SQL Direct User Guide

AVEVA Solutions Ltd

Disclaimer

Information of a technical nature, and particulars of the product and its use, is given by AVEVA Solutions Ltd and its subsidiaries without warranty. AVEVA Solutions Ltd and its subsidiaries disclaim any and all warranties and conditions, expressed or implied, to the fullest extent permitted by law.

Neither the author nor AVEVA Solutions Ltd, or any of its subsidiaries, shall be liable to any person or entity for any actions, claims, loss or damage arising from the use or possession of any information, particulars, or errors in this publication, or any incorrect use of the product, whatsoever.

Copyright

Copyright and all other intellectual property rights in this manual and the associated software, and every part of it (including source code, object code, any data contained in it, the manual and any other documentation supplied with it) belongs to AVEVA Solutions Ltd or its subsidiaries.

All other rights are reserved to AVEVA Solutions Ltd and its subsidiaries. The information contained in this document is commercially sensitive, and shall not be copied, reproduced, stored in a retrieval system, or transmitted without the prior written permission of AVEVA Solutions Ltd. Where such permission is granted, it expressly requires that this Disclaimer and Copyright notice is prominently displayed at the beginning of every copy that is made.

The manual and associated documentation may not be adapted, reproduced, or copied, in any material or electronic form, without the prior written permission of AVEVA Solutions Ltd. The user may also not reverse engineer, decompile, copy, or adapt the associated software. Neither the whole, nor part of the product described in this publication may be incorporated into any third-party software, product, machine, or system without the prior written permission of AVEVA Solutions Ltd, save as permitted by law. Any such unauthorised action is strictly prohibited, and may give rise to civil liabilities and criminal prosecution.

The AVEVA products described in this guide are to be installed and operated strictly in accordance with the terms and conditions of the respective licence agreements, and in accordance with the relevant User Documentation. Unauthorised or unlicensed use of the product is strictly prohibited.

First published September 2007

© AVEVA Solutions Ltd, and its subsidiaries 2007

AVEVA Solutions Ltd, High Cross, Madingley Road, Cambridge, CB3 0HB, United Kingdom

Trademarks

AVEVA and Tribon are registered trademarks of AVEVA Solutions Ltd or its subsidiaries. Unauthorised use of the AVEVA or Tribon trademarks is strictly forbidden.

AVEVA product names are trademarks or registered trademarks of AVEVA Solutions Ltd or its subsidiaries, registered in the UK, Europe and other countries (worldwide).

The copyright, trade mark rights, or other intellectual property rights in any other product, its name or logo belongs to its respective owner.

SQL Direct User Guide

Contents   **Page**

SQL Direct User Guide

1 About this Guide

1.1 Scope of this Guide

This guide tells you how to use AVEVA SQL Direct on a PC running Windows XP.

For information about training courses, see AVEVA's worldwide sales and support offices, see the Customer Support pages on our website at www.aveva.com/location.

1.2 About this User Guide

SQL Direct Components	Describes the components of SQL Direct - SQL Direct Configuration, the OUTFITTING Provider, the Table Designer and some potential data consumers. It describes what each component does and how they all fit together to form SQL Direct.
Using SQL Direct Components	Tells you how to use SQL Direct Configuration, the OUTFITTING Provider, and the Table Designer.
Using SQL Direct - Some Examples	Gives some examples of how to use SQL Direct with simple data consumers.
Using SQL Direct with Microsoft SQL Server	Describes how to use SQL Direct with Microsoft SQL Server / SQL Desktop Engine (MSDE) and Microsoft Access.
Some Technical Details	Describes some of the technical details of SQL Direct. This chapter is likely to be useful chiefly for OUTFITTING Administrators.

Note: This User Guide assumes that you are familiar with OUTFITTING DESIGN.

1.3   Some Useful WWW Addresses

Site	URL
Microsoft Data Access and Storage web site:	http://msdn2.microsoft.com/en-us/data/default.aspx
Microsoft Developer Network web site:	http://msdn.microsoft.com/default.asp
Microsoft SQL Server web site:	http://www.microsoft.com/sql/

1.4   Reference Material

System Administrators may find the following useful:

- *Microsoft OLE DB 2.0 Programmer's Reference and Data Access SDK* (Microsoft Professional Editions). Microsoft Corp (Editor); paperback. Microsoft Press; ISBN: 0735605904; November 1998.

- *Teach Yourself OLE DB and ADO in 21 Days (Teach Yourself).* John W. Fronckowiak; paperback. Sams; ISBN: 067231083X; August 1997.

- *Learn OLE DB Development with Visual C++ 6.0.* Nathan Wallace; paperback. Wordware Publishing; ISBN: 1556226349.

- *Learn Advanced OLE DB Development with Visual C++ 6.0.* Nathan Wallace; paperback. Wordware Publishing; ISBN: 1556226586.

- *Visual C++ 6.0 OLE DB Programming.* Craig McQueen; paperback. Wrox Press; ISBN: 1861001614

2   Introducing SQL Direct 12.0

SQL Direct enables you to access the OUTFITTING DESIGN Database directly from within software products that support Microsoft Universal Data Access interfaces such as ODBC, OLE DB and ADO.

Products that support these interfaces include the Microsoft Office applications Word, Excel and Access, other proprietary databases such as ORACLE and SYBASE, end-user applications written in Visual Basic and Visual C++, and of course SQL Direct.   We collectively refer to these products as data consumers or data providers, according to whether they consume or provide data access services (irrespective of whether data is read from or written to any particular database). Some products can act both as consumers and providers.

Using SQL Direct tools, you can query any data that the OUTFITTING DESIGN module can access i.e. attributes, pseudo-attributes, and UDAs.

The following diagram gives an overview of the software components that make up SQL Direct (shown in pale blue) in relation to data consumer applications and OUTFITTING.
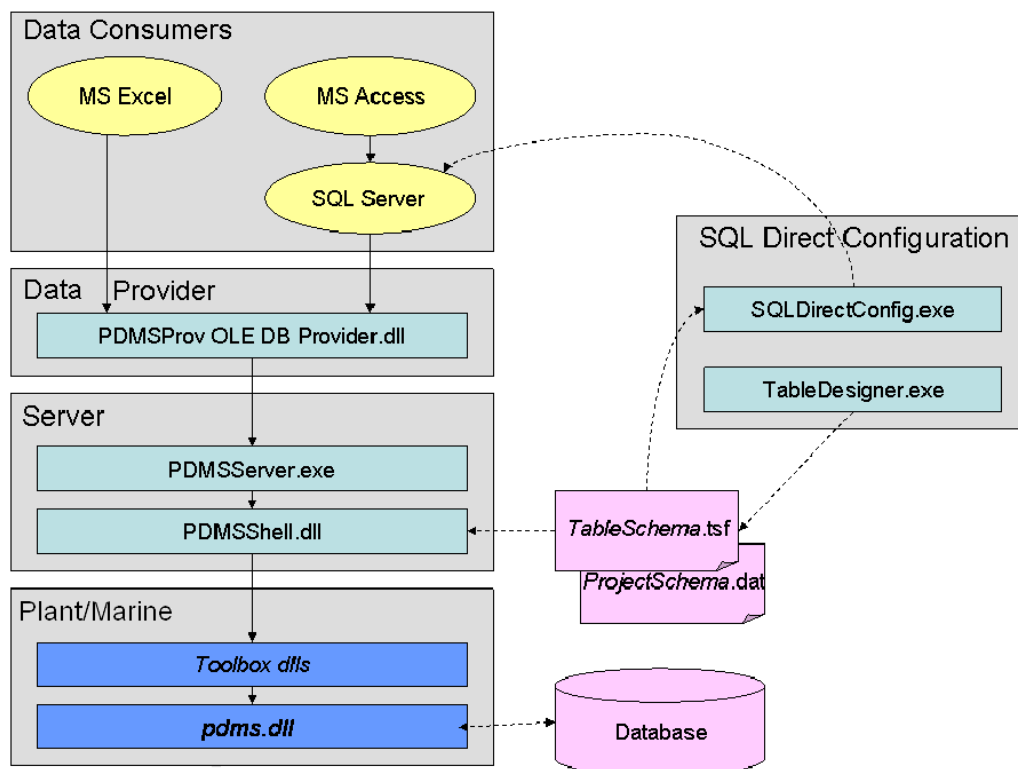
Although not necessary to access data through OLE DB and ADO, you can enhance SQL Direct using Microsoft SQL Server (or Microsoft SQL Desktop Engine (MSDE), which is an equivalent but earlier product).

SQL Server provides access to additional functionality by acting as a data consumer and then providing a relational view of the data to other applications, for example Microsoft Access, through its "linked server" facility. This mode of data access can be further wrapped as an ODBC data source.

We have tested SQL Direct with MSDE 2000, Service Pack 4. However, throughout the rest of this document we will refer to Mirosoft SQL Server, or simply SQL Server.

2.1   Using SQL Direct - Setting Up

To use SQL Direct 12.0 you must have AVEVA 12.0 or later installed, as well as have some project data.

If you want to use the entended functionality based on SQL Server, you will also need to have this product installed.

Then, you need to:

- **configure** SQL Direct with the necessary paths and environment variables to match the particular project database or databases you wish to access.
- **design** a set of tables that defines the structure of the data to be passed to the data consumer. SQL Direct comes with some preconfigured tables, although you may also design your own.

If you want to use SQL Server, you also need to:

- **configure** SQL Server with the necessary data views to match the tables you have defined.

See *SQL Direct Components*, *Using SQL Direct Components* and *Using SQL Direct - Some Examples* for more details of these operations.

3 SQL Direct Components

There are two components of SQL Direct that are visible to the user - *SQL Direct Configuration* and the *Table Designer*. These are both related to initial set-up.

Normal day to day operation of SQL Direct proceeds, invisibly to the user, through the OUTFITTING Provider component and the OUTFITTING Server which interacts with OUTFITTING itself.

3.1 SQL Direct Configuration

3.1.1 Introduction to SQL Direct Configuration

Start SQL Direct Configuration from the **Start** menu, ![start], by selecting the program from the AVEVA entry:

The main SQL Direct Configuration window looks like this:
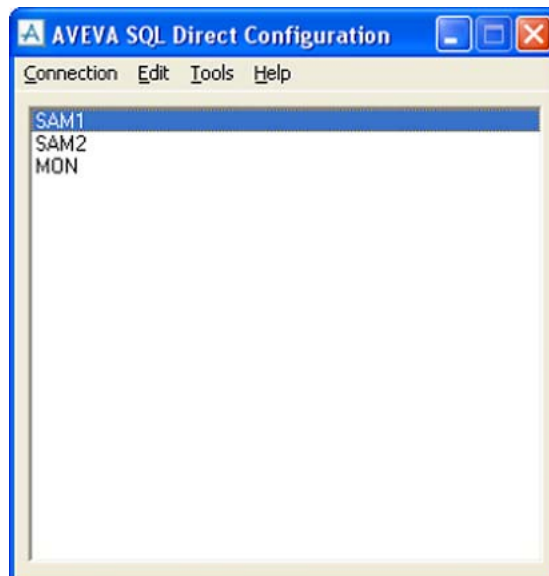
SQL Direct Configuration allows you to configure SQL Direct with connection information to match the particular project databases you want to access. Using the tools menu, you can also start the Table Designer, start and stop the OUTFITTING Server and SQL Server processes, and configure data views within SQL Server.

3.1.2 SQL Direct Configuration - Menu Options

SQL Direct Configuration has three function menus plus the help menu. The options available are described in the following sections.
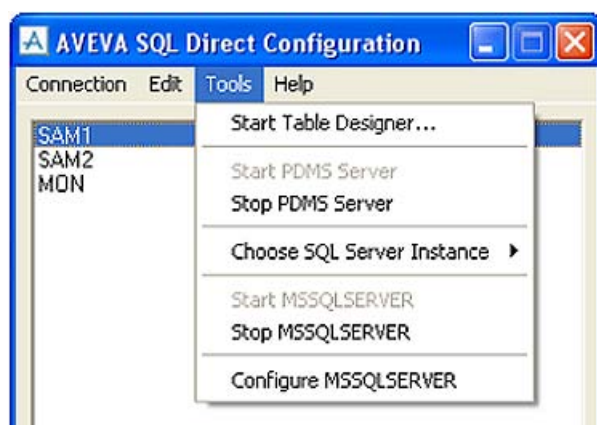
The Connection Menu
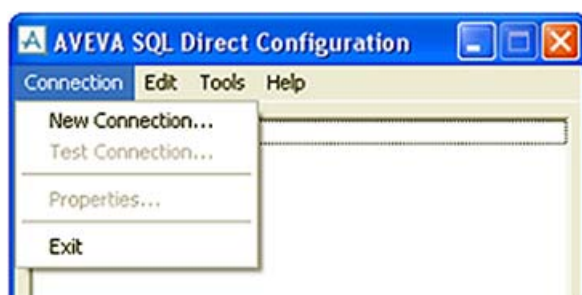
| Menu Option	Function
New Connection…	Allows you to add a new set of connection details. This invokes an empty **Connection Properties** form.
Test Connection…	Tests connection to OUTFITTING using the currently selected set of connection details. (A simple ADO based test program is provided for this purpose.)
Properties…	Invokes the **Connection Properties** form for the currently selected connection.
Exit	Exits the SQL Direct Configuration program.

Edit Menu

Menu Option	Function
Cut	Removes the currently selected set of connection details from the list. The details are retained in the paste buffer.
Copy	Copies the currently selected set of connection details into the paste buffer.
Paste	Replaces the currently selected set of connection details with the set from the paste buffer.

Menu Option	Function
Paste New	Adds a new set of connection details populated with the settings from the paste buffer.
Delete	Removes the currently selected set of connection details from the list. They are not retained in the paste buffer.

Tools Menu

Menu Option	Function
Start Table Designer…	Starts the Table Designer program.
Start OUTFITTING Server	Starts the OUTFITTING Server process.
Stop OUTFITTING Server	Stops the OUTFITTING Server process.
Choose SQL Server Instance ->	Allows you to choose the SQL Server instance on your machine (if more than one) that the Start, Stop and Configure options will apply to.
Start SQLEXPRESS	Starts the SQL Server process (for the chosen SQL Server instance).
Stop SQLEXPRESS	Stops the SQL Server process (for the chosen SQL Server instance).
Configure SQLEXPRESS	Creates the necessary metadata within the chosen SQL Server instance to provide data views consistent with the table schema file defined in the currently selected connection details.

3.2 Table Designer
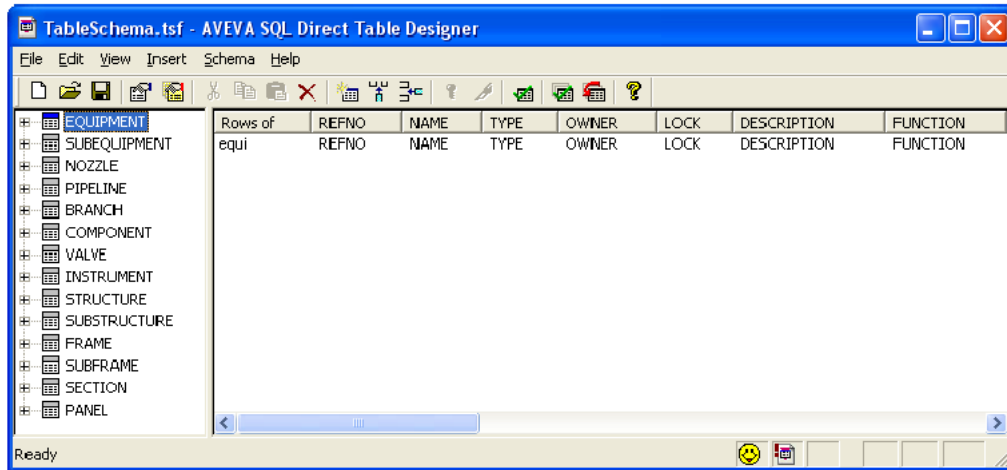
The Table Designer enables you to define tables of OUTFITTING data. Within each table, you can define one or more row types and column names, and the data item to fetch for each. Once you have defined and saved your table definitions, SQL Direct can use them to determine which data to pass to a data consumer that requests information from the database.

See *Using the Table Designer* for full details

3.3 OUTFITTING Provider

The software component that coordinates the provision of data to a data consumer (i.e. a client application) is called the data provider.

During normal, day-to-day operation of SQL Direct, the OUTFITTING data provider works invisibly, automatically forming data rowsets in response to data requests from client programs.

The underlying technology that supports the interaction between data providers and data consumers is called Object Linking and Embedding Database (OLE DB), and data providers and consumers are more fully known as OLE DB Providers and OLE DB Consumers.

3.4 OUTFITTING Server

All data access by SQL Direct to and from OUTFITTING is handled by a program called the OUTFITTING Server. This program is started up automatically as required by the OUTFITTING Provider, and also Table Designer.

OUTFITTING Server uses the connection details defined in SQL Direct Configuration to link to named projects and MDBs.

Login to OUTFITTING uses normal user names and passwords, and access levels are determined by the access levels granted to that user.

When a client first connects to OUTFITTING Server, these three pieces of connection information must be established:

- the connection identity (connection string)

- the user name
- the password.

For convenience you can store this information in a client application's cache or save file. Microsoft Office clients, for example, offer the opportunity to store this information for future access. If a client application supplies this information, operation of OUTFITTING Server is completely invisible.

Note: You should be aware that retaining passwords in client caches weakens data security.

If security is important then you should not retain passwords in client caches. When a client connects to OUTFITTING Server without full connection information, OUTFITTING Server will invite you to provide the missing information on a login form:

After login, the operation of OUTFITTING Server is normally automatic and requires no user intervention. OUTFITTING Server will run until you log out of Windows, or shut down your machine. However, you can also start and stop OUTFITTING Server manually through the Tools menu of SQL Direct Configuration.

You can also start OUTFITTING Server through the shortcut on the SQL Direct program menu. In this case, a Command window is displayed once OUTFITTING has been connected to, providing diagnostic output of commands as they are executed by OUTFITTING Server.

3.5   Use on Shared PCs

SQL Direct is a single user tool. Once you have logged in to OUTFITTING through any of the SQL Direct tools, it is not possible for another user to log in during the same session, or for you to access OUTFITTING with a different connection string from the one you logged in with. Other login details will be rejected with "access denied" errors.

To access SQL Direct with new user credentials, you must first shutdown OUTFITTING Server, either through the SQL Direct Configuration Tools menu, or by closing the "OUTFITTING Command Line" window (if visible), or by executing the following command from a command prompt window:

<SQL Direct install path>OUTFITTING Server /STOP

Once OUTFITTING Server has been shut down, you can either restart OUTFITTING Server through the SQL Direct Configuration Tools menu, or allow it to start up automatically as required.

4 Using SQL Direct Components

4.1 Using SQL Direct Configuration

This section describes the SQL Direct Configuration menu options. For an introduction to and a general description of the SQL Direct Configuration, see *SQL Direct Configuration*.

4.1.1 Specifying OUTFITTING Connection Details

Before OLE DB clients can use the services of the OUTFITTING Provider, you must use SQL Direct Configuration to set up one or more sets of connection details.

A set of connection details is identified by a connection string identifying the project and MDB it will open for the connection, and a number of environment variable settings needed for OUTFITTING to operate correctly.

You can input completely new sets of connection details by selecting the **Connection>New Connection…** menu option.

You can view or modify an existing set by selecting the **Connection>Properties…** option.

Either option will invoke the **Connection Properties** form:

The fields on this form have the following meaning:

Field	Purpose
Connection String	A unique identifier for this set of connection details. This is the connection string to quote when setting up a OUTFITTING Provider link from a client.
Standard Variables	These are all mandatory. Set **PDMSEXE** to point to the install directory for OUTFITTING. Set **PDMSWK** to a suitable OUTFITTING working directory. Set **PDMSUI** to where you store PML form definitions and related code Set **PMLLIB** to where you store the files for general PML application ware. You may use the **Browse...** button to locate these directories and reduce typing.

Field	Purpose
Project Details	These are all mandatory. **Project Name** is the OUTFITTING project name. **MDB** is the OUTFITTING MDB that should be opened on connection. **Table Schema** is the location of the file containing the definition of the tables exported from this OUTFITTING project through OUTFITTING Provider. You create and modify table schema files with the Table Designer program. You may use the **Browse...** button to locate this file.
Database Variables	These tell OUTFITTING where the database files are for your project. Note that the labels reflect the name of the project: so if the project is "SAM" the labels are **SAM000**, **SAMMAC**, etc. You need to fill in only these: the **PIC** and **ISO** fields are for future use and you can leave them blank. **More Env Vars** gives you the opportunity to define any further environment variable setting required for this OUTFITTING project, for example foreign database locations. The field contains the name of a file holding further environment variable definitions, one per line, in the format: **SET <varName>=<value>.** You may use the **Browse...** button to locate these locations.

When editing with this form, the **OK** and **Apply** buttons allow you to save the changes. The **OK** button additionally dismisses the form; and you can discard changes using the **Cancel** button.

You can test to see if the connection details are correct using the **Test Connection** button. This launches a separate program, **ADOTest**, to test the connection as a Microsoft Active Data Object (ADO) client.

4.2 Using the Table Designer

4.2.1 Starting the Table Designer

You can start the Table Designer in one of two ways:

- Select **Start>All Programs>AVEVA>SQL-Direct 12.0>Table Design for** SQL Direct from the **Start** menu.
- Choose **Tools >Start Table Designer...** from the SQL Direct Configuration menu.

Whichever route you choose you will need to complete the login details for the project you wish to connect to (unless OUTFITTING Server is already running and logged in to OUTFITTING with the correct details):

Field	Purpose
User Name	Your user name.
Password	Your password.
Connect String	The set of connection details to use.

When a *Table Schema* file already exists for this connection, Table Designer displays the main form for that table schema (see below). If no table schema file already exists the main window is empty:
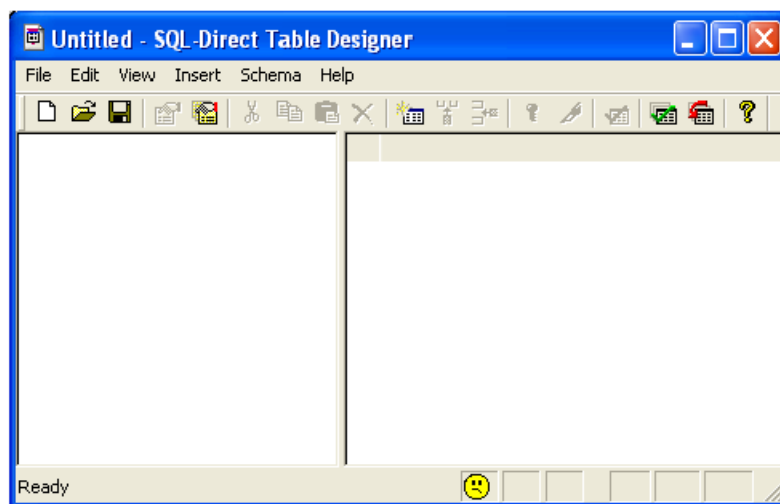
4.2.2   Basic Concepts

The Table Designer opens showing the current *table schema*, if there is one. The table schema maps to the current project database. SQL Direct uses table schemas to define which data to pass to a data consumer when requested.

You can use the Table Designer to add new tables or modify or delete existing ones.

Some Definitions

Term	Definition
Project Schema file	This file lists all the element types available in a project along with their attributes. All Table Schema files (see below) refer to a Project Schema file to map the columns in a table to element attributes. In general, one Project Schema file can serve many Table Schema files.
Table Schema file	This file describes the tables used to retrieve data. Different table schemas may be appropriate for different reporting disciplines.
Table	A logical grouping of data. A table consists of columns (attribute values) and table rows (element types). When data is retrieved, the OUTFITTING Provider searches for all elements of the types listed in the table rows and extracts the value of the attribute specified in the appropriate table row for each column.
Column	Each column in the table describes one element attribute whose value is to be returned when the table is queried. The column name is often the same as the attribute name, but does not have to be.
Table row	A table row describes what sort of data are to be returned. Each table row in a table specifies an element type and a list of *bindings*, one for each column. A table contains one or more table rows. In this document, "row" usually means "table row".
Data row	A row of data returned from the database corresponding to a single element. The values returned are determined by the entries in the table rows for that element type for the table.
Binding	A mapping in a table between the column and an attribute of an element. Each table row consists of an element type and a binding for each column of the table describing which attribute of the element supplies the data for that column.
Key column	A column in the table that uniquely identifies an object in the database. A table must have a key column before any other column can be made writeable. Practically, the key column must be either REFNO or NAME (REFNO is preferred). Key columns are always read-only.

4.2.3   Creating a New Table Schema

To create a new table schema from scratch:

1. Generate a *Project Schema* file if one does not already exist using the **Schema>Generate Project Schema**… menu option.

2. This procedure extracts the element type and attribute information from OUTFITTING. Project Schemas are project specific since different OUTFITTING projects will typically define different User Defined Attributes (UDAs).

3. Select the **File>New** menu option or click the ☐ button on the toolbar and select the Project Schema file from the dialogue screen displayed:
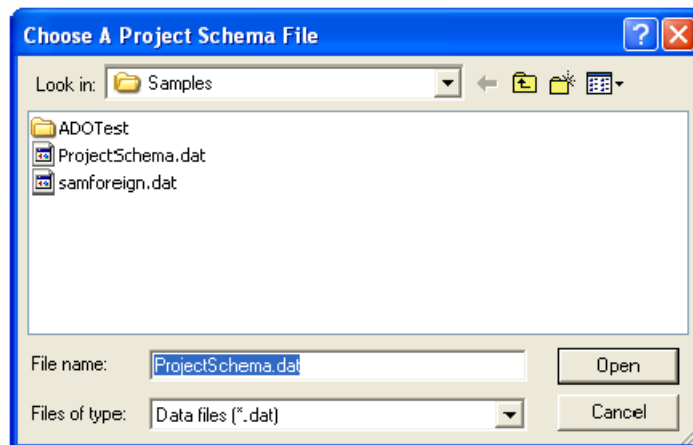

4. Create tables, columns, and rows as described below.

5. When complete, save the *table schema* to file using the **File>Save As** menu option or click 🖫 on the toolbar.

4.2.4   Editing an Existing Table Schema

Table Designer will normally open the currently active table schema when it starts. You can open a different table schema file using the **File>Open** menu option or selecting from the **File>most recently used list** or clicking 📂 on the toolbar.

With a populated table schema, the Table Designer main form looks like this:
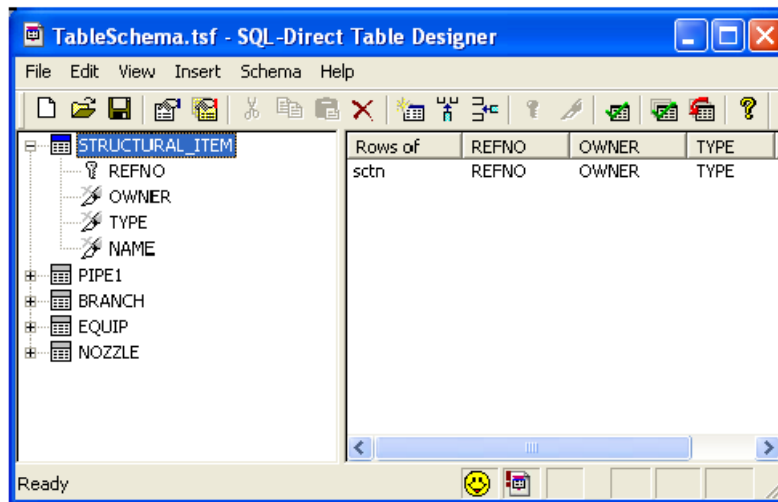
The left hand pane lists the *tables* and *column names* as a tree. The right hand pane list the *table rows* and associated *column attribute bindings* in a grid.

4.2.5   Creating a Table

To create a table:

1. Open an existing Table Schema or create a new one as described above.
2. Insert a new table and give it a *name.*
3. Define the table *columns* and the table *rows.*

Inserting and Naming a New Table

Choose **Insert>Table**, click on the toolbar, or right-click in the left-hand pane and select **Insert Table**.

Table Designer creates a new table and selects the default name `"NEWTABLE"`. We recommend you change this to a more meaningful name.

Choose a name that describes the purpose of this table, e.g. `"SITES"` or `"PIPING"`. You can use letters, numbers and underscores in the table name, but you cannot use spaces. The right hand pane should now show a column heading labelled "Rows of".

This column lists all the element types that provide the data for this table.
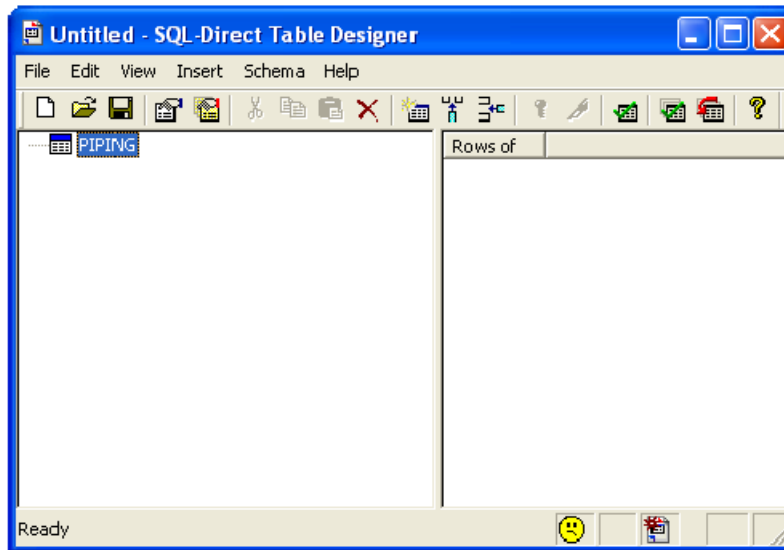
Adding Columns to the Table

Select **Insert>Column**, click ᛨ on the toolbar, or right-click on the table name and select **Insert Column**.

Similarly to when you create a new table, the column will have a default name, "NEWCOLUMN". Choose a new name (e.g. "REFNO" or "NAME") and press **Enter**. Note that column names can contain spaces.

The new name appears in the right-hand pane. Repeat this process for all the columns that you need. Note that Table Designer inserts the new column to the right of the current selection, so if you select a column, it inserts the new column to the right of the selected column; but if you select the table name, it inserts the new column as the first column in the list.

Note: If you select a column in the left-hand pane, only that column will appear alongside the "Rows of" column in the right-hand pane.
If you select the table name, the right-hand pane shows all columns of that table.

When you've inserted the columns, your Table Designer should look something like this:
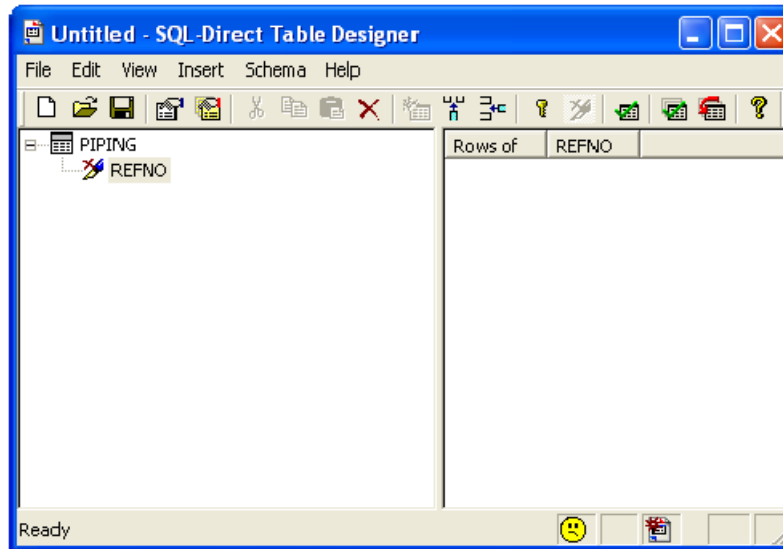
Important: If you want to write data back to the database, one of your columns must map to REFNO or NAME. We recommend you use REFNO.

- **Adding Rows to the Table**

    You can now enter the rows that tell SQL Direct Configuration where to get the data from. Select the table name and then select **Insert>Row** or click ⋮⋮ on the toolbar

    A drop-down list will appear under the "Rows of" column in the right-hand pane. Click on the down-arrow and select an element type from the list. OUTFITTING Provider locates all elements of this type when it retrieves data.

    If you labelled any columns with the exact name of an element attribute, e.g. `"REFNO"`, Table Designer fills in the attribute for you; Table Designer assumes if the name of a column matches an attribute name, then that is the attribute you want.

    If there are any entries along your row under the headings marked `"unset"`, then you need to tell Table Designer which attribute provides the data for that column, i.e. you need to set the *binding* for that column. Doubleclick on the word `"unset"` and select the attribute you want from the drop-down list. The list contains all the attributes for the type listed in that row.

    Having selected an attribute for an "unset" (or changed one that Table Designer put in by default), Table Designer will attempt to re-use that attribute in any subsequent rows you may add. In other words, it looks at the bindings for the row immediately above the new row when choosing defaults, and uses the column names for the first row. It is quicker therefore, to get the first row exactly right, then add new rows, rather than add all the rows before checking the bindings.

    Repeat from *Inserting and Naming a New Table* to create all the tables that you need.

    To save the table you just defined as a **.tsf** file, select **File>Save**, or **File>Save As**…, or click 💾 on the toolbar.

4.2.6 Editing Existing Tables

Change a Table

To change a table (or column) name, select the table (or column) in the left-hand pane, select **Edit>Rename**, or right-click on the name and choose **Rename** or select the name, pause, then click on the name again (note: not a double-click).

Delete a Table or Column

To delete a table or column, select the table or column in the left-hand pane, select **Edit>Delete**, click ✕ on the toolbar, or right-click on the name and select **Delete**.

Delete a Row

To delete a row, select the row in the right-hand pane, select **Edit>Delete**, or click ✕ on the toolbar.

4.2.7 Loading and Saving Tables Data

Save a Table

To save the table you just defined, select **File>Save**, **File>Save As**…, or click 💾 on the toolbar and save the schema as a **.tsf** file.

Load a Table

To load a table schema, select **File>Open**…, or click 📂 on the toolbar. Table Designer displays the currently active table schema on startup.

Note that **Schema>Reload Table Schema** (📇 on the toolbar) discards all changes you have made to the current table schema.

4.2.8 Writeable (and Deleteable) Data

If you have followed the above procedure, you now have a set of read-only tables. You should also notice that every **Edit** menu command (**Edit>Undo**, **Cut**, **Copy**, **Paste** and **Change**, and the equivalent toolbar buttons) and the **Toggle read-only flag** toolbar button (🔧) is disabled.

This is because you can only mark rows as writeable if you also define a *key column*.

SQL Direct Configuration uses the key column to identify an element when it is writing data back, so you need to specify either the REFNO or the NAME as the key column. Table Designer does not enforce this - in theory you can mark any column as the key, but in practice only REFNO or NAME are useable.

Select the required column in the left-hand pane, then select **Schema>Key Column**, the 🔑 button, or right-click on a column name and select **Key Column**.

Once you have marked a key column, you can turn off the read-only flag for any other column by selecting **Schema>Read Only**, clicking 🔧 on the toolbar, by or right-clicking on a column name and selecting **Read Only**)**.**

Remember that the onus is on the person designing the table to ensure that writing to that attribute is sensible. Table Designer does not enforce any rules about attributes that must be read-only.

4.2.9 Checking a Table

At any time you can select **Schema>Check Selected Table** or click on the toolbar to check your table file for consistency. This process involves checking:

*   you have one or more rows and columns in the table;
*   there is a valid binding for each column/row intersection.

4.2.10 Checking the Table Schema

At any time you can select **Schema>Check Table Schema** (or click on the toolbar) to check your table schema file for consistency. This process involves checking:

*   you have one or more rows and columns in all tables;
*   there is a valid binding for each column/row intersection.

Note: *Valid binding* here means the attribute specified is an attribute of the element type, not that it necessarily makes sense in the overall context of the table.

If the check does detect any errors, you can correct them as they are discovered.

Note: If you are working on the active table schema, you cannot save it to the same file if it contains any errors. You have to save to a different file.

Schema>Check Table (on the toolbar) checks the individual table currently selected in the tree view.

4.2.11 Activating a Table Schema

Once you have created your table schema, checked it, and saved it, you can make it *active* by returning to the SQL Direct Configuration program and adding the table schema file to the connection details.

4.2.12 'Smiley' in the Status Bar

The "smiley" is an indication of how valid your table is.

If he's happy , then your table schema has been checked and is OK (within reason - see below).

If he's sad , then the table schema has been checked but there are errors.

If he's undecided , it means that you have made some changes but haven't yet checked the table schema.

The panel to the right of the smiley indicates whether or not you are editing the "active" table schema file (i.e. the one that SQL Direct Configuration is configured to use for the current connection). The icon in here reminds you that making changes may be dangerous.

The third icon panel indicates whether or not you have made changes to the file and have not yet saved them.

4.2.13 Querying Tables and Table Columns

The **Table Designer Properties** form shows properties of the currently selected table or column. Invoke this form by selecting **File>Properties**, the button or right-clicking on a table or column name in the right-hand pane and selecting **Properties**.

You can leave the **Table Designer Properties** form active while you navigate around the table schema. You can use this form to change the name of a table or column and to set the data properties of columns (how the data is returned: STRING, INTEGER or REAL [REFNO is equivalent to STRING]). The default data type is STRING which is usually adequate for most reporting purposes (although you may wish to change the default size of 22 to something bigger).

The **Table Schema Properties** form lets you change the project schema file associated with the table schema. Bring up this form by selecting **File>Schema Properties** or the button. The form also allows you to specify how the table schema file should point to the project schema file - using *relative* or *absolute* pathnames.

Absolute vs Relative Pathnames

From the **Table Schema Properties** form you can specify whether the table schema file should refer to the project schema file with an absolute or a relative pathname. Which you choose depends how you wish to use and distribute these files.

If you will have one centralised project schema file and multiple table schema files distributed in different places, then choose **Absolute**. However, if you will distribute the project schema file with the table schema file, or you use centralised files of both types, choose **Relative**.

Remember that **Absolute** means that the table schema file points to a particular project schema, no matter where the table schema file is located, whereas **Relative** means that the *project* schema file is assumed to be *at or below* the hierarchical level of the *table* schema file. The **Change**… button enables you to assign a different project schema file to be associated with the current table schema file. This operation also changes the pathname addressing to **Relative**.

Note that until you have saved the table schema file at least once (i.e. Table Designer has a directory to work from) you can't select **Relative**.

5 Using SQL Direct - Some Examples

This chapter gives examples of using SQL Direct to access OUTFITTING data with some of the principal data consumer applications available, but without having to have SQL Server installed. It is assumed that:

- OUTFITTING is correctly installed
- SQL Direct is correctly installed
- You have used the Table Designer to set up some tables. See *Table Designer* and *Using the Table Designer*.

Full details of the installations are in the relevant installation guides.

Example database files for each of the applications named in this chapter are supplied as part of your installation. You can find them in:

C:\AVEVA\SQL-Direct12.0\Samples

5.1 Reading OUTFITTING Data into Microsoft Excel

A ScreenCam video demonstration file (**excel.exe**) showing the use of Excel to read in OUTFITTING data comes with your installation.

1. Start Microsoft Excel
2. Select **Data>Import External Data>Import Data…**. The **Select Data Source** dialog box appears:
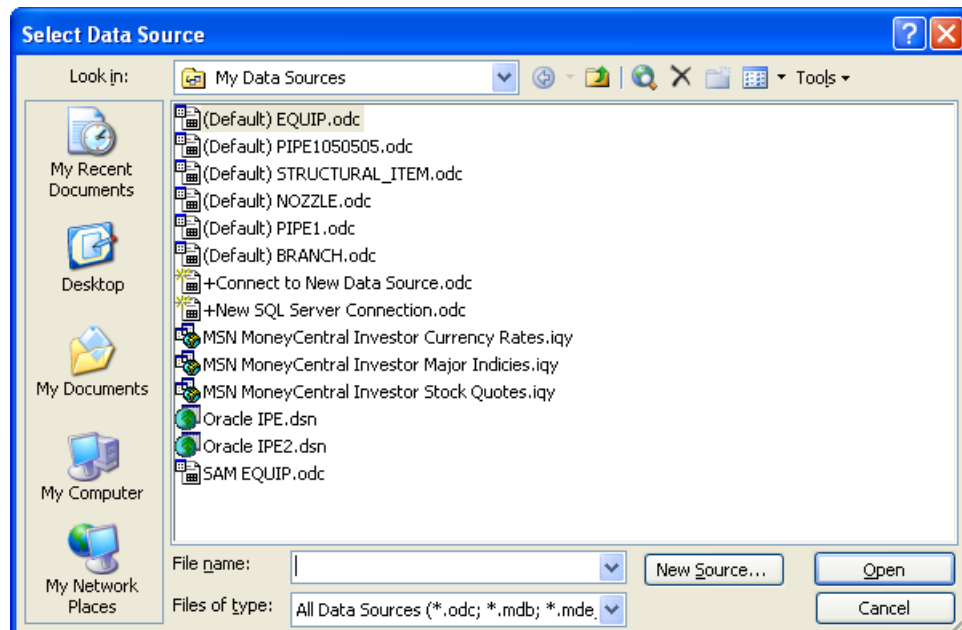
**Note:** The list of data sources may differ from that shown above.

3. Any previous data source connections appear in this list and can simply be selected and opened using the **Open** button. This takes you directly to the **Import Data** screen shown at Step 10 below.

4. To create a connection to a new data source, click the **New Source…** button and select **Other/Advanced** on the **Data Connection Wizard** list displayed:
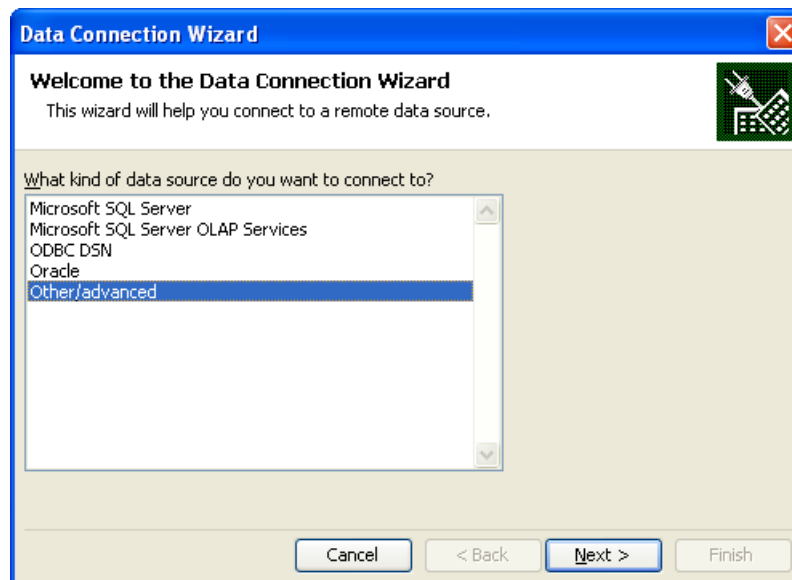

5. Click the **Next>** button and select OUTFITTING**Prov OLE DB Provider 12.0** on the **Data Link Properties** list displayed:
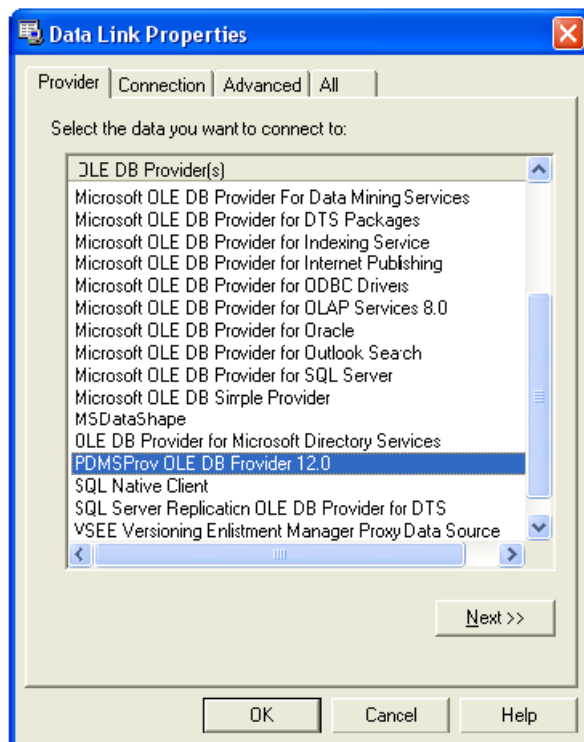
6. Clicking the **Next>>** button will take you to the **Connection** tab of this **Data Link Properties** form:
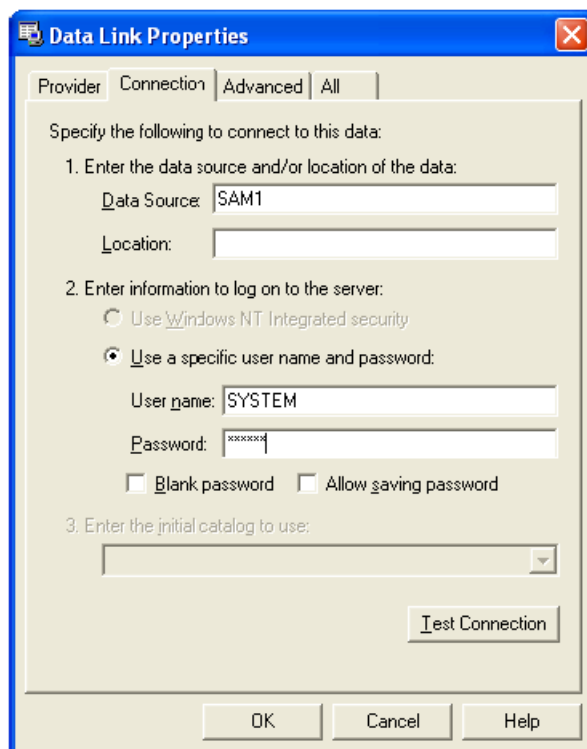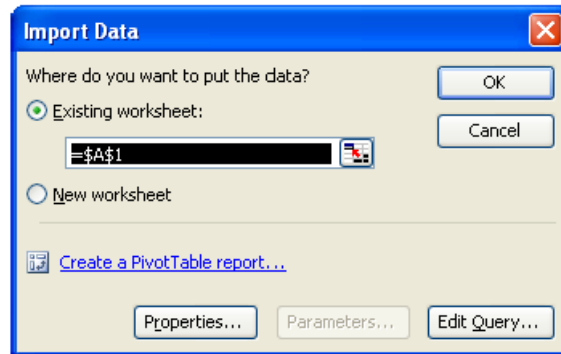
7. Fill in the **Data Source** field with the connect string for the connection, and fill in the **User Name** and **Password**. You may have to uncheck the **Blank Password** check box before you can supply the password.

8. Clicking the **OK** button brings you to the **Select Database and Table** page of the **Data Connection Wizard**:
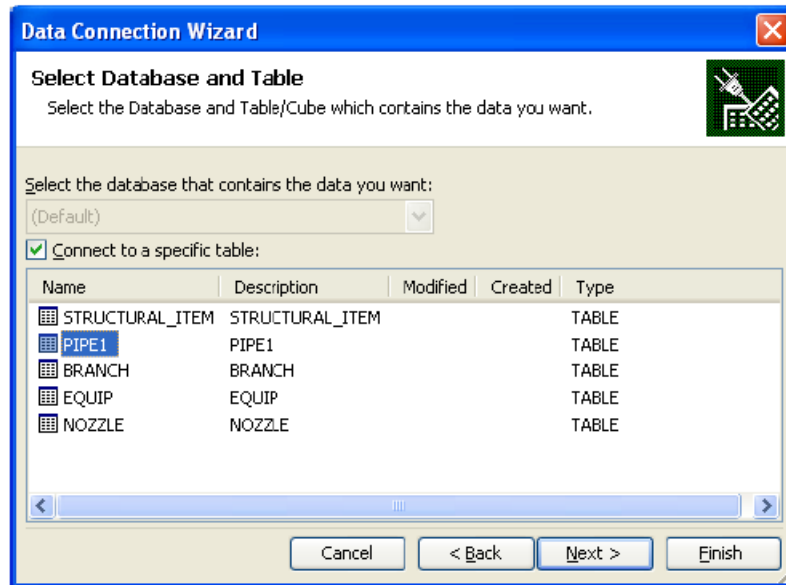
9. Select the table you wish to display and click the **Next>** button. This gives you an opportunity to name and save the link on the **Save Data Connection File and Finish** page:
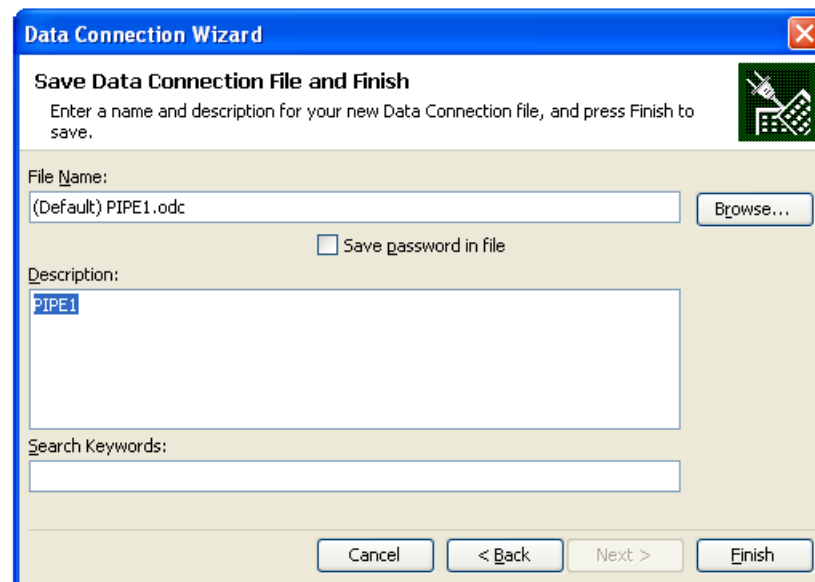
10. Click the **Finish** button and **Open** the new link on the next screen. This leads to the final **Import Data** screen:

11. Finally press the **OK** button and the table will be displayed on the Excel page:

Note: Data read into an Excel spreadsheet in this way can be edited if desired. However, there is no link back to OUTFITTING, and changes made this way will not be reflected in the database.

Data can be modified when necessary using OUTFITTING itself (e.g. a modify, create or delete operation) and should be followed by a **Savework** operation. When this has been done, executing a **Refresh Data** command in Excel will result in a corresponding change taking place in the spreadsheet (the **Refresh Data** command is on the **Data** menu, or you can click the ! icon.)

If you configure SQL Direct for use with SQL Server (see *Using SQL Direct with Microsoft SQL Server*) you can modify the data with an SQL Query. To do this, position the cursor outside the imported data, and select **Data>Import External Data>New Database** Query. Select **New Data Source…** and click **OK**. Enter any name of your choosing for the data source, and select **SQL Server** as the driver. Click the **Connect…** button and enter (**local**) as the server; click **OK**, then **OK** again.

With your new data source highlighted, click **OK**; this brings up the Microsoft Query window. Close the **Add Tables** dialog that appears, and click the **SQL** button in the menu bar or select **View>SQL…** Enter an SQL Query to update your data and click **OK**. Click **OK** to the 'SQL Query can't be represented graphically…' warning, and again to the 'SQL statement has been executed successfully' message. Close the Microsoft Query window, clicking **Yes** to the 'You have not created a query containing data…' warning. Finally, execute a **Refresh Data** command in Excel to see your updated data.

You can save your Excel spreadsheet. After reloading, you may find that the Refresh Data command does not work. This is because your password will not be saved in the file by default. Edit the query (right click the data grid and select **Edit** Query**…**) and add the **Password=<password>**; option to the connection details.

You can email your Excel spreadsheet, with its current content, to another SQL Direct user. The recipient can refresh the data from his own SQL Direct configuration, providing he edits the query (right click the data grid and select **Edit** Query**…**) to enter his own connection details. He will also need the right table definitions in his table schema file.

5.2 Reading OUTFITTING Data Using ADO from a Microsoft Visual Basic Program

See example at **C:\AVEVA\SQL Direct 12.0\Samples**

The sample program **ADOTEST** is a simple example containing an ADO data control and an ADO data grid. A combo-box illustrating use of ADO connection and recordset objects lists the available OUTFITTING tables. Selecting a table displays it in the grid.

This application requires Visual Basic 6. The executable depends on **MSVBVM60.DLL** which you will have if you have installed Visual Basic 6.
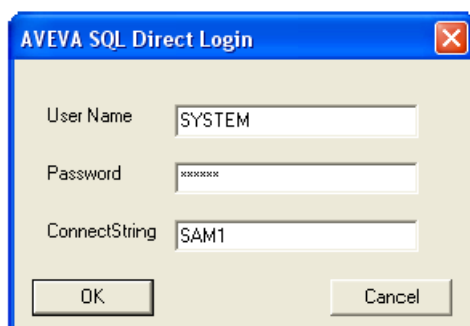
To use **ADOTEST**:

1. Start the program from within SQL Direct Configuration by highlighting a connection and selecting **Connection>Test Connection…** (see *Specifying OUTFITTING Connection Details*). On start up the main screen will look like this:
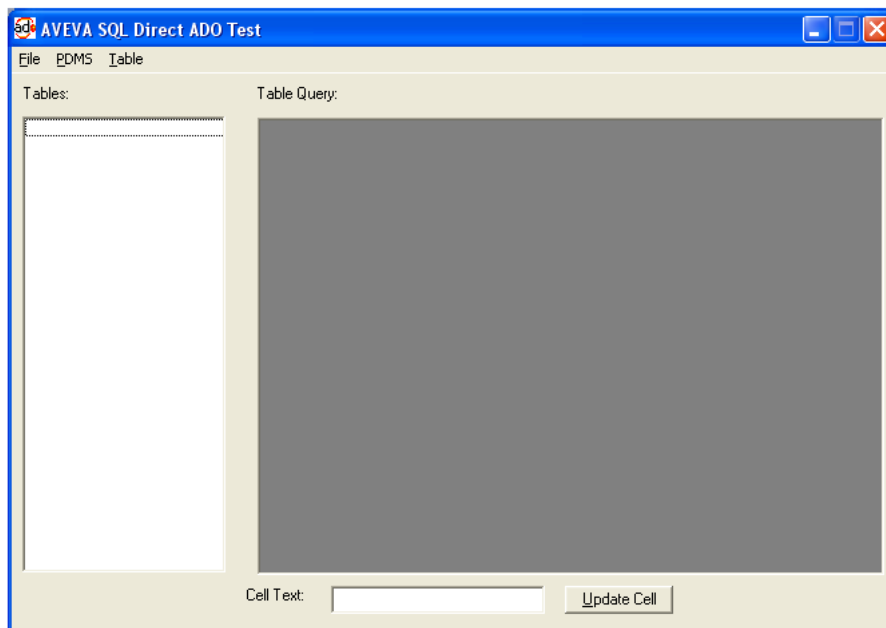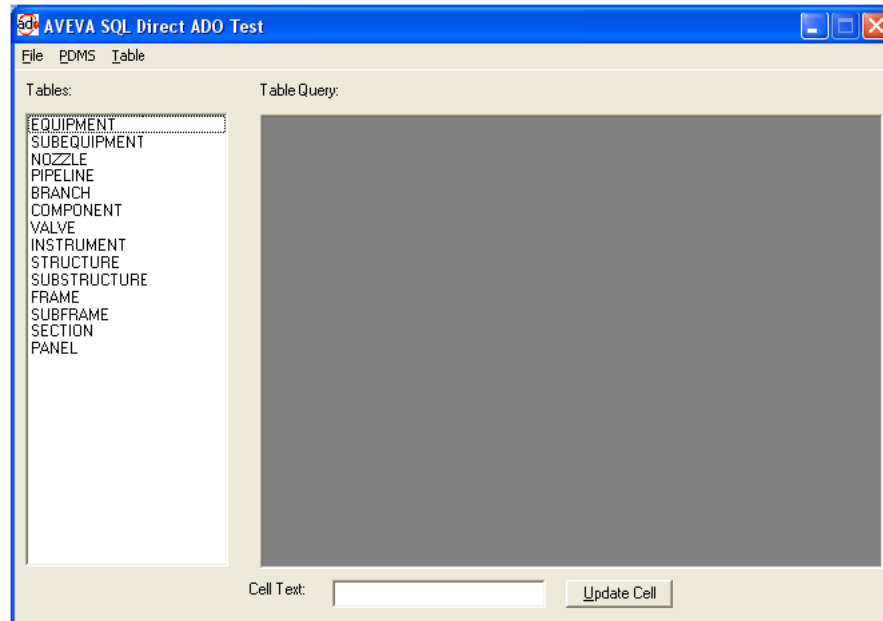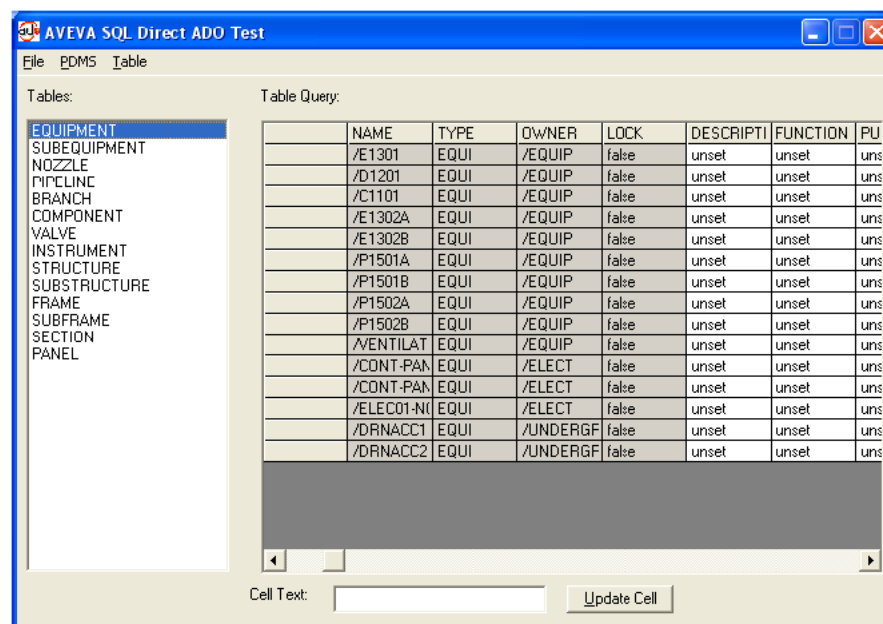
2. Select **OUTFITTING>Connect** to display the login screen:

3. Fill this in for the connection you want to test and the main form will be populated as follows:

4. Select a table from the list and then the **Table>Query Rows** menu option (or you can simply double-click the table). This displays the data in the right hand pane:


## 5.3   Writing Data Fields Back to OUTFITTING

The following extends the ADO example of the previous section.

After executing a **Table>Query Rows** command (see the previous section), the right-hand pane displays the writeable fields on a white background, while the readonly fields are displayed on a grey background.

Select a cell to edit by clicking inside it with the mouse cursor. The field editor box labelled **Cell Text** displays the field's value:
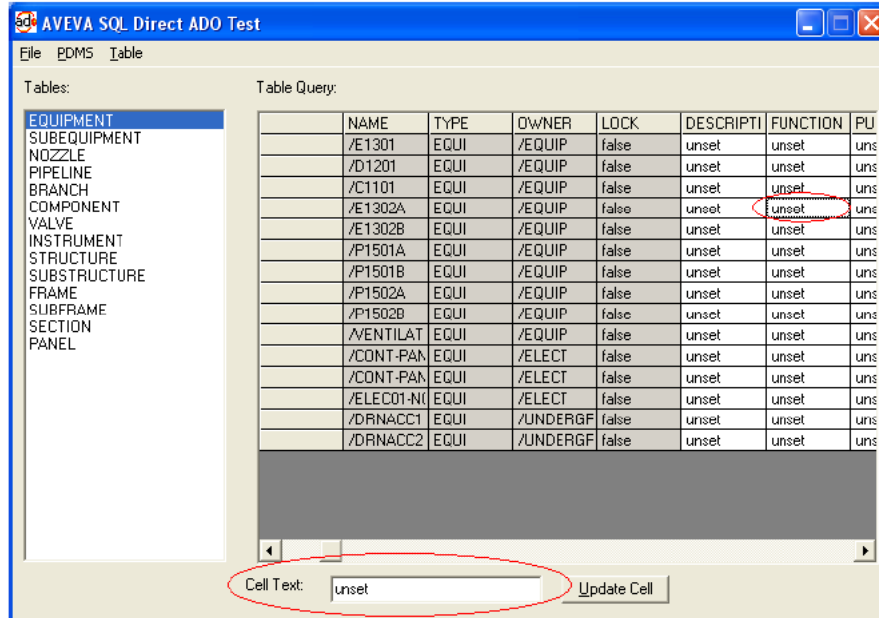
Type the new value into the editor and press the **Update Cell** button to copy the value back to the cell in the grid:
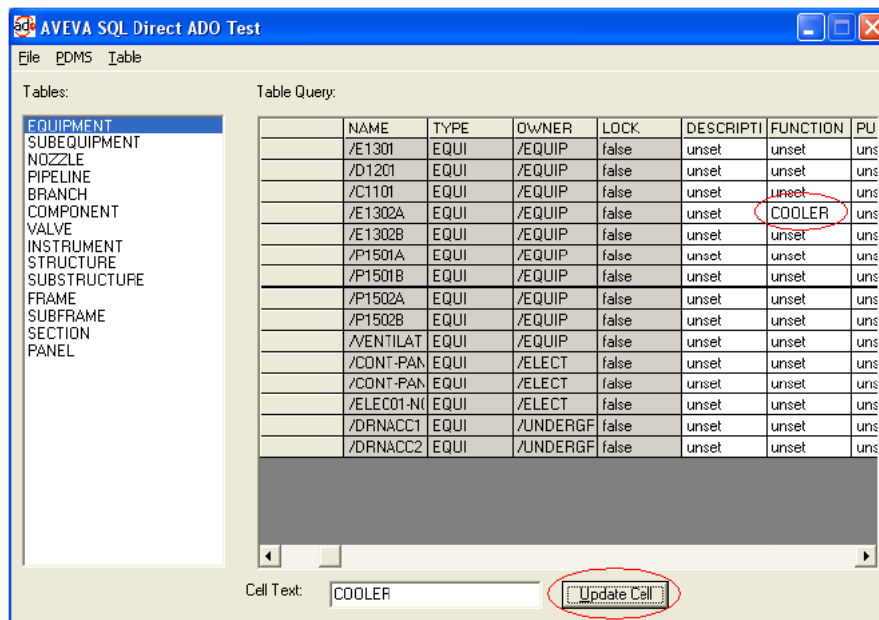
Continue to edit fields in this way as desired. The values have not been written back to OUTFITTING yet. This happens when you close the connection, for example by selecting the **OUTFITTING> Close Connection** menu option. This gives you the opportunity to accept or reject the changes:
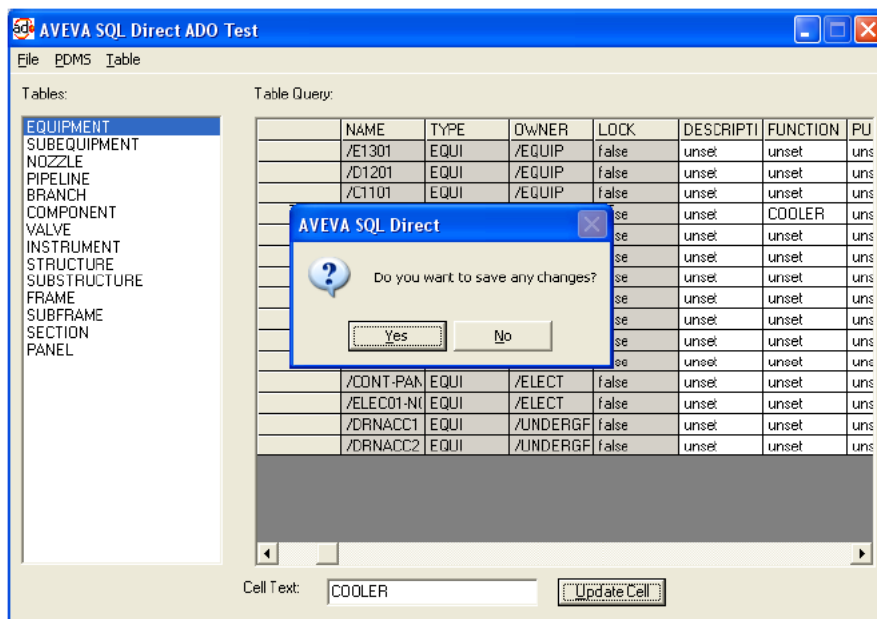
If you click **Yes** the changes will be committed in OUTFITTING. Clicking **No** aborts the changes. In either case the connection is subsequently closed.

To see the code behind this example you can look at the ADOTest Visual Basic source provided in the samples directory.

6 Using SQL Direct with Microsoft SQL Server

6.1 Introduction

This chapter details using SQL Direct with Microsoft SQL Server.

Note that you can use OUTFITTING Provider from many clients, for example Excel, or clients using Microsoft Active Data Object (ADO) technology, without the involvement of SQL Server. *Using SQL Direct - Some Examples* details some of these methods of use.

However if you want to make more sophisticated SQL queries against the OUTFITTING tables, or if you want to join OUTFITTING rowsets with data obtained from other providers, SQL Direct requires SQL Server to do the SQL processing involved.

6.2 Setting up SQL Server

For details on obtaining SQL Server, see the Microsoft SQL Server web site (address at the front of this document). Normally it is downloaded - SQL Server Express (and MSDE before it) are available free of charge.

Install SQL Server by following the instructions provided by Microsoft. The software comes with tools to start and stop the SQL Server service. You can also start and stop the service through the SQL Direct Configuration **Tools** menu.

You can if you wish also configure SQL Server as an OBDC datasource. To do this, first start SQL Server, then select **Control Panel>Administrative Tools>Data Sources (ODBC)**, select **File DSN** (you can also use a System DSN), click **Add** and select **SQL Server** from the list of drivers.

Click **Finish**, then name and create a new datasource which connects to the local server. Use the default configurations, i.e. keep clicking **Next** until the sequence is complete.

6.3 Using SQL Server with SQL Direct

If SQL Server is installed, the SQL Direct Configuration program **Tools** menu will have the SQL Server menu items enabled. The **Configure SQL Server** option will be enabled once you have created and highlighted a set of connection details.

6.3.1 Configuring SQL Server

Following saving a set of tables from Table Designer, (See *Loading and Saving Tables Data*) you must use the **Configure SQL Server** menu option before you can view the tables using any application that accesses the data through SQL Server.

You must configure SQL Server *after* installing both SQL Server and SQL Direct and *before* using any application that accesses SQL Server data.

6.3.2   Accessing OUTFITTING Data through SQL

Once you have configured SQL Server using the SQL Direct Configuration **Configure SQL Server** option, you can access OUTFITTING data using SQL queries. To do this, you need a software application that provides a SQL command window. Examples of such tools are:

- Microsoft SQL Server Management Studio 2005
- Microsoft Visual Studio .NET 2003 (Server Explorer window)
- Microsoft SQL Server Enterprise Manager 2000

(Unfortunately, there is no suitable tool actually provided with MSDE 2000, but all the above products can be used successfully with data in MSDE.)

These products initially provide a "tree view" of the data within SQL Server. You can expand each branch, down through different databases to individual tables and data views. The SQL Direct Configuration **Configure SQL Server** option will have created data views within the "master" database on your SQL server, the views being named according to the tables you created in Table Designer, and saved in your table schema file referenced in your connection settings.

You can retrieve data from data views by right-clicking on them and selecting **Open View** or **Retrieve Data From View** (depending on which product you are using).

You can also open a SQL command window and execute SQL statements. (In Visual Studio .NET 2003 you need to have retrieved data from the view first in order to open the SQL window, as the 'SQL' option on the query menu bar is greyed out until you have done this.)

Once you have opened a SQL command window you can access OUTFITTING data using SQL queries. For example:

```
SELECT *
FROM EQUIPMENT
WHERE REFNO = '=12345/67'
```

Where EQUIPMENT is a table you created in Table Designer, REFNO is a column name and =12345/67 is a value of REFNO. Note the use of single quotes around character data values.

Full documentation of the SQL Query language and its use is beyond the scope of this user guide. There are many texts available, both online and in book form.

6.3.3   Writing OUTFITTING Data through SQL

SQL provides statements to update and delete data in a database as well as simply query it. For example:

```
UPDATE EQUIPMENT
SET AREA = 100
WHERE REFNO = '=12345/67'
```

and:

```
DELETE
FROM EQUIPMENT
WHERE REFNO = '=12345/67'
```

are fairly self explanatory.

You can also update data by editing the values in the output grid produced by retrieving data from the view or executing a SELECT Query. Position the text cursor by clicking on the item

you wish to change, edit the value, then move the cursor off the row. SQL will update the data in the database.

Note: If you are using SQL Server 2000 / MSDE, you need to make sure that the Microsoft Distributed Transaction Coordinator (DTC) is running before you can update data by editing the output grid. The DTC can by started up / stopped through the SQL Server Service Manager

The SQL INSERT statement is not supported by OUTFITTING Provider at this time, neither is inserting data by adding rows to the output grid.

6.4 Using Microsoft Access with SQL Direct

Microsoft Access (version 2002 onwards) has the ability to create "Linked Tables". These are tables of data that are held outside of Access in an external database or file. Linked tables can be based, for example, on data in SQL Server. With SQL Direct, you can base linked tables on data that is actually in OUTFITTING rather than a native SQL database. Access will not know the difference.

Two methods of defining linked tables are presented. The first method requires that an ODBC resource has first been created. The second method defines linked tables directly using OLE DB.

For further details about linked tables, see your Microsoft Access documentation.

6.4.1 Method 1 - Linking to OUTFITTING Data through ODBC (recommended)

This example covers the steps necessary to create a linked table of OUTFITTING data through a previously created ODBC resource. Linked tables thus created can be used to write data back to OUTFITTING as well as fetch data.

The ODBC resource must point to the SQL Server database in which our table definitions have been configured (in our case the MASTER database, in which the OUTFITTING data views have been created by SQLDirectConfig). See *Setting up SQL Server* for information on how to create the ODBC resource.

Note: It is also possible to create a copied (unlinked) table. The procedure is similar to that given below, but start with **File>Get External Data>Import…**

1. Start Microsoft Access.
2. Open an existing database, or create a new database, as appropriate. Select **File>Get External Data>Link Tables…** The **Link** dialog box will appear.
3. From the "Files of type:" pull-down list, select **ODBC Databases( )**. The **Select Data Source** dialog box will appear:
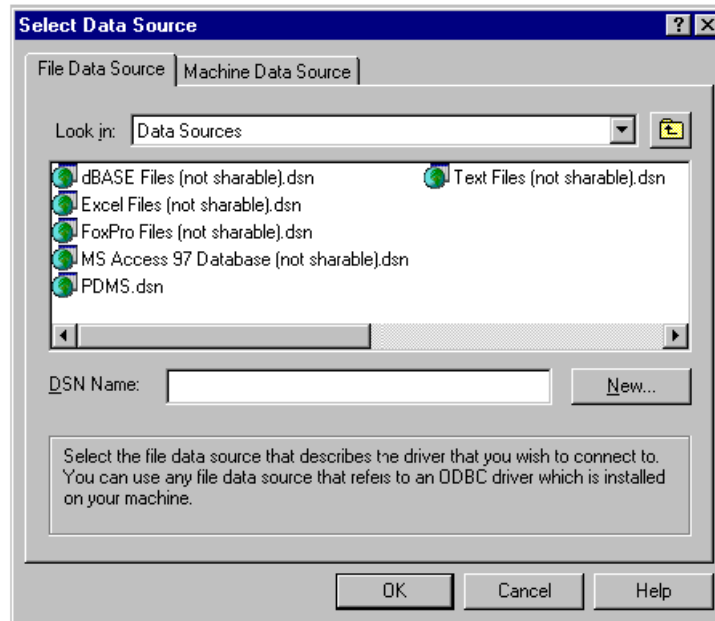
The list of data provider applications may differ from that shown above.)

4. Select the OUTFITTING ODBC data source you created earlier (see *Setting up SQL Server*). If you created a System DSN (rather than a File DSN) it will be on the "Machine Data Source" tab (rather than the "File Data Source" tab). Click **OK**; the **Link Tables** dialog box will appear:
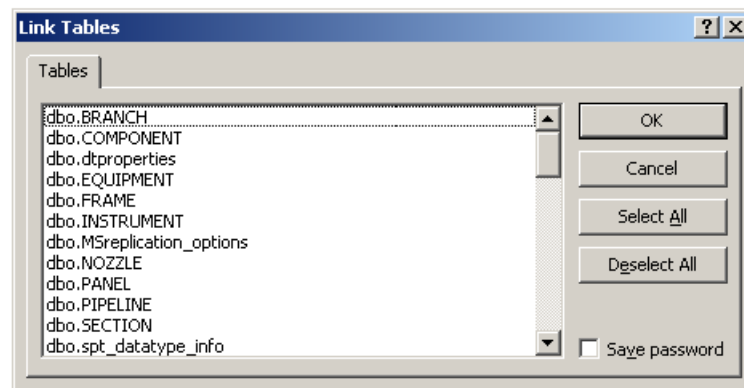
5. The list in the Tables tab will show the available OUTFITTING tables (as defined using the SQL Direct Table Designer) as well as a number of SQL Server system tables which can be ignored. Select the required table(s), then click **OK**.

6. The **Link Tables** dialog box will change its appearance and the **Select Unique Record Identifier** dialog box will appear:
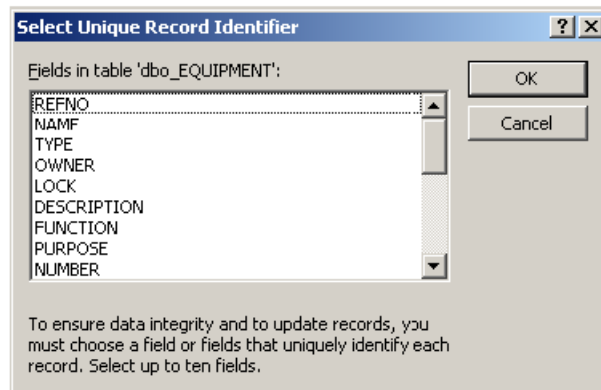
7. The **Select Unique Record Identifier** dialog box asks you to select a field (or fields) from the selected table that will uniquely identify each record. Choose **REFNO**, and click on **OK**.

The requested table(s) will now appear within the Tables tab of the **Database** dialog box in Access. Double-click on the appropriate table name to view the data in tabular form.

Note: Data in the output grid corresponding to updatable attributes in OUTFITTING can be updated by changing the value(s) and moving the text cursor off the row. In order for this to work, however, you must have started the Microsoft Distributed Transaction Coordinator (DTC) through the SQL Server Service Manager tool.

6.4.2 Method 2 - Linking to OUTFITTING Data through OLE DB

This example covers the steps necessary to create linked tables of OUTFITTING data directly through the OUTFITTING OLE DB Provider / SQL Server without using ODBC.

Unfortunately, linked tables created in this way cannot be used to write data back to OUTFITTING, due to apparent limitations in Access. If write access is important to you, please use Method 1 (above).

Note: It is also possible to create a copied (unlinked) table. The procedure is similar to that given below, but start with **File>Get External Data>Import…**

1. Start Microsoft Access (version 2002 or later), and also make sure that SQL Server is running.
2. Go to Access **File** menu, select **New**, and create a **Project using new data…**
3. In the database wizard presented, select the appropriate SQL Server (i.e. the one on the machine where SQL Direct is installed) and enter the SQL Server login credentials. Click **Next** and then **Finish**.
4. Go to the **File** menu again, and select **Get External Data->Link Tables…**
5. Make sure **Linked Server** is selected and click **Next**.
6. Select the data source. This is similar to the procedure described in *Reading OUTFITTING Data into Microsoft Excel* for Excel. If you have a saved data source that you want to use, select it from the list, click **Open** and proceed to step 12. If you don't have a data source saved, click the **New Source…** button and proceed to step 7.
7. Make sure **Other/Advanced** is selected, and click **Next**.
8. Select **OUTFITTINGProv OLE DB Provider 12.0** from the list and click **Next**.

9. Fill in the **Data source** - the connection settings name you saved in SQL Direct Configuration, your **User name** and **Password** (deselect the "blank password" option), and select "Allow saving password" if you wish. Then click **OK**.

10. You are now offered the option to select your database - there should only be one so just click **Next**.

11. Fill in a file name for the connection (if you wish, or just use the default) and again select the "save password in file" option if you wish. Then click **Finish**.

12. You can now select which table(s) you wish to link to the current project (or all tables if you wish). Make your selection and click **Finish**.

Access will now create the linked tables. Double-click on the appropriate table name to view the data in tabular form.

The linked tables created in this way are once again data views, very similar to the ones that SQL Direct Configuration creates, except that Access creates a separate database for each Access project in which to place the views.

6.5 Reading OUTFITTING Data into Microsoft Word

The following procedures can be used to read OUTFITTING data into a Microsoft Word table. Both the methods given here use ODBC and SQL Server.

Note: Using Microsoft Word, it is not possible to provide a dynamic link back to the data. If the OUTFITTING data changes in the database, the Word table must be recreated to see the changed data. Neither is it possible to write any data changes made in Word back to the database.

## 6.5.1 Method 1 - through a Named Data Source

1. Start Microsoft Word and open a .doc file.

2. If it is not already displayed, display the Database toolbar by selecting **View>Toolbars>Database**. Click the **Insert Database** icon ( ) on the Database toolbar. The **Database** dialog box will appear.

3. Click on **Get Data…** The **Select Data Source** form will appear.

4. If you have not already created a named data source, click on **New Source…** The **Data Connection Wizard** will appear. Choose **Microsoft SQL Server**, and click **Next>**.

5. Enter a single dot (".") for the server name and click **Next>**.

6. Highlight the table you wish to use, and click **Next>** again.

7. Select a file name for your data connection, and click **Finish**.

8. The **Select Data Source** form reappears - make sure your named data connection is highlighted and click **Open**.

9. The **Database** dialog box reappears. Optionally click on the **Query Options…** button, and enter any data restrictions you wish to impose on the data returned.

10. Finally click on the **Insert Data…** button and select the data records you want from the **Insert Data** dialog box.

The requested data will now appear in your current Word document as a table.

6.5.2 Method 2 - through MS Query

1. Start Microsoft Word and open a .doc file.

---

2. If it is not already displayed, display the Database toolbar by selecting **View>Toolbars>Database**. Click the **Insert Database** icon ( ) on the **Database** toolbar. The **Database** dialog box will appear.

3. Click on **Get Data…** The **Select Data Source** form will appear.

4. Select **MS Query…** from the **Tools** menu. Microsoft Query will start up, and the **Choose Data Source** dialog box will appear.
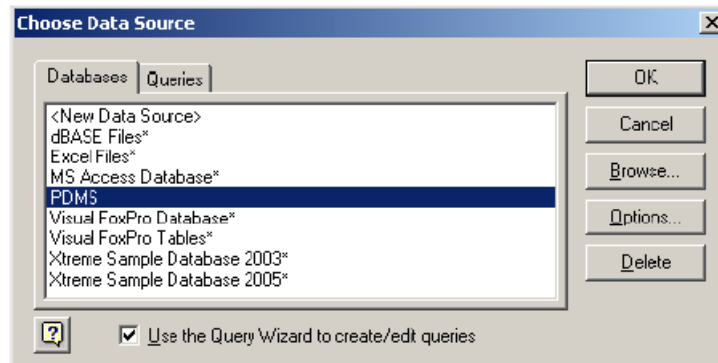
(The list of data provider applications may differ from that shown above.)

5. Select the **Use the Query Wizard to create/edit queries** check box.

6. Click on the ODBC resource in the list corresponding to the database where the OUTFITTING views were created (see *Method 1 - Linking to OUTFITTING Data through ODBC (recommended)* for a description of how to do this) and **OK** the dialog box. The **Query Wizard - Choose Columns** dialog box appears:
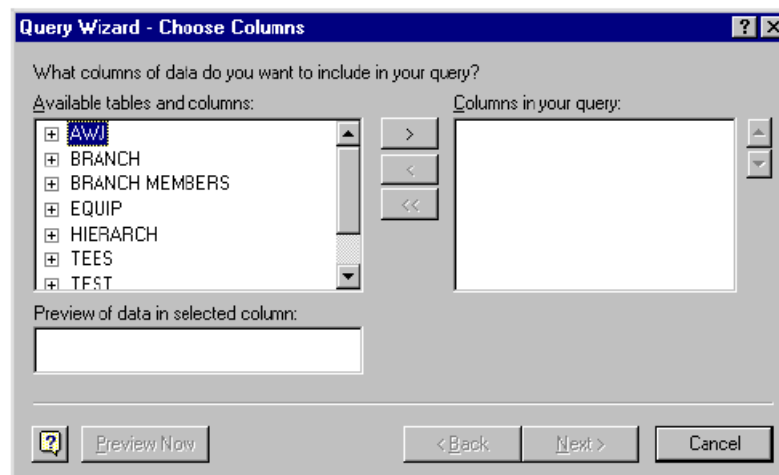
7. The **Available tables and columns:** list will show the available OUTFITTING data tables (defined using the SQL Direct Table Designer). Select the required table, then click . The column headings for the selected table will appear in the **Columns in your query:** box.

8. Click **Next>**. Keep clicking **Next>** as the wizard dialog progresses (unless you wish to perform other operations at each stage) until the **Query Wizard - Finish** dialog box appears. Click **Finish**.

9. Returning to the **Database** dialog box, click **Insert Data…** The **Insert Data** dialog box then appears, which gives you some options for the formatting and content of the imported data. **OK** the form.

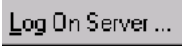The requested data will now appear in the current Word document as a table.

6.6   Reading OUTFITTING Data into Crystal Reports

This example covers the steps necessary to import data from Microsoft Access and from OUTFITTING to form a merged table within Crystal Reports. This section does **not** give general information on how to use Crystal Reports - see the online help for that product.

1. Start Crystal Reports.

2. Select **File>New…** The **Create New Report** dialog box will appear.

3. Select [icon] to activate a "wizard" to create a standard report. The **Create Report Expert** dialog box will appear:
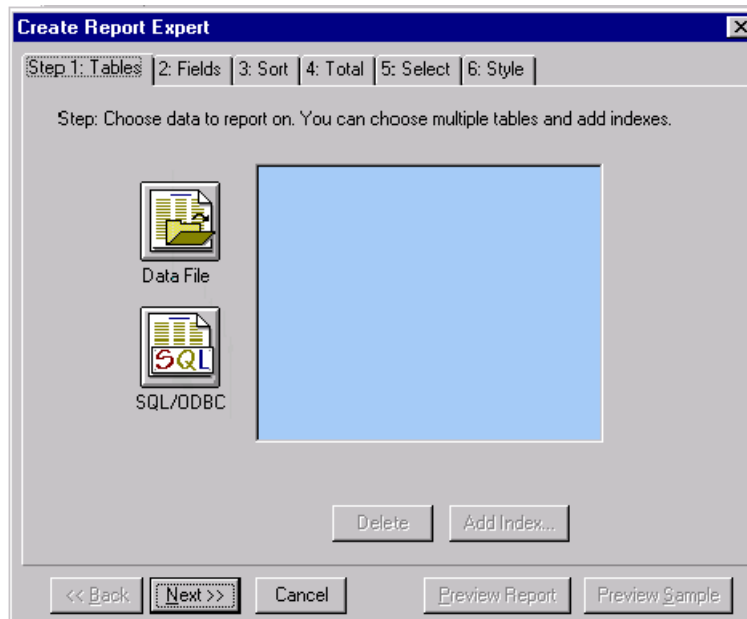
4. Step 1 (the **Tables** tab) enables you to specify the type of data source. Click on [icon] and select **ODBC -** OUTFITTING from the list in the resulting **Log On Server** dialog box (your System Administrator may have installed and configured your SQL Direct service provider to show text other than 'OUTFITTING') and **OK** the dialog box.

5. The **Choose SQL Table** dialog box will appear. From the SQL Tables list choose the table (previously defined using SQL Direct's Table Designer) which contains the data you wish to report on. Having selected the table, click **Add**; the selected table name will appear in the main list area on the **Create Report Expert** dialog box (**Tables** tab).

6. On the **Choose SQL Table** dialog box click [Log On Server ...] to redisplay the **Log On Server** dialog box. Now select **ODBC - MS Access 7.0 Database** from the list to specify (in this example) the source of the data which is to be merged with the already-

specified data table, and which is to be reported on. **OK** the dialog box. The **Select Database** dialog box appears.

7. Navigate to the appropriate Access .mdb file and **OK** the **Select Database** dialog box. The **Choose SQL Table** dialog box will reappear. From the **SQL Databases** list select the **MS Access 7.0 Database** previously selected. The **SQL Tables** list will become populated with the names of the tables present in the selected Access database. Having selected the table, click **Add**, then **Done**.

This completes step 1 of the **Create Report Expert** process.

8. Step 2 of the **Create Report Expert** process consists of linking the OUTFITTING data table with the Access data table. The **Links** tab shows lists of column headings in the two tables. 'Join' the tables by selecting (left mouse button) a common column heading in one table; with the mouse button held down, move the cursor over to the same column heading in the other table and release the button. A line will appear joining the two tables via the common column heading. The 'joint' table to be reported on has now been defined. Click on Next >> .

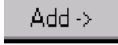This completes step 2 of the **Create Report Expert** process.

9. Step 3 of the **Create Report Expert** process consists of setting the fields (column headings) which are to appear in the report. From the **Database Fields** list on the left of the **Links** tab select the column headings you wish to see in the report, choosing fields from the 'OUTFITTING table' and from the 'Access table'. Select each required field, clicking on Add -> after each selection to assemble the required report fields in the **Report Fields:** list. Click on Next >> when you have completed this process.

This completes step 3 of the **Create Report Expert** process.

10. Step 4 of the **Create Report Expert** process consists of choosing report fields to sort and group by. From the **Report Fields** list on the left of the **Sort** tab, select the report fields you wish to sort and group by, clicking Add -> after each selection to assemble the required group fields in the **Group Fields:** list. Click Next >> when you have completed this process.

This completes step 4 of the **Create Report Expert** process.

11. Steps 5, 6 and 7 of the **Create Report Expert** process enable you to further refine the content and detail of the report to be produced. Options chosen for each of these steps will be up to individual user requirements and so are outside the scope of this description. See the Crystal Reports online help for details of the options available at each of these steps.

12. Having selected the required report style from the **Style** tab, click Preview Report to see the report.

Index

W