Click to create a new login item. Please see the Login List Item Edit Window topic for more information.

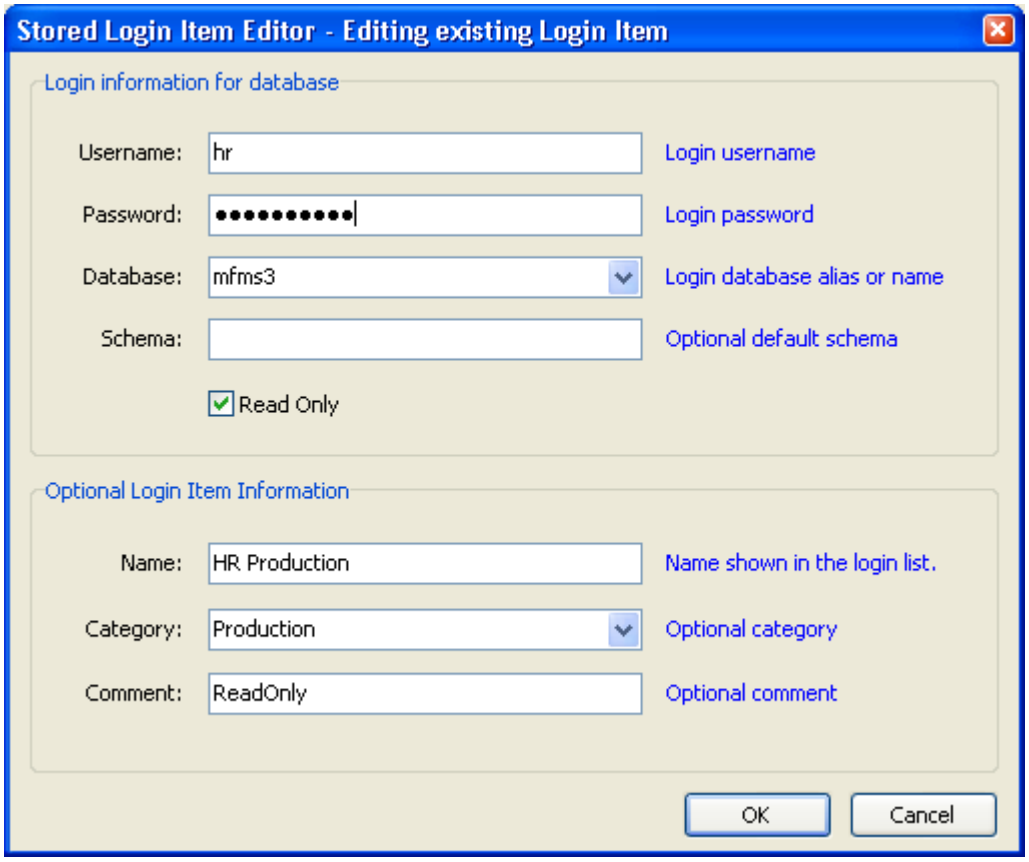
  Edit:

  Select one or more login items and click this button to edit them. Please see the Login List Item Edit Window topic for more information.

  Delete:

  Select one or more login items and click this button to delete them permanently. Note that you can backup the login list using the Import/Export button.

3.1.1 Login List Item Edit Window



This window is displayed when you choose to create a new login list item, edit an existing login list item, or choose to edit multiple login list items at once.

Creating a new Login List Item:

Clicking "New" under the login list in the login window will show the login list item edit window. The login information will be partially filled from the login window settings. Note that if the login option setting "Auto create new logins" is checked then new items will be created as you login to your databases.

Editing an existing Login List Item:

Selected a login list item and clicking "Edit" will show the login list item editor.

Editing multiple Login List Items:

Selecting multiple items from the login list and clicking "Edit" will show the login list item editor which will initially be empty. Entering a value in any field will set that field for all selected items.

Login Item Fields:

Username, Password, Database:

These are the fields described in the Login Window help topic.

Schema:

This is an alternate default schema. It is an optional value that will set the default schema on login to a different schema than the login username. You need to have been given

Oracle access rights to the alternate schema to use it. Leave this field blank in most cases.

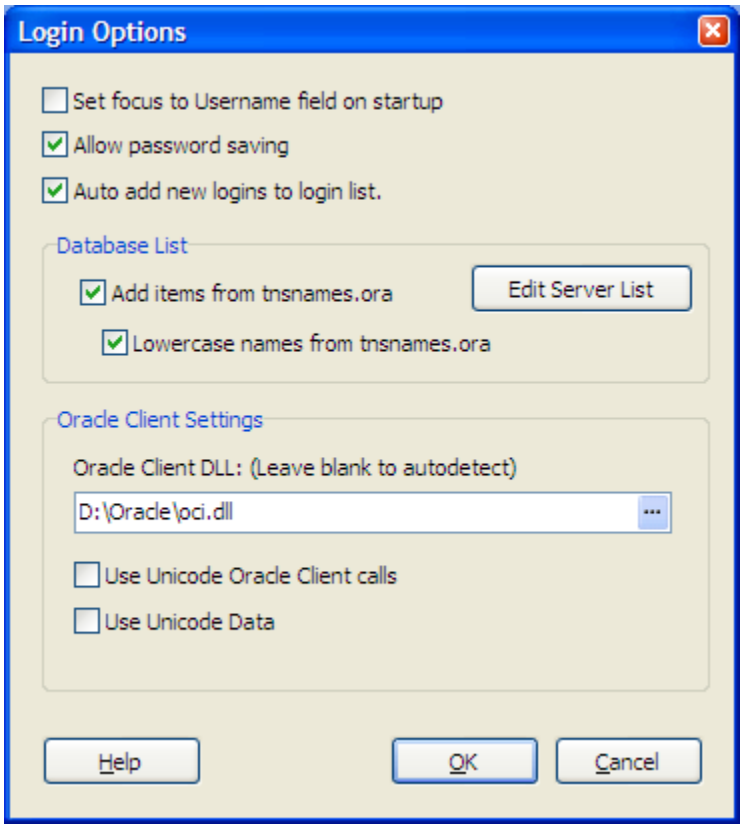
Read Only Checkbox:
Check this to make the login read only.

Name:
This is the name of the login item that is shown in the first column of the login list. If you leave this blank it will be set to [username]@[database]. If a login item with this name already exists this one will have a number added to make it unique.

Category:
This is an optional field where you can create a category that goes along with your login item. Common ones are "Productions", "Development", "Test", etc. These fields can be used as a filter in the login list to only show a specific category.

Comment:
Any text comment for the item.

3.1.2 Login Options window



The Login Options Window is where options related to the login window are set.

This window contains the following settings:

Set focus to username field on startup:
Sets the focus to the username field when the login window is opened.

Allow password saving:
Uncheck this option to never save passwords for login items. **This will remove all passwords from your list the next time you open the login window.**

Auto add new logins to login list:
If this option is checked logins will be added to the login list as you login to your databases. Automatically created login items have a comment of "Auto".

Database List:
These options determine if the database drop down list will include databases from your tnsnames.ora file and if they will be automatically lowercased.

Oracle Client DLL:
If login fails and indicates that the registry settings are incorrect or that the oci linking file can't be find, you can use this field to hardcode the correct oci dll. Look in your ORACLE_HOME\bin directory for a file named oci.dll and set this field to that name (e.g., oci.dll, ora806.dll) This setting can be used to force use of the Oracle Instant client (which doesn't require registry setting changes.)

Use Unicode Oracle Client calls:
Uncheck this option to use the older ANSI Oracle Call Interface. The default setting (checked) allows you to use Unicode characters in your SQL statements. If you are using an older Oracle client than your server version unchecking this and 'Use Unicode Data' can help you continue to work. It is recommended to update your Oracle client to at least the version of your database.

Use Unicode Data:
Uncheck this option to use the client's character set for data retrieval. This is useful if the server's character set is set incorrectly such that it won't convert properly to Unicode.

3.2 Main window

The Main Window consists of the following parts:

The menu and toolbar:

The menu items and toolbar buttons all have quickhelp hints. As you pass over the button or menu item a short description will appear in the message area (at the bottom of the main window.)

Tabs:

The main work area can contain multiple 'tabs'. One tab is open for each file or script that you are working on. Please see the topic 'How to work with public and private tabs.'

The SQL Editing area:

Type your SQL statement(s) here. You may use tabs and returns to format your statement. You may quickly execute your statement(s) by typing <Shift-Enter>.

Script and delimiter examples.
Enter and run SQL statements.

The splitter bar:

This horizontal bar can be dragged up or down to change the relative sizes of the editing and results areas.

The data grid results area:

When you execute an sql query, the results will show up here. You may resize the columns by sliding the header dividers left and right. You may copy results to the clipboard with the Edit menu. Note that results are put in the clipboard in a standard spreadsheet like format that may be easily pasted into Excel or other spreadsheets.

The status bar and timing information:

The status bar at the bottom of the main window has three "panes" The pane on the left shows informative messages and status messages while running statements and scripts. The middle pane shows how many records were affected by the last statement. The pane on the right shows the elapsed time of the last statement/script execution.

3.3 Options window

The Options Window is where most program options are set. Changed options can be just for the current session or can be made the default by clicking "Ok & Save as Defaults". The "Restore all Defaults" button can be used to reset all options to their original settings. The individual options have help text that can be seen by hovering your mouse cursor over the item for a few seconds.

General:

Date & Time Format Options
Tabs & Windows Options
Script & General Options
Restore all Defaults button

Editor:

SQL Editor Settings
Typing Popup Options

DataGrid:

DataGrid settings including Font and autosizing options.

Menus:

Settings related to the program menus.

Syntax Highlighting:

Settings that affect the appearance of keywords, comments and strings.

SQLBuilder:

Settings that affect the SQLBuilder.

File Filters and Extensions:

Allows changing the default list of file extensions that Golden works with.

Advanced Options:

Miscellaneous options.

3.4 SQLBuilder window

The SQL Builder allows you to build an SQL statement by pointing and clicking. Clicking on a word inserts it into the cursor position of the current sql buffer. If the Smart checkbox is checked then the statement will be formatted nicely. The dictionary checkbox toggles the table list between the current schema and the data dictionary.

See "How to use the SQL Builder to build SQL statements" for an example!

The SQL Builder window consists of the following parts:

The SQL Builder toolbar:
The toolbar buttons all have quickhelp hints, just pause the mouse cursor over a button for a second, and an informative message will pop up. The small buttons insert a comma, a space, and an apostrophe respectively. The "Refresh" button will update the object tree using the user account of the currently displayed tab.

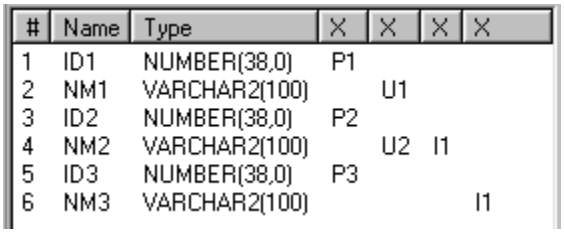
The SQL keyword list:
This list has commonly used SQL keywords. Click on a word to insert it into the active sql text buffer at the current cursor position (or replace the current selection.) Note that this list comes from the sqlbuild.txt text file. You may edit this file with any text editor to create a customized keyword list.

The splitter bars:
The horizontal bars can be dragged up or down to change the relative sizes of the lists.

The object tree:
This is a list of the tables, etc. available to you. Click a table name to fill in the listbox below with the column names of the table. Double click a name to insert it into the current sql text buffer. Note that the view can be changed to show User Objects, All Accessible Objects, or Dictionary Objects. Right click this field for more options.

The object tree search field:
Enter some text in the field and press Enter (or use the up and down arrow buttons) to search for the text in the current object tree. Note that only loaded information will be searched. You may have to expand some of the tree items in order to load the object information. There is also a right click menu called 'Fully expand this item' to quickly fill in an owner's objects.

The object detail list (Column List):
Click on the object tree above to fill in this list. Click on an item to insert it into the current tab text. For tables, this list also shows the datatype of each column, the primary key and any unique or nonunique indexes. Click on the column headers to sort by that column (click again to reverse sort.) Right click this field for more options.



#	Name	Type	X	X	X	X
1	ID1	NUMBER(38,0)	P1			
2	NM1	VARCHAR2(100)		U1		
3	ID2	NUMBER(38,0)	P2			
4	NM2	VARCHAR2(100)		U2	I1	
5	ID3	NUMBER(38,0)	P3			
6	NM3	VARCHAR2(100)				I1

This column list shows a table with 6 columns and:

- A primary key consisting of (ID1, ID2, ID3)
- A unique index consisting of (NM1, NM2)
- A nonunique index consisting of (NM2)
- A nonunique index consisting of (NM3)

3.5 Print Preview window

The Print Preview Window shows a representation of what a printed SQL script or result set would look like. It allows you to change margins, display of grid lines, page size, orientation, and other printer related settings.

The Print Preview Window consists of the following parts:

The Toolbar:

Print button:
Click the print button to bring up your printers options page.

Page setup button:
Click to see the page options for this printing.

Zoom List:
Changes the zoom and size of the preview.

Page buttons:
Changes the visible page in the preview.

Close button:
Closes the preview.

The Page View:

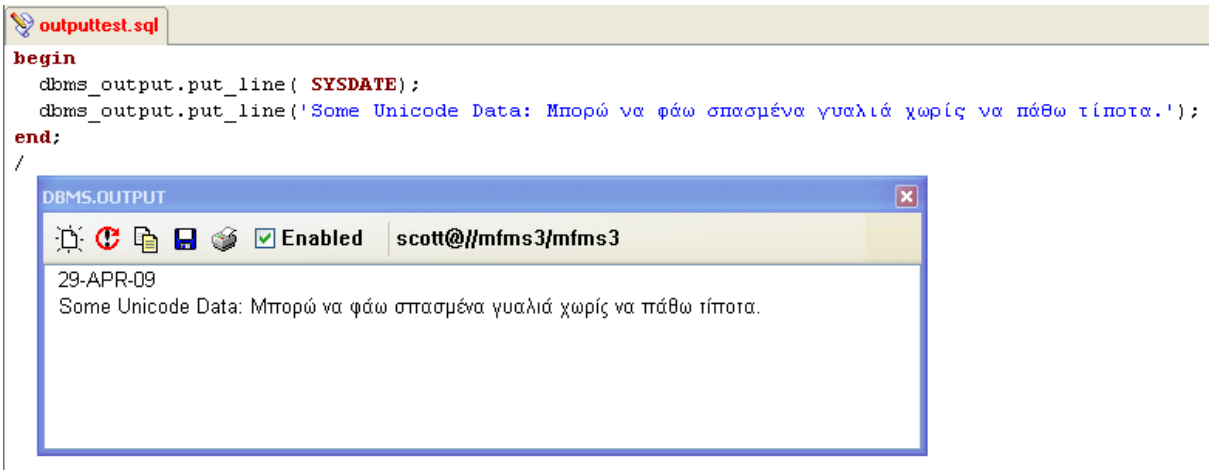
The page view shows a graphic of what the current report will print.

3.6 DBMS Output window

The DBMS Output window is for use with PL/SQL to output data.

Showing the window initializes the DBMS_Output package, you do not have to do it yourself. Then just call the dbms.put and dbms.put_line statements in your code. This is often used for pl/sql debugging. There is an example in the How to use the DBMS Output window topic.

The DBMS Output contents can be printed, cut to the clipboard, or cleared via the toolbar buttons.



4 How to...

This section describes how to perform various common tasks in Golden.

How to...

4.1 Work with public and private tabs

The main work area consists of a tabbed 'notebook' where multiple files can be opened at once. Tab text is underlined in red when the tab's SQL script has been modified and needs to be saved.

There are two types of tabs, public and private.

Public Tabs:

Public tabs share the main program login displayed in the caption of the main window. For example, if you logged into scott/tiger at startup, all new public tabs will share that single connection to the database. This conserves resources. Since they share a connection to the database, some operations can cause blocking. Basically, long operation that don't involve retrieving records (updates, deletes, etc.) can block other public tabs from operating. Note that as soon as the blocking operation is finished, work will continue as normal. If you have a long statement to run, it is better to run it in a private tab, described below.

Private Tabs:

Private tabs are tabs that have their own private database session. When creating a private tab (from the 'File' menu) you will be prompted for the user/password/server information.

Private tabs display their username and server above the script window. Besides allowing you to connect to multiple accounts/servers at once, private tabs will never cause blocking as described above. The file menu has choices for creating public and private tabs.

4.2 Enter and run SQL statements

First login to the database when prompted after you start Golden. This is the account that will be used for all queries unless you create a Private Tab (a tab that has its own separate connection to a database.) Once Golden finishes the login, you will see at least one 'tab' window. A tab window is a window inside the Golden main window which has an sql editor on top and a datagrid on the bottom. You type sql statements or scripts into the editor portion and see the results of queries in the datagrid. After typing a statement (or opening a script file) you may run it by using the 'Run Script' menu option (on the Script menu), pressing F5, or by clicking the Run Script button on the button bar. If your statement was a query, the results will be displayed in the datagrid. The elapsed time to run the statement(s) will be displayed on the right hand side of the status bar. The number of records affected by your statement will be displayed in the middle of the status bar.

You have the ability to run an entire script, run a script starting at the current cursor position, or run a single statement and stop. Please see the rest of the help pages for more information. Make sure to check out the keyboard shortcuts page!

There are many other ways to run scripts. All the options can be seen by looking at the Script menu.

Multi-Statement Scripts:

Separate SQL statements with a semi-colon and a linefeed.

Example:

```
delete * from emp where ename like 'F%';
select * from emp;
```

End PL/SQL blocks with a '/' character alone on a line.

Example:

```
delete * from emp where ename like 'F%';
begin  -- starts a pl/sql block
      insert into emp select * from emp2;
end;
/
select * from emp;
```

4.3 Work with the data grid

Overview:

Golden's data grid is where you will see the results of your queries. It features sorting, column moving/resizing, filtering and more. Also see: Edit data with Edit Mode

Selecting Cells, Columns, or Rows:

Click a cell and drag to select a range of cells. Click a column header to select a column. Click and drag a column header to select multiple columns. Use Ctrl-Click column headers to add or remove columns from the current selection. Click a row "header" (the leftmost column of the row)

to select a row. Click and drag a row header to select multiple rows. Use Ctrl-Click row headers to add or remove rows from the current selection.

Column Resizing:

To resize a column, drag the header's right edge to the left or the right. Data that doesn't fit in the cells can optionally had an ellipsis (...) added to them. Double click the header's right edge to autosize the colum to the longest visible cell.

Column Moving:

To move a column, first select it by clicking the header and then drag the column to the left or the right.

Sorting:

To sort a column, click the 'sort button' that is displayed in the column header. The sort button will toggle between an ascending or descending sort. Control clicking another sort button adds a secondary sort.

Filtering like current cell:

To filter the grid to show only matching cells, select the cell and choose 'Filter records like selected cell'. Choose 'Clear filter' to remove the filter.

4.4 Edit data with Edit Mode

Overview:

Besides writing insert, update and delete SQL statements you can edit data in Golden's datagrid after running a query in Edit Mode. Golden will allow you to edit simple queries (queries of a single table) or the first table of multi-table queries. Golden works best when the first column in the select statement is the ROWID of the table you wish to edit (an example would be 'select e.rowid, e.* from emp e') If you do not include the ROWID as the first column then the table must have a primary or unique key to be edited. Note that queries with multiple tables MUST include the ROWID of the first table in the from list (which is the editable table) as the first column. A simple example of this would be

```
select e.rowid, e.empno, e.ename, d.deptname from emp e, dept d where e.deptno = d.deptno
```

This query would let you edit the empno and ename fields of the EMP table. The editable table name is shown on the toolbar above the datagrid. Columns that can't be edited are shown in RED.

Entering Edit Mode:

Type in a query and click the data editing button (the one that looks like the run button but with an 'E' on it.) You can then directly edit cells in the datagrid. Records get saved to the database when you move the current focus to a different record (by clicking or using the arrow keys.) You can also use the 'Accept Edit' button on the editing button bar. Press escape to abort editing the current cell and then the current row (so you may have to press escape twice to restore a row to it's pre-edited state.

Transactions:

The following only applies if you do not have AutoCommit on in the program options.
No other users of the database will see your changes until you click the commit button or logout. You will see the changed data if you requery the database. This can be very useful to check your work. Then if you are satisfied with the results you can commit, or rollback if you don't want to accept the changes and return the database to the state of your last commit. It is a good idea to commit your data after you have made sure that it is correct. Oracle will commit your data changes when you cleanly logout from the database. Oracle will rollback your data changes if you lose your connection to the database.

Selecting and editing a cell:

Note that a datagrid cell can be the 'current' cell but not be in edit mode. You can tell if a cell is in edit mode if the text cursor is shown inside the cell. You can enter edit mode by clicking on data after you have selected the cell (but not double clicking, click once to select and again after a moment to move the cursor into the text.) You can also press the F2 key to begin editing (this is a Windows standard.) Once you are finished editing, you can press Enter or Tab to accept the edit.

Inserting Data:

You can insert records by typing them into the bottom empty record displayed in the datagrid or by pressing the Insert key or button to insert a record anywhere in the datagrid. You can also paste from the clipboard into the datagrid. The data should be tab delimited text data. You will be prompted if you are trying to paste more than one record. As mentioned above, exiting the record to another record will send the data to the database.

Updating Records:

You may update records by typing values right over the existing ones. You may update by pasting data from the clipboard. You will be prompted if you are trying to paste more than one record. The paste will occur starting at the currently selected cell.

Deleting Records:

You may delete records by moving to the record and then clicking the 'Delete records' button. You can select multiple records to delete by clicking and dragging a selection on the left most column of the datagrid. Ctrl-Clicking allows you to select multiple individual records. You will be asked to confirm your deletion.

Hint:

You can still edit a table that doesn't have a primary key if you use the rowid as the first field in the query
(i.e. select rowid, emp.* from emp)
If you are getting a message saying "Another user has modified the record" and you know that this hasn't happened, try adding 'ROWID' as the first column and re-run the query in edit mode.

4.5 Use the SQLBuilder to build SQL statements

Here is an example of using the SQL Builder to create the sql statement "select empno, ename, job from emp where mgr is not null" This example assumes that the user logged on as SCOTT/TIGER.

1. Using an empty sql edit window, click on the SQL Builder toolbar button.
2. Click on 'Select' in the keyword pane of the SQL Builder, this will copy the word 'Select' into the sql edit window.
3. Now click on the 'emp' table in the table pane of SQL Builder. Note that the columns list fill with the column names and datatypes of the emp table. Notice that the 'emp' word does NOT get pasted into the sql edit window.
4. Now click on 'empno' and then 'ename, and then 'job' in the column pane of sql builder. Note that the fields and commas are pasted into the sql edit window.
5. Now click on 'from' in the keyword pane.
6. Now DOUBLE CLICK 'emp' in the table pane. You must double click a table name to paste it into the edit window because single clicking it just displays its fields.
7. Now click on 'where' in the keywords pane, 'mgr' in the columns pane, and 'is not null' in the keyword pane.
8. You're done! Click the Run button to run the query!

4.6 Use the DBMS Output window to get pl/sql output

To display output from the DBMS_OUTPUT package you must first show Golden's DBMS Output window (from the View menu.) If the window is displayed, any DBMS_OUTPUT statements (put, put_line) will display data in the window.

Here is an example of a pl/sql block that output data to the DBMS Output window:

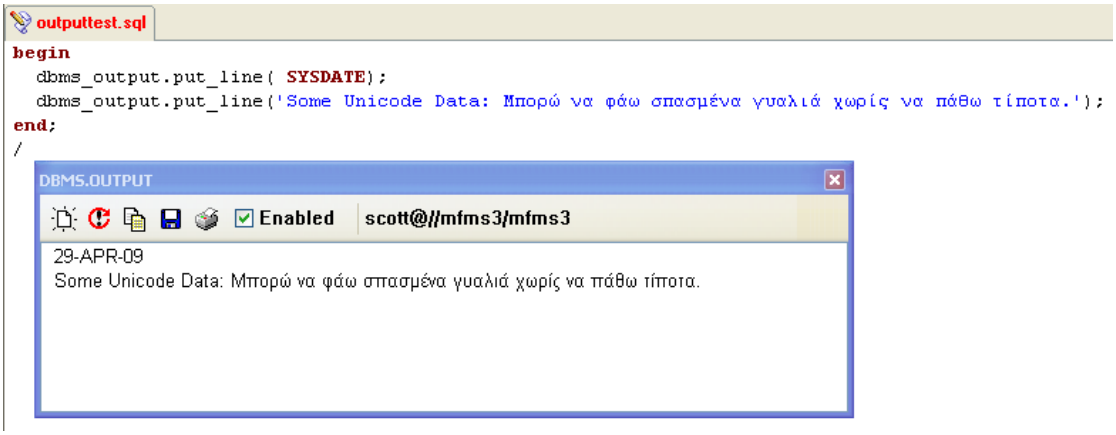
Make sure to show the DBMS Output window before running it. Note that you can cut and paste from this window.

```
declare
      CurrTime varchar(20);
begin
      select to_char( sysdate, 'HH24:MI:SS' ) into CurrTime from dual;
      dbms_output.put_line( 'Current Server Time:' );
      dbms_output.put( '    ' );
      dbms_output.put_line( CurrTime );
end;
/
```

Produces the output:

```
Current Server Time:
  09:39:09
```

Here is an example of a screen shot showing a script which called put_line and the output in the dbms.output window:



4.7 Use the DESCRIBE command

The DESC[RIBE] command is used to display the structure of tables and the arguments of stored procedures.

Here is an example of its use:

```
DESC EMP
```

This will display the columns and datatypes that make up the EMP table.

DESC UPDATE_ACCOUNTS

Will display information about the arguments and return value (if it is a function) of a stored procedure. You can also describe functions and procedures in packages by describing the package.

4.8 Use the EXECUTE command

The EXECUTE command (can also use EXEC) is used to run stored procedures on the server.

Bind variables can be used with the EXEC command to pass arguments and return values including refcursors. Please see the 'How to use Bind Variables' help topic for more information and examples.

Output of OUT arguements and function results are displayed using 'BIND variables.' Bind variables are names starting with a colon (i.e. :RETVAL) The contents of bind variables are displayed when the exec call is complete.

Here is an example of its use:

```
REM My sample function:
create or replace function marktest( arg1 number, arg2 out varchar )
return number is
begin
  arg2 := arg1 + 10;
  return arg1-5;
end;
/
REM Here is the execute call:
EXEC :RETVAL := marktest( 11, :OUTARG );
PRINT
```

Running the above script will display:

RETVAL = 6
OUTARG = 21

4.9 Use the EXPORT command

Query results can be saved to various file formats or directly export to an Excel or OpenCalc (Open Office) spreadsheet using the EXPORT command. Please be sure to not have the "clear datagrid on statement" option on as it will clear the datagrid just before the export.

Saving to a CSV file:

This will save the current data in CSV format. The fields will be comma separated and fields will be double-quote qualified if needed. Data may contain commas (in which case the field will be qualified with double-quotes).

EXPORT CSV "FILENAME" [NOTITLES]

Saving to a Tab delimited file:

This will save the current data in TAB delimited format. The fields will be tab separated and fields will be double-quote qualified if needed. Data may contain tabs (in which case the field will be qualified with double-quotes).

EXPORT TAB "FILENAME" [NOTITLES]

Exporting to Excel or OpenOffice Calc:

This will export directly to Excel or Calc. Note that if Excel is busy when the export is started you may see an error and have to try again when it isn't busy.

EXPORT EXCEL ["TemplateFilename"|NONE] ["SaveToFilename"] [NOOVERWRITEWARNING] [CLOSE]

EXPORT OOCALC ["TemplateFilename"|NONE] ["SaveToFilename"] [NOOVERWRITEWARNING] [CLOSE]

Using the CLOSE option will always open a new instance of Excel or OOCalc and then close it after the file has been saved.

4.10 Use the SPOOL command

Golden supports the 'SPOOL' command. This command allows you to store the results (and optionally the SQL commands) of a script to a text file. You can also spool data to the "Text view" of the datagrid area. To do that, change the view to the text view and check "Spool to text view" before you run your script. Here are the commands related to this feature:

SPOOL FILENAME [OPTION]
Will copy results and timings to a text file. If the filename contains spaces enclose it with double quotes (i.e. SPOOL "C:\MY FILES\MY SPOOLFILE.TXT"
You do not need to use this command if you are using the "Spool to text view" feature mentioned above.

Options are:
CREATE: Create a new file and show an error if the file already exists.
REPLACE: Create a new file and replace it if it already exists.
APPEND: Append data to the end of an existing file (will create a new file if it doesn't exist.)

SPOOL OFF
Turns off spooling and closes the spool file.

SET ECHO ON/OFF
If ECHO is on then SQL commands are placed in the spool file before the results.

SET FEEDBACK ON/OFF
Determines if statement stats (rows affected and timings) are reported..

SET HEADING ON/OFF
Determines if column names and the underlines are included when spooling data.

Example script using the SPOOL commands:

```
REM Start a spool file:
SPOOL "C:\MYSPOOLFILE.TXT"
SET ECHO ON
SELECT * FROM DEPT;
INSERT INTO DEPT ( DEPTNO, DNAME, LOC ) VALUES ( 60, 'HR', 'ORLANDO' );
SPOOL OFF
```

will generate the file:

```
06/25/2001 09:53:10 am Golden Spool File

SQL> SELECT * FROM DEPT

DEPTNO DNAME      LOC
------ ---------- --------
10     ACCOUNTING NEW YORK
20     RESEARCH   DALLAS
```

```
30     SALES      CHICAGO
40     OPERATIONS BOSTON

4 rows selected in 0.002 Secs

SQL> INSERT INTO DEPT ( DEPTNO, DNAME, LOC ) VALUES ( 60, 'HR', 'ORLANDO' )

1 rows created in 0.002 Secs
```

4.11 Use Prompt Variables

Prompt variables allow your scripts to be more interactive. Prompt variables are variables that can be defined and given a value which will be substituted in place of the variable before statements are passed to the database for processing. Prompt variables are specified using the & and && characters. The use of these can be a bit confusing at first, but they work in Golden exactly as they work in SQLPlus.

Each Tab in Golden (editor 'page') manages its own list of prompt variables. You can use the 'Variables' submenu on the 'Script' menu to edit and clear the current tab's prompt variables.

In a script, a prompt variable that starts with a single & character is a temporary prompt variable. You will be prompted for its value each time it is seen by the parser. Using the && character before a prompt variable makes it permanent. You will no longer be prompted for its value even if it appears later with a single &.

Here's an example of a statment with a temporary prompt variable:
```
select * from emp where ename like '&Lastname';
select * from emp where ename like '&Lastname';
```

When you run this statement in Golden, you will be prompted to enter a value for &Lastname twice (once for each statement)

Here's an example of a statment with a single prompt variable:
```
select * from emp where ename like '&&Lastname';
select * from emp where ename like '&Lastname';
```

Here you will only be prompted for Lastname once and that value will be used for both statements.

You can define a prompt variable in script using the DEFINE command:
```
define lastname = "FORD";
select * from emp where ename like '&lastname';
```
Note that the define command acts like a && prompt variable (it is permanent and you won't get prompted for that variable again.

You can undefine a prompt variable using the UNDEFINE command:
```
undefine lastname;
```

4.12 Use Bind Variables

Bind variables allow your scripts to be more interactive and can be very useful in calling pl/sql stored procedures. Bind variables are variables that can be defined and given a value which will be passed to the server and processed by the database. Bind variables are created using the VAR command and used in statements by placing a colon : in front of the Bind Variable name. Note that each Script tab in Golden manages its own Bind variables (they aren't shared among tabs.)

The var command syntax to create a bind variable is:

var varname vartype
The choices for vartype are:
INTEGER: In Oracle this is a NUMBER(38,0) type.
NUMBER: This is a floating point value.
STRING: A string of 2000 (4000 for Oracle 8+) max characters.
PLSQLSTRING: A string of 32K characters (varchar size in pl/sql)
REFCURSOR: A referenced cursor (see example below.)

Note that if you use string or plsqlstring, Oracle will convert your variable to the appropriate type (except REFCURSOR) automatically.
Another Note: Golden will autodefine any bind variables it sees in your statements as string type and give them an initial value of "" if they don't already exist.

The var command syntax to assign a value to a bind variable is:
var varname = "Data" (remove the quotes if it isn't a string.)

The var command for clearing all bind variables (you can also use the 'Clear all bind variables in this tab' menu choice Script | Variables) is:
var clearall

To see all the defined bind variables use:
var
or
print

To see a single bind variables use:
var varname

Note that Golden displays one REFCURSOR at a time in the spreadsheet results area. You can switch between REFCURSORs by using the 'Results | Bind Variable Cursors' menu. The REFCURSOR will be displayed in the spreadsheet window. All other bind variables are displayed in the text view window.

Here is an example of a very simple script using bind variables:

```
REM First we define some bind variables.
var MyVar1 string
var MyVar2 string
var MyVar3 number
var MyVar4 integer
REM This next line allows you to set a value for a bind variable.
var MyVar2="Software";
REM Now we run a pl/sql block with the bind variables in it.
begin
  :MyVar1 := 'Benthic ' || :MyVar2;
  :MyVar3 := 3*1.4;
  :MyVar4 := 5+6.2;
end;
/
print
```

And the result of running this script is:

```
-- 01/04/2000 08:24:30 pm
MYVAR1 STRING = Benthic Software
MYVAR2 STRING = Software
MYVAR3 NUMBER = 4.2
MYVAR4 INTEGER = 11
```

Here is an example of a very simple EXEC call using bind variables:

```
var arg1 string
var arg2 string
var arg3 string
var arg4 string
var ret1 string
var arg1 = "Gort, ";
var arg2 = "Klaatu ";
var arg3 = "barada ";
var arg4 = "nikto";
exec :ret1 := MyConcatFunction( :arg1, :arg2, :arg3, :arg4 );
print ret1
```

And the result of running this script is:

```
 -- 01/04/2000 08:45:07 pm
 RET1 STRING = Gort, Klaatu barada nikto
```

Using and Displaying REFCURSOR bind variables

Refcursor bind variables work the same way as other bind variables except that you can not give them an initial value and they are displayed in the spreadsheet window. Each REFCURSOR containing data is shown on the 'Results' menu on the 'Bind Variable Cursors' submenu. Just select a REFCURSOR to see the data in the spreadsheet.

Note that if you call your REFCURSOR bind variable "cursor", you do not need to declare it with a VAR statement.

The example uses the following package and function:

```
create or replace package types
as
   type cursorType is ref cursor;
end;
/
create or replace function GetMyTablesCursor return types.cursortype
as
   l_cursor    types.cursorType;
begin
   open l_cursor for select * from tab;
   return l_cursor;
end;
/
```

Here is the example:

```
var MyCursor refcursor
exec :MyCursor := GetMyTablesCursor;
```

Or you could just use:

```
exec :cursor := GetMyTablesCursor;
```

4.13 Use Workspaces

Workspaces allow you to save the current tab information to a file. The workspace can then be opened, restoring your tabs and scripts.

What is saved:

The 'Save Workspace files as links' options setting (on the General tab) specifies how tabs that contain files are saved. If the option is checked than tabs that have filenames associated with them only save the filename to the workspace. When the workspace is loaded, linked filenames are found and their text is loaded.

Setting tab names:

Tab names default to 'QueryN' or the current filename loaded in the tab. To change a tab's name, make the tab the current tab by clicking on it. Then right click the tab area to bring up a window where the tab can be changed. Tab names are only saved if you then save the workspace.

Moving Tabs:

Tabs can be dragged to a new position using the mouse. You can also use the "Change tab order or names" item on the "View" menu.

4.14 Use Popup Lists

A Popup List is a list of tables, columns and other lists of items designed to help you type SQL commands. It's like a popup version of the SQLBuilder. The popup list can be "popped up" by pressing Ctrl-Space. The popup list can also be set to automatically pop up when you type certain characters. This option is on the "SQL Editor" page of the Options window.

. New: Popup Lists for database objects.
'.' trigger for schemas, tables and columns
Pressing '.' after an alias or table name will bring up
a popup list of columns for that object.
Pressing '.' after a schema name will bring up the list of
tables for that schema.

'(' trigger for procedure arguments
Pressing '(' after a stored procedure/function/package method name
will bring up a popup displaying the arguments for that method. Overloaded
package methods will show multiple argument lists.
This popup closes when you type the closing ')' character. Note that
built in system functions do not currently have argument popup lists.

Popup List Filtering:
Typing characters while a popup list is displayed will 'filter' the popup
list to just those items that match the entered letters. Use the backspace
key to remove filter characters. The selected item can be changed by using
the up and down arrow keys. Pressing 'Enter' or any punctuation or whitespace
character will accept the selected item. Pressing Escape will close the popup
list.

Popup List Cycling:
When the popup list is shown the left and right arrow keys
cycle the popup list type: schema <--> table <--> columns. Changing
the Schema and pressing the right arrow will go to that schema's table
list and pressing the right arrow on the table list will show the selected
table's column list.

The popup list type can be changed by clicking the list type button
at the bottom of the popup list.

Use Ctrl-Space to bring up the table list popup at any time.

Alias Handling:
Aliases can be setup by either using the 'Alias List...' item on the 'Script'
menu or by right clicking an object in the SQLBuilder and choosing 'Add Alias...'
Note that alias lists are specific to your login schema.

New Options for Auto Popup Lists on the Editor page of the options window:
Disable Auto Popup Lists: To turn off triggering of popup lists from the '.' and '('
Note that Ctrl-Space still brings up the lists.
Auto Popup Delay: milliseconds between trigger character and popup. During this time
pressing any other key aborts the popup.

4.15 Export Data to files or Excel/OpenCalc

Query results can be saved to various file formats or directly export to an Excel or OpenCalc (Open Office) spreadsheet.

Saving to CSV:

This will save the current data in CSV format. The fields will be comma separated and fields will be double-quote qualified if needed. Data may contain commas (in which case the field will be qualified with double-quotes).

Exporting to Excel or OpenOffice Calc:

This will export directly to Excel or Calc. Note that if Excel is busy when the export is started you may see an error and have to try again when it isn't busy. The EXPORT command (described below) allows you to export directly from a script.

Using the EXPORT command in scripts:

See: Use the Export command

4.16 Create and use the favorites menu

On the "Menus" page of the Options Window, is a field called "Favorites Menu Root" If this field is set to a disk directory that contains *.sql files, a Favorites menu will appear on the main window. This menu will contain items for each of your .sql files and also will contain folders for any subdirectories that have .sql files.

For items in your Favorites menu, you can:

Click item: Will load the file into the active sql editor window and run it. Be careful when autocommit is set! I suggest only using queries on your favorites menu!

Shift-Click item: Will load the file into the active sql editor window, but will not run it.

Control-Click item: Will prompt you to delete the .sql file for this item.

For example:

I have the following directory structure:

C:\Benthic\ <-- Contains Golden32.exe
C:\Benthic\SQL\
File: Show Tabs.sql
File: Show Indexes.sql
C:\Benthic\SQL\Patient Queries\
File: Show Patients with Hypertension.sql

If my "Favorites Menu Root" field is set as "C:\Benthic\SQL" (without the quotes), My favorites menu will contains the Items "Show Tabs" and "Show Indexes". It contains a subfolder called "Patient Queries" with one item, "Show Patients with Hypertension"

If the favorites menu doesn't seem to work:

A minimum test would be to create a .sql file called test.sql that contains the text "select * from tab". Place it into the same directory as Golden32.exe and set the Favorites Menu Root to "." (without the quotes.) Your favorites menu should then contain "Refresh Favorites" and "test"

5 Help and Tips

Please select an option below:

5.1 Connection/Login Problems

Assuming that the database and network is in working order, the most common cause of connection problems is a damaged Oracle Client installation. In many cases, re-installing the Oracle Client and checking the database aliases in tnsnames.ora will solve the problem. Benthic products try to load the most current version of the Oracle Client that is installed on your system. It does this by searching for the OCI (Oracle Call Interface) .dll file on your machine. It is possible that there is an 'orphaned' OCI file that is causing the problem. Re-installing can solve the problem as Oracle scans for OCI dll files during the install and can remove them. If you are still having a problem, you can enter an oci dll filename on the Login Options Window to fix the problem. Of course, if you are still having problems, please contact Benthic Software and we will be glad to help you! Also look at our web site's support page!

5.2 Quick Tips

Use sql to make sql:
It can be useful to use an sql query to generate other sql statements. This is often useful in maintainance.
Enter an sql query like

```
select
    'alter ' || object_type || ' ' || object_name || ' compile;'
from
    user_objects
where
    object_type = 'PROCEDURE' and status = 'INVALID';
```

Execute it and cut the result set from the spreadsheet. Paste it into a query window and execute it.

Script delimiters:

Separate SQL statements with a semicolon and Enter (;<crlf>).

Example:

```
delete * from emp where ename like 'F%';
select * from emp;
```

If you have pl/sql in your scripts, separate statements with <crlf>/<crlf>.

Example:

```
delete * from emp where ename like 'F%';
begin
    -- Kind of a useless example, but hopefully you get the idea!

    insert into emp select * from emp2;
end;
/
select * from emp;
```

Comments:

In scripts, you can use the REM comment. Inside sql statements or pl/sql use -- for single line comments or /* */ for block comments.

5.3 Keyboard shortcuts

Application Help

Display Help Contents	F1

Tab and File Controls

New Tab	Ctrl+N
Clear Current Tab	Shift+Ctrl+N
New Tab with Private Login	Shift+Ctrl+Alt+N
Open File Into New Tab	Ctrl+O
Open File Into Current Tab	Shift+Ctrl+O
Open File Into New Tab	Shift+Ctrl+Alt+O
Save File	Ctrl+S
Save File As...	Shift-Ctrl-S
Close Tab	Ctrl+F4
Close All Tabs	Shift+Ctrl+F4
Close Program	Alt+F4
Goto Next Tab	Ctrl+Tab
Goto Prior Tab	Shift+Ctrl+Tab
Goto Tab	Alt+# (# = keys 1 thru 0 for tabs 1 thru 10)

Workspace

Open Workspace	Ctrl+W
Save Workspace	Shift+Ctrl+W

Editing Keys

Standard Editing:

Undo	Ctrl+Z
Redo	Shift+Ctrl+Z
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Select All	Ctrl+A

Find & Replace:

Find	Ctrl+F
Find Next	F3
Replace	Ctrl+H

Bookmarks and Navigation:

Top of script	Ctrl+Home
Bottom of script	Ctrl+End
Set a bookmark within a script	Shift+Ctrl+# (0-9)
Goto a bookmark	Ctrl+# (0-9)
Toggle Between Beginning Of Line And First Non-Whitespace Character	Home

Text Block Commands:

Block Indent	Ctrl+I or Tab when selection
Block Unindent	Ctrl+U or Shift+Tab when selection
Comment Out Selected Lines	Ctrl+- (Ctrl + Dash)
Uncomment out selected Lines	Shift+Ctrl+- (Shift+Ctrl+Dash)

Find Matching brackets:

Highlight inside brackets or quotes	Ctrl+B
Find previous matching ({[<	Ctrl+[(Place caret on bracket first)
Find next matching ({[<	Ctrl+] (Place caret on bracket first)

Miscellaneous:

Toggle case of selection	Ctrl+T

Oracle Session

Login	Ctrl+L or Ctrl+J

Script Execution

Run Script	F5 or Shift+Enter while typing.
Run Script From Cursor	F6
Run One Statement At Cursor	F7 or Ctrl+Enter while typing or Ctrl+RightClick.
Run selected text	Ctrl+F7
Run Script And Go To Edit Mode	Ctrl+E
Run single statement at cursor in edit mode	Shift+Ctrl+E
Commit	Ctrl+F5
Rollback	Ctrl+F6
Show Execution Plan	Ctrl+P

Clipboard Extensions

Copy SQL To Clipboard With Language Formatting	Shft+Ctrl+C
Paste SQL From Clipboard Stripping Language Formatting	Shft+Ctrl+V

Window Controls

Toggle Between Edit And Results Windows	Ctrl+R or F8
Toggle SQL Builder	F9
Toggle DBMS Output Window	F10
Toggle Scratch Results Window	F11
Toggle Log View	F12

5.4 Script Commands

This is a list of the script commands that Golden supports. Note that many of these work similarly to the SQLPlus commands.

ACCEPT:

Used to pre-prompt for a "prompt variable" (instead of the standard &,&& statement prompting).

ACC[EPT] variablename [DEF[AULT] default] [PROMPT text] [HIDE]

DEFAULT:	Specified a default value for the variable.
PROMPT:	Specifies the prompt message displayed to the user.
HIDE:	Hides the value entered as if it was a password prompt.

Example:
accept empno default 5 prompt Please enter a value for EMPNO:

BEEP:

Plays the system beep sound and causes the taskbar button to highlight. In general used at the end of long running scripts or statements to indicate that the work is complete.

Example:
select ... long running query;
beep

CLEAR:

Used to clear the tabs COLUMN defines, the datagrid or the log window.

CL[EAR] COL[UMNS] | SCR[EEN] | LOG

COLUMNS:	Clears any defined columns (see COLUMNS below).
SCREEN:	Clears the datagrid.
LOG:	Clears the log window.

COLUMN:

Used to set the formats or display attributes of columns by column name as well as other advanced actions. Settings are specific to the current tab.

COL[UMN] [column [option ...]]

Where option represents one of the following clauses:

CLE[AR]
Clears the options for that column.

FOR[MAT] format
Sets a format for the column. Only date and number columns can be formatted. The format string consists of standard formatting characters.

Example:
column fish format '9,990.00';
select 12345.60 fish from dual;
displays: 12,345.60

HEA[DING] text
Sets the heading text for that column.

JUS[TIFY] {L[EFT] | C[ENTER] | R[IGHT]}
Set the justification for that column.

NEW_V[ALUE] variable
Used to store the value of a column into a prompt variable. Golden can also do this using the Define command but it is not SQLPlus script compatible.

NUL[L] text
Used to specify text to show if the value is null. This overrides Golden's default null indicator value.

ON | OFF
Used to turn off or on this column's settings.

CONNECT:
Used to connect to your database or change the current login. This command will convert a public tab (one that shares the main program's login connection) into a private one (a private tab has a separate login to the database). If the password part of the login is left blank, you will be prompted for it. In general passwords shouldn't be included in saved script files for security reasons.

CONN[ECT] [{logon | / | proxy} [AS {SYSOPER | SYSDBA}]]

DEFINE:
Used to define a prompt variable and give it a value. When a variable is defined, you will not be prompted for the value if &name appears in an sql statement as the defined value will be used instead. Note that a bind variable can be used as the VALUE (i.e. define myval = :mybindvar).

DEF[INE] NAME = VALUE

Example:

define mypromptvar = testing
select '&mypromptvar' from dual;

DESCRIBE:
Used to show table/view structure, function parameters, or package information. Note that this information can also be seen in Golden's SQLBuilder pane.

DESC[RIBE] [schema.]object[@db_link]

Also see How to use the DESCRIBE command.

DISCONNECT:
Disconnects (logs out) a private tab from the database.

EXECUTE:
Used to run a stored procedure.

Please see How to use the EXECUTE command.

EXPORT:
Used to export the current datagrid data to an external file or program. Please see How to use the EXPORT command.

PAUSE:
Shows the user a message, pausing the script, and allows them to cancel (which will stop the script at the current point.)

PAUSE Textmessage

PRINT:
This statement will display the value of a bind variable or if no bind variable name is given, all the current bind variables.

PRINT [BINDVARNAME]

PROMPT:
Shows the user a message, pausing the script, and allows them to cancel (which will stop the script at the current point.)

PROMPT Textmessage

SET:
Used to set properties that affect various aspects of the program.

DEFINE | SCAN ON | OFF
Turns prompt (&) scanning on or off.

ECHO ON | OFF
HEADING ON | OFF
FEEDBACK ON | OFF
These settings (and SERVEROUTPUT listed below) affect the spool file. Please see How to use the spool command.

RECORDLIMIT #
Sets a limit on the number of records returned from queries. Use -1 for unlimited.

ERRORS ON | OFF
Turns script errors on or off.

SERVEROUTPUT ON | OFF [SIZE N]
Determines if PL/SQL buffer output appears in the spool file. Size is in bytes. The default size is used if the size option isn't used.

SPOOL:
Used to export script data and/or statements to a file. Please see How to use the SPOOL command.

START (@):
Used to run an external script file.

@"C:\MyScripts\mysqlfile.sql"

UNDEFINE:
Deletes a prompt variable. The variable will be prompted for the next time it is used in a

statement.

UNDEF[INE] NAME

VARIABLE:

Declares a bind variable that can be used in an Exec call or PL/SQL block. Please see How to use bind variables.

VAR[IABLE] [variable [type]]

Index

- & -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- K -

- L -

- M -

- O -

- P -

- Q -

- R -

- S -

- T -

- U -

- V -