User Guide and Reference Manual

Easysoft® ODBC for RMS

Easysoft Limited

Published by

Easysoft Limited
Thorp Arch Grange
Thorp Arch
Wetherby
LS23 7BA
England

Version 15.

This manual documents version 1.5 of Easysoft ODBC for RMS.

Licence Agreement

The copyright and all other intellectual property rights of whatever nature in the Software (which term shall where appropriate include any new releases of the Software provided to you) ("the Software") supplied herewith is owned by Easysoft Limited (registered in England and Wales under number 2780502) whose registered office is at Thorp Arch Grange, Thorp Arch, Wetherby, LS23 7BA ("the Registered Office") ("the Owner"). You may not load the Software into any computer or copy it without the licence of the Owner. The Owner offers you a non-exclusive licence on the terms set out herein. One of the terms of this Licence is that during the period of time you are licensed to use the Software you must be a party to a valid maintenance and support contract with the Owner in respect of the Software on terms acceptable to the Owner. Your supplier will have provided you with details of the alternative maintenance and support services offered by the Owner and the terms of such services.

If you do not wish to be bound by the terms in this Licence you may within *14* days of delivery return the Software, its packaging and documentation unused to your supplier together with proof of purchase for a full refund.

1 Licence Terms

1.1 You are permitted to:

(a) load the Software and use it only on the number of computers agreed in writing between you and your supplier and only on those which are under your control;

(b) make up to 3 copies of the Software for back-up purposes only in support of the permitted use. The copies must reproduce and include the Owner's copyright notice;

1.2 You are not permitted:

(a) to use the Software on any computer or system which permits electronic access to it by more than one user other than as specifically agreed in writing between you and the supplier;

(b) to rent, lease sub-licence, loan, copy, modify, adapt, merge, translate, reverse engineer, decompile, disassemble or create derivative works based on the whole or any part of the Software or its associated documentation except as expressly provided herein;

(c) except as expressly provided herein, to use, reproduce or deal in the Software in any way;

(d) to transfer the Software or its associated documentation or this Licence to another person, business, entity or Company.

2 Acceptance

By loading the Software onto any computer you are further confirming that you have accepted the terms of this Licence.

3 Procedure

3.1 When you have loaded the Software onto a computer you should telephone the Licence Validation Centre (as notified to you by your supplier) who will give you a password which will enable you to use the Software. By requesting a password you are further confirming that you have accepted the terms of this Licence.

3.2 Your supplier will submit to you the terms of and conditions of the maintenance and support service which you have agreed to purchase from the Owner which must be executed and returned to the Owner or to a person specified by him.

4 Term

4.1 Any licence granted under this document is effective until it is lawfully terminated by the Owner pursuant to any contract between you and the Owner and in addition shall terminate forthwith if:-

4.1.1 you fail to abide by the terms of this Licence; or

4.1.2 at any time during the continuance of this Licence you are not party to a valid maintenance and support agreement in respect of the Software with the Owner or have not otherwise agreed to purchase maintenance and support services in respect of the Software from the Owner; or

4.1.3 you are in breach of any of the terms of a maintenance and support agreement between you and the Owner.

4.2 Upon termination you shall no longer use the Software and shall return to the Owner or destroy (as the Owner shall instruct) the Software and all documentation relating thereto and all copies of the same and shall if so required by the Owner furnish the Owner with a certificate certifying that the same has been done.

5 Ownership

You own only the media (or authorised replacement) on which the Software is recorded. You may retain the media on termination provided the Software has been erased and you can provide reasonable evidence to the Owner to this effect. The Owner shall at all times retain ownership of the Software and all intellectual property rights and copyright in the Software, as recorded on the original media and all subsequent copies thereof regardless of form. This Licence applies to the grant of the Licence only and not to the contract of sale of the media.

6 Warranties

6.1 The Owner warrants that the media on which the Software is supplied will be free from defects in materials and workmanship under normal use for a period of 90 days after the date you obtain the Software from the Owner or your supplier ("the Warranty Period"). If a defect in the media shall occur during the Warranty Period it may be returned with proof of purchase to the Owner who will replace it free of charge.

6.2 The Owner warrants that during the Warranty Period the Software will perform substantially in accordance with the Software Product Description contained in the Manual provided with the Software (provided that the Software is used on the computer and with the operating system for which is was designed and is used in the manner set out in the user manual with the software and hardware equipment specified in the Software Product Description) and that the Software Product Description correctly describes the operation of the Software in all material respects. If the Owner is notified of significant errors during the Warranty Period it will correct any such demonstrable errors in the Software or such documentation within a reasonable time or (at its option) provide or authorise a refund of the price of the Software (against return of the Software and its documentation).

6.3 The above represent your sole remedies for any breach of the Owner's warranties and the Owner shall have no liability or obligations under the above warranties other than to remedy breaches thereof by the provision of materials and services within a reasonable time and without charge to you. If the Owner shall fail to comply with such obligations its liability for such failure shall be limited to a sum equal to the price paid by you to the supplier for the Software.

6.4 The express terms of this Licence are in lieu of all warranties, conditions, undertakings, terms and obligations implied by statute, common law, trade usage, course of dealing or otherwise all of which are hereby excluded to the fullest extent permitted by law.

6.5 The Owner does not warrant that the Software will meet your requirements or that the operation of the Software will be uninterrupted or error-free or that defects in the Software will be corrected. You shall load and use the Software at your own risk and in no event will the Owner be liable to you for any loss or damage of any kind (except personal injury or death resulting from the Owner's negligence) including lost profits or other consequential loss arising from your use of or inability to use the Software or from errors or deficiencies in it whether caused by negligence or otherwise except as expressly provided herein.

7 Law

This Licence shall be governed by English law. No Easysoft distributor, agent or employee is authorised to make any amendment to this Licence.

If you have any questions concerning this Licence please write to:

Easysoft Limited
Thorp Arch Grange
Thorp Arch
Wetherby
LS23 7BA
England

Contacting Easysoft

You may request licences, report problems and ask for support by email, FAX, WWW, mail or telephone. Please include the following information (where relevant) when asking for support:

- who and where you are
- name and version number of the Easysoft product
- package and version number that creates problems
- server operating system version number
- whether similar problems have arisen previously
- connection protocol, and if TCP/IP, which company produced the protocol stack

Telephone

Support:	+44 (0)1937 863 450
Sales:	+44 (0)1937 860 000
Licensing:	+44 (0)1937 863 450

Telephone support and the licensing service are provided only during Monday to Friday (excluding public holidays) between the hours of 8:30 a.m. and 5:30 p.m. local UK time.

Due to Daylight Saving time (BST) add 1 hour to these times during the summer months (April to September). Exact dates vary from year to year. During winter months, local time is GMT.

FAX

Support:	+44 (0)1937 863 550
Sales:	+44 (0)1937 860 001
Licensing:	+44 (0)1937 863 550

Mail

Easysoft Limited
Thorp Arch Grange
Thorp Arch
Wetherby
LS23 7BA
England

Email

Support:	support@easysoft.com
Sales:	sales@easysoft.com
Licensing:	license@easysoft.com

Our web site and ftp site are available 24 hours a day. They offer free patches, upgrades, documentation and beta versions of Easysoft products. Some of the options may have restricted access.

World Wide Web

Technical support, product information and other items of interest are available on our Web site.

http://www.easysoft.com

FTP

Connect to:	ftp.easysoft.com
User:	anonymous
Password:	<your email address>

If you find any errors or omissions in this manual, or if you have any suggestions for improvement, please forward your comments by email (documentation@easysoft.com), FAX or mail to Easysoft (attention of Easysoft Documentation).

Contents

List of Figures

List of Tables

Welcome

Scope of Document

The primary functions of this manual are:

- to describe how to install Easysoft ODBC for RMS on your server and PC
- to describe how to manage data sources and drivers
- to describe how to define files which reside on a server so that they can be used by ODBC compliant applications
- to provide a reference to the Easysoft Administrator
- to describe the background to ODBC and the Easysoft system

Purpose of Software

Easysoft ODBC for RMS is a suite of software which is used by Windows applications to gain access to non-relational data which resides on server systems. There are three related software products:

- Easysoft Server Component. This converts queries generated by Windows products into a form that is suitable for accessing server files.
- Easysoft Client Component. This installs the Easysoft ODBC driver on the PC and allows you to set up data sources through which you can access server data.
- Easysoft Administrator. This is used to define files which reside on the server so that they can be accessed by ODBC compliant applications.

Audience

This manual is for you if you intend to do one or more of the following:

- install the Easysoft Server Component on your server. You do not need to know details of ODBC, but you should know the basics of OpenVMS and you should have the necessary privileges required to perform an installation.

- install the Easysoft Client Component on the PC. You don’t need to know about ODBC, but you should know the basics of Windows.

- using the Easysoft Client Component, set up one or more data sources which can be used to access data residing on the server. You need to know the transport protocol used to connect to the server, the server name and username and password details relating to the users who will use the data sources.

- install the Easysoft Administrator on your PC. You do not need to know about ODBC, but you should know the basics of Windows.

- using the Easysoft Administrator, define server files so that they can be accessed by ODBC compliant applications on your PC. You do not need to know about ODBC, but you should know the basics of Windows and you must know the file structure of the files that you wish to define.

Document Structure

The first chapter forms an introduction to ODBC and the Easysoft system.

The installation of the Easysoft Server Component is described in chapter two, and the Easysoft Host Administrator functions are defined in chapter three.

Chapters four and five deal respectively with the installation of the Easysoft Client Component on the PC platform and the creation of a data source. Chapter six is a tutorial which explains how to use an ODBC compliant application (Access) to access the Easysoft demonstration data on the server.

Chapter seven explains how to use the Easysoft Excel Macro for RMS.

A detailed installation tutorial for the Easysoft Administrator on the PC is given in chapter eight. Chapter nine is a step-by-step guide to using the Easysoft Administrator and chapter ten is a comprehensive reference to the Administrator.

The converter utilities are described in chapter eleven, and the final chapter addresses troubleshooting issues.

Document Conventions

This document uses the following typographic conventions.

`User input` Information that you must type in from the keyboard or that you must add to source code.

`Output` Screen output, prompts and source code.

Cancel Menu options. Also used for buttons in the Windows environment for example, .

<pathname> Angle brackets (<>) are used to indicate that you should substitute the appropriate information. For example, you would type a directory name in place of “pathname”.

Windows Editing

This section defines terms that relate to the editing of data in the Windows environment.

Text box (also known as an **entry field**): an area on a dialog box where you can enter text.

List box: shows a list of items that a user can select. If there are too many items to fit in the available space, a scroll bar appears.

Drop-down list box: a list box that initially shows only one item. The entire list is shown when the down arrow is selected.

Drop-down combo box: allows the user to select an item from a drop-down list; alternatively, text can be typed directly in the entry field.

If you use the keyboard to select items in a dialog box, you can move sequentially through the buttons by using the Tab key. Use alt-<character> to select an item, where character is the underlined letter on a button. For example, press **`alt-c`** to click the Copy button. If two buttons have the same underlined character, for example, **Copy** and **Close**, toggle between items by repeated pressing of the alt+<character> combination.

CHAPTER 1

Introduction

This chapter gives background information to Easysoft ODBC for RMS. The first major section deals with ODBC and the second with the Easysoft system itself.

Fundamentals of ODBC

ODBC allows interoperability between different Database Management Systems (DBMS) and applications. It works by supporting heterogeneous data access - applications access different data sources by using drivers which access the data. Applications can submit any SQL statement which is supported by a driver.

ODBC was developed by Microsoft to give a single API (Application Programming Interface) which can access a variety of data sources. This allows developers to write applications which are not targeted to any specific DBMS. End users connect applications to their databases by using add-in modules called drivers, which are available from database vendors and driver developers.

Successful operation of ODBC depends upon defining a set of standards common to all participating elements of a system; this is briefly discussed in the section entitled “Conformance” on page 4.

The overall Easysoft ODBC installation process on the PC is as follows. A program called setup is invoked which copies the Easysoft driver to a Windows system directory. It then installs the Microsoft ODBC Administrator (always supplied with drivers) if it doesn’t already exist on the system. This Administrator is started automatically, and the user tells it to install drivers (so that ODBC knows where / how to access them). Following this, data sources are configured - in other words, a relationship is defined between data and its associated DBMS, the operating system and the network. An application can now access the data even though it is not in the format that the application normally uses.

ODBC Architecture

The four components of the ODBC architecture are the Application, the Driver Manager, the Driver and the data source. Their relationship is shown in Figure 1, and their functions are described following this.

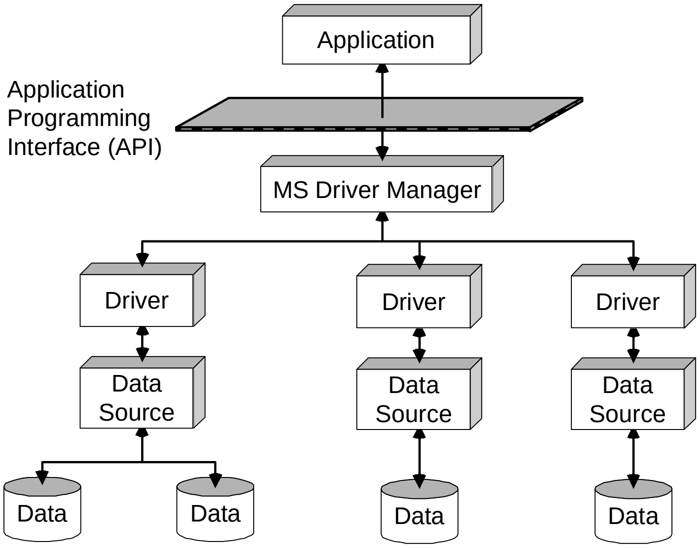


Figure 1.

ODBC Architecture

The ODBC Interface, which essentially defines the API and SQL syntax, is not part of the architecture, and therefore it is described in the section entitled “Conformance”, page 4.

An application, such as Lotus 1-2-3 or Microsoft Excel, calls ODBC functions which are sent to the driver via the Driver Manager. As far as the application is concerned the driver and Driver Manager appear to be a single functional unit. A single application may wish to access data from a number of different sources and ODBC allows this.

The Driver Manager (provided by Microsoft) loads drivers for an application when an application calls certain SQL functions and, if requested, traces calls and keeps a log file of these. Another function of the Driver Manager is to maintain a list of the drivers used by data sources.

The driver processes the function calls sent by the application, submits SQL calls to the data source and returns results to the application. It may change the syntax of the request to conform to the syntax used by the DBMS in the data source. Another function of the driver is to format errors into standard error codes and return these codes to the application.

In the context of ODBC a data source is more than just a set of data files. It includes the associated operating system and, if they exist, both the network and the DBMS which manages the data. Furthermore, a unique combination of data + operating system + DBMS + network may be mapped to many different data sources; for example, sales data may be accessible to a sales team and an accountant. The sales people can be given read/write access to the data, whereas the accountant can only read the data. An application may access more than one data source at any given time if it so requires.

Driver Types

Two basic types of driver are defined by ODBC. A single-tier driver processes data directly (that is, it processes both ODBC calls and resultant SQL statements) whereas a multi-tier driver sends SQL statements to the data source. This is shown in Figure 2.

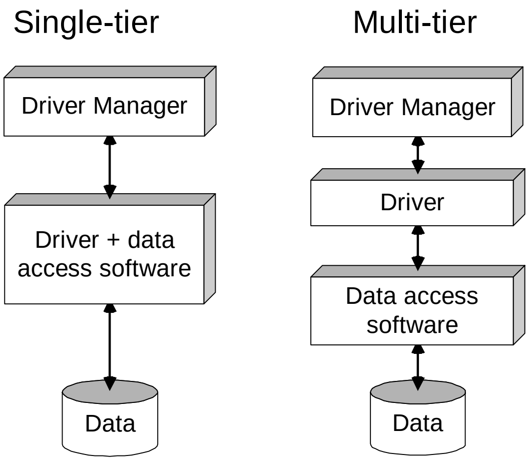


Figure 2.

Single- and multi-tier driver architecture

There is a difference between driver architecture and hardware architecture. A single-tier driver can run on a physical client-server hardware architecture (that is, the driver accesses data that is stored on a server, but the driver itself still sends SQL from the client to the server). Conversely, a multi-tier driver may reside on a single machine - the driver sends SQL requests to the data access software which processes these requests.

Typically, however, in a multi-tier configuration, the application, driver and Driver Manager reside on one machine (the client), and the database and data access software reside on a different system (the server).

Conformance

ODBC allows data communication between different applications by means of plug-in modules - in other words, drivers from many different vendors can be used. However, all DBMSs and applications provide different functionality and so for systems to communicate, it is necessary to define standards for functionality. ODBC defines conformance levels in two areas of functionality, the ODBC API and SQL grammar (including data types), and for both of these there are three levels of conformance, each one being more comprehensive than the level below.

The conformance levels for the API are named: core API, level 1 API, level 2 API.

The conformance levels for SQL are named: minimum SQL grammar, core SQL grammar, extended SQL grammar.

Selecting an appropriate level of conformance depends upon the needs of the application. Microsoft suggest that driver developers implement all level 1 ODBC API functions since many ODBC applications require this. Conforming to a given level does not mean that additional functionality cannot be provided, but if a driver is claimed to conform to a given level, then all the functionality of that level should be provided.

Conformance levels can be determined in a number of ways:

- from driver documentation
- call the SQLGetFunctions function from the Driver Manager
- if a driver supports the SQLGetInfo and SQLGetTypeInfo functions you can use these to return information on conformance. Although these are level 1 functions many core level drivers support them.

The Easysoft System

This section presents the global architecture of Easysoft ODBC for RMS. We look at the relationship between data sources, catalogs, databases and physical files and then the mapping between files and tables is analysed in more detail (it is assumed you are familiar with the basics of ODBC). Following this, the overall procedure for using Easysoft software is explained briefly.

Easysoft Architecture

Easysoft ODBC software in its entirety is used to connect ODBC compliant applications to files which reside on remote file servers. It has a multi-tier architecture in that the data access software resides on the server and the driver passes function calls to this. There are three related products (the Easysoft Server Component, the Easysoft Client Component and the Easysoft Administrator) which are used to give the required functionality. Figure 3 shows the logical architecture of these components and their relationship within the overall ODBC architecture.

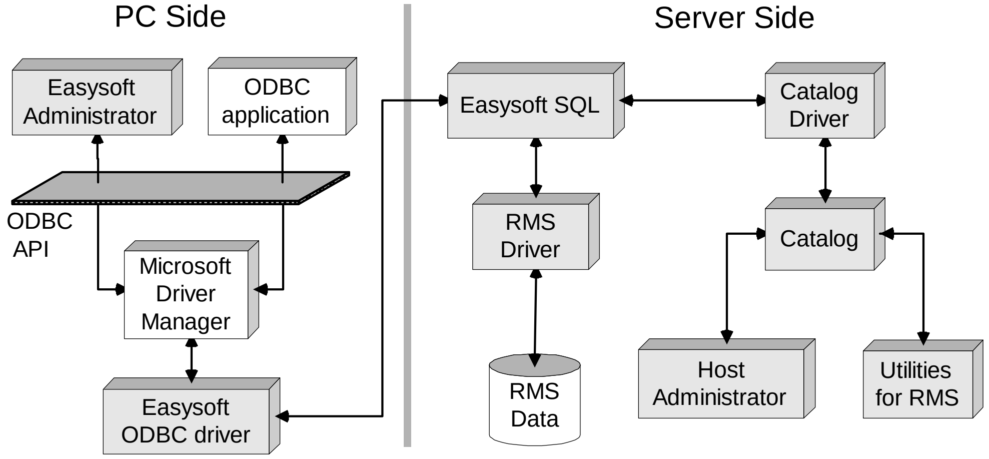


Figure 3. **Easysoft software architecture**

Easysoft Client Component. This resides on the PC and contains the Microsoft ODBC Administrator and the Easysoft ODBC driver. The Microsoft ODBC Administrator is used to install the Easysoft ODBC driver on the PC and to configure data sources.

Easysoft Administrator. This contains the software which is used on the PC to define server files for use through the Easysoft ODBC driver. This software presents server files in a relational format; it maps files to tables and fields to columns. In addition to various utility functions, the Administrator is used to define users and security privileges.

Easysoft Server Component. This contains a number of sub-components:

Easysoft SQL - deals with processing of SQL statements.

Host Administrator - runs on the host machine and is used to manage catalogs at a high level - primarily their creation. It is also used to administer licensing functions. Chapter three, “Host Administrator Reference” has full details.

Catalog - used to store information about the server files.

Catalog Driver - the mechanism for obtaining information on the structure of the data.

RMS Driver - used to access server data.

RMS Utilities -used as an aid to accessing data, for example, data dictionary converters.

Data Sources, Databases and Catalogs

Figure 4 indicates the relationship between data sources, databases and catalogs.

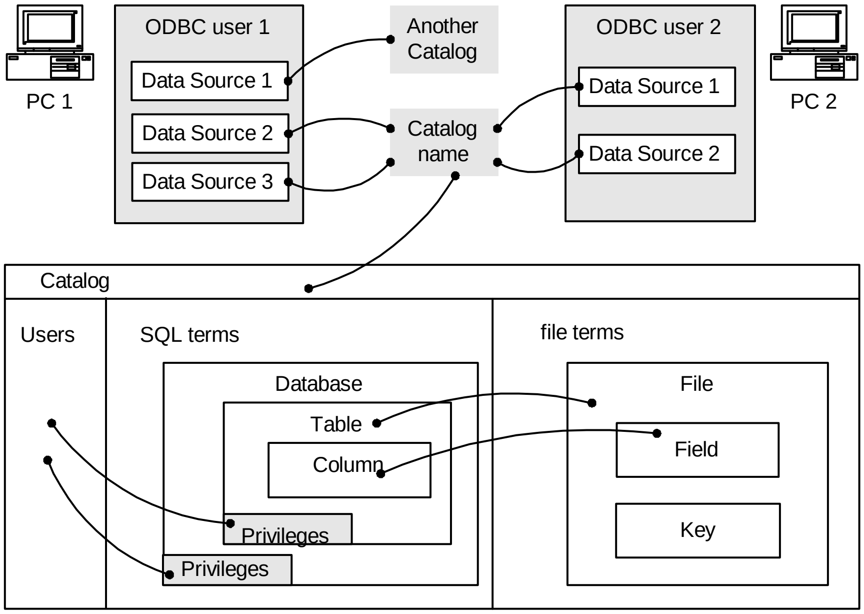


Figure 4. Relationships between components

Different data sources and databases are used to organise data - typically privileges for users and data functionality, for example, a different database for sales, personnel, development, etc.

Data sources are defined at the PC level. Each data source on a PC has a unique name. The data source defines which server is to be accessed, the ODBC driver to use when accessing the server data, the catalog associated with the data, network transport information and the user of the server.

To ensure ease of upgrading it is advisable to keep data source names on different PCs identical if they are to be used to access the same catalog.

SQL tables map to files and SQL columns map to fields. A database is a uniquely named collection of SQL tables, and before the mapping between files and tables can be made, there must be at least one defined database. Associated with tables and databases are privileges which allow access rights to be defined. The function of the catalog is to hold all this mapping information.

Defining Tables

The basic entities to consider when defining SQL tables are the physical files residing on disk, file definitions and table definitions.

A file definition might better be considered as a record definition since its function is to define the record structure of a physical file. There is no direct connection between a file definition and a physical file - we return to this in a moment.

Each database consists of a set of SQL tables: there are three essential components of a table definition, namely Table Name, File Definition and File Specification. Table Name is the name of the table as it appears in the SQL database, File Specification refers to a physical file on disk and File Definition is essentially a definition of a record structure. File definitions and file specifications are not tied to any one database; a single file specification can be used in many different databases.

There is a simple 1:1 mapping between a table and the column definitions for that table.

The mapping between physical files and SQL table definitions is not complex, although there are numerous options. At first sight, there appears to be redundancy; one file definition can be used in many table definitions, one file definition can map (via a table definition) to many physical files, many table definitions can use the same file specification (that is, physical file) and different file definitions can map onto one physical file. The reader might ask why there is not just simply a 1:1 mapping between file definitions and physical files; the remainder of this section deals with that question.

Look at the table definition in Figure 5; one table definition references one file definition and one physical file. This is straightforward; a single file definition is used to define the structure of the physical file.

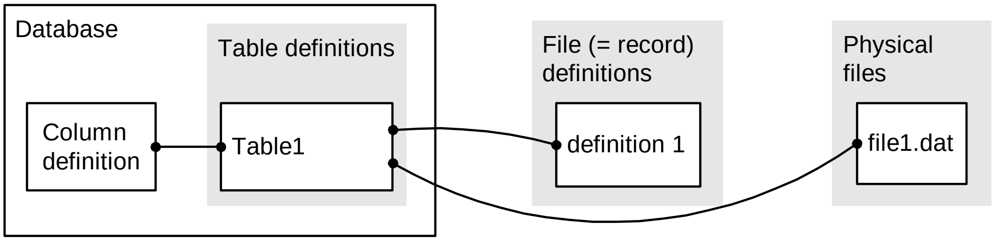


Figure 5. One file maps to one table

More complex mappings are possible. Consider table2 and table3 in Figure 6. Here, the File Definition (that is, definition of record structure) for both tables is the same and the file specification also is the same (file2.dat). The reason for this arrangement is that the column definitions for table2 and table3 can be different so that different users have different views of the data.

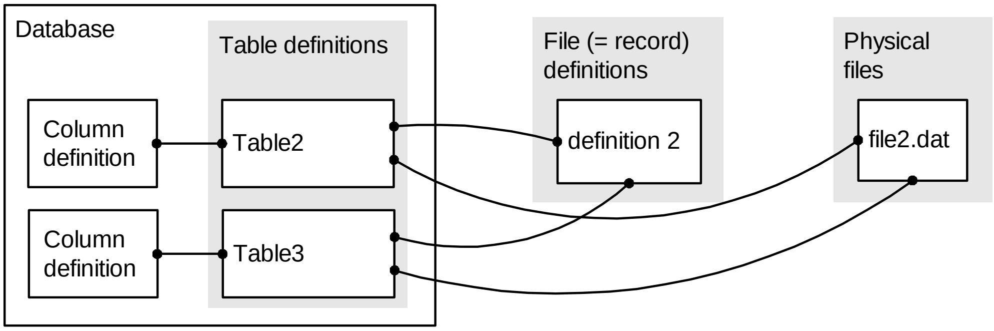


Figure 6. One file maps to two tables

Now consider the tables called table4 and table5 in Figure 7. Both use the same file definition, but they access different physical files. This implies that the physical structure of file3.dat and file4.dat is identical in the parts that are defined. One situation where this might arise is where one file is used in earnest for company data and the other is used for development purposes. Another possibility is that there are separate files for, say, data which is grouped on a yearly basis. An example which demonstrates that an entire file structure need not be defined is where, say, the first *n* fields are identical, but fields *n*+1 upwards are undefined for ODBC purposes and are possibly different.

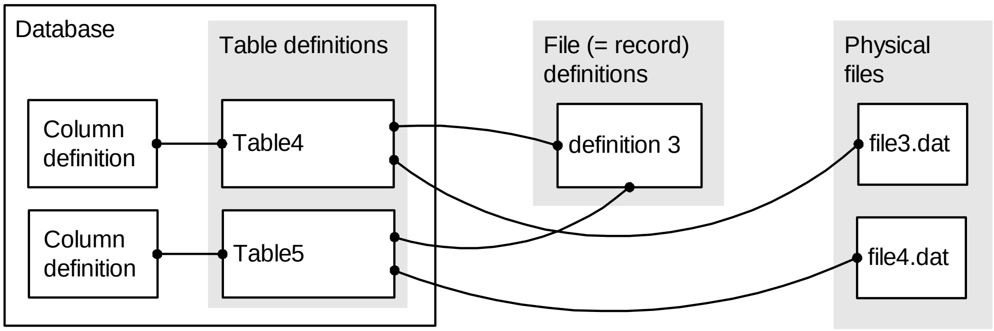


Figure 7. Different files have same record structure

Finally, consider Figure 8. Here one physical file has two different file definitions. Each of these definitions is used by a different table definition. The end result of this is the same as the case where two different table definitions use the same file definition and file specification. This case is another example showing that it is not necessary to fully define the fields in a file - you only need to define those you are interested in accessing. This configuration could be used, for example, where there is a non-relational file which contains both header and detail records. The header records could be defined in SQL terms using one file definition, and the detail records would be defined using another definition.

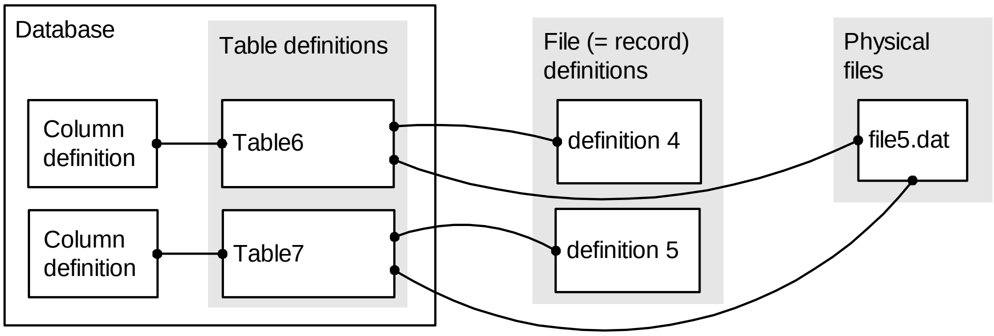


Figure 8. **Different file definitions used on the same file**

These are the basic options - the possible combinations are not discussed as the basic principles remain the same. One final comment is that in spite of all these examples, typically, there is a 1:1 relationship (using the mapping in Table Definition) between a file definition and a physical file.

Overall Process of using Easysoft Software

How exactly is the Easysoft system told of the existence and structure of the data files on the server, and what do you do to enable an ODBC-compliant application to access this data? The recommended procedure is:

1. Install the Easysoft Server Component on the server, license the software and then create a catalog using the Easysoft Host Administrator on the server.

2. Install the Easysoft Client Component on the PC and then define one or more data sources associated with a catalog.

3. Install the Easysoft Administrator on the PC and then define one or more databases for each of the data sources defined in step 2. For each database define the file to table mappings.

4. Use an application to access tables defined in step 3.

(Although strictly the installation and configuration processes need not be done at the same time it is recommended that the configuration order is followed - for convenience, the installation may as well be done immediately preceding the configurations).

Create a Catalog

Easysoft SQL needs a set of files in order to manipulate the data which resides on the server. Collectively, these files are known as the Easysoft Catalog. Data associated with a data source is defined in these Easysoft Catalog files, which hold, for example information on the location and structure of data files. A catalog can contain information on many different data sources.

You can have as many catalogs as you wish, but each catalog must reside in a separate directory. In most cases it is usual to have just one catalog to deal with all the data. However, there are valid reasons for having more than one catalog. For example, if two sets of unrelated files exist (say one dealing with live company data and the other with test data), then two catalogs may be appropriate.

Similarly, you may not want the Easysoft demonstration data - simply delete it and its associated catalog without affecting your data in any way.

A catalog is created using the Easysoft Host Administrator on the server. Note that it does not create directories - an existing directory must be specified, and if a suitable one does not exist, you must create it. The location and name of the catalog directory is unimportant, indeed, the catalog files can be in the same directory as the data files, although this poor housekeeping is not recommended! So, using the server operating system commands, create a directory for the catalog, and then using the Host Administrator create the catalog.

The Easysoft SQL installation creates two catalog directories - one for a catalog of demonstration data, and the other for the catalog which is to contain details of live data. The demonstration catalog is automatically created, but you must create the other catalog using the Host Administrator.

Define a Data Source

Now that a catalog exists, the Microsoft ODBC Administrator on the PC is used to configure data sources. A unique name is specified for a data source and then the location of the catalog associated with this data source is specified. (At this stage the transport protocol, server name, remote service must also be specified; passwords can either be specified at this stage, or later). The catalog eventually contains all the definitions of data, although at this stage it does not.

Now that a data source has been defined it is possible to define files on the server which can then be accessed by ODBC compliant applications.

Define a Database

The Easysoft Administrator is used to define server files so that they can be used by ODBC-compliant applications.   Once defined, it is possible to connect an ODBC-compliant application, such as Microsoft Access, to the server files. The application sees these files as if they were relational.

CHAPTER 2

Installation and Licensing on the Server

This first parts of this chapter explain how to install the Easysoft Server Component and license the software. The Maintenance section explains how to re-license the software and how to change Catalog passwords.

Steps for New Installations and Upgrades	**New**	**Upgrade**
Preparation		
1. Log on to server using System Manager’s account	✓	✓
2. Ensure 20,000 blocks available	✓	-
3. Availability of DEC C Runtime components kit	VAX only	-
4. Ensure no PC clients are using Easysoft software	-	✓
5. Ensure catalog uploaded from PC to server	-	✓
6. Special Instructions for Upgrades - 1.1.6 and Below	-	✓ (v1.1.6)
7. Transfer zipped files to server	✓	✓
8. Unzip files	✓	✓
Installation and related procedures		
9. Install the Easysoft Server Component	✓	✓
10. Create catalog	✓	-
11. Set quotas	✓	-
12. Edit system startup file	✓	-
Licensing		
13. Obtain Licence Key		
14. Generate your site number	✓	-
15. License product	✓	✓

The overall installation and licensing procedure for new installations and for upgrades is shown above. Required steps are indicated by a tick (✓). If one of the stages is not applicable to you, ignore it and continue with the next step.

Notes

When you upgrade on the server, you *must* also upgrade the Easysoft ODBC driver on the PC; the version number of these two products must be identical.

Currently there is no provision to un-install the software on the server platform.

Server platform versions:	VAX - VMS 5.5 or higher.
	Alpha - VMS 6.1 or higher.

When you upgrade the software on the server, the default catalog is automatically upgraded. In cases where either more than one catalog exists, or where the catalog is not in its default location, then use the UPGRADE CATALOG command (page 38).

The RMS driver (VAX-RMS, not to be confused with the Easysoft ODBC driver) supports fixed-length, variable-length , STREAM, STREAMLF, STREAMCR, VFC and UNKNOWN record formats.

Preparation

Step 1. Log on to Server

NEW

UPGRADE

The Easysoft Server Component is installed using the standard VMSINSTAL command procedure, normally located in the directory SYS$UPDATE. Log into any account capable of running VMSINSTAL and then set the default directory to SYS$UPDATE.

Step 2. Ensure 20000 Blocks Available

NEW

Before beginning the installation check that there are at least 20,000 free blocks (10 Mb, assuming the blocks are 512 bytes) available on the system disk. If you intend to install the software on a disk other than the system disk, check that there are at least 20,000 free blocks available on that disk also. Use the following DCL command to determine this:

```
$ SHOW DEV D
```

Step 3. Availability of DEC C Runtime

NEW

This version of Easysoft ODBC for RMS requires the DEC C Runtime library components.

- This step applies to VAX only; if you have an Alpha machine, continue with the next step.
- OpenVMS version 6.0 and above - this OpenVMS version already has the runtime component, so omit this step and continue with the next one.
- OpenVMS version 5.x - continue with this step.

If the DEC C components kit has not already been previously installed and if it is not installed at this stage, you will be asked later in the Easysoft installation (Step 9) whether you would like to install the full DEC C runtime kit or just use the minimum components of the DEC C kit as supplied by us.

Installing the full DEC C kit is the ideal solution, but any programs produced previously using the VAX C Runtime library (VAXCRTL) and which are re-linked (because, for example, they are being modified) will be linked with the DECCRTL and would therefore require any external third-party site running that new executable to have the DEC C kit installed. For this reason you might choose not to install the full kit and just install the minimum components (see Step 9); if this is the case, go to the next step.

To check whether the DEC C Runtime components kit is installed type:

```
$ DIR SYS$SHARE:DECC$SHR.EXE
```

DECC$SHR.EXE is the installed component, so if it exists, do nothing and continue with the next step, otherwise install the runtime component as shown below:

1. Copy the zipped file (DECCRTL.ZIP) to the SYS$UPDATE directory on the VAX (see Step 7 for an example)
2. Unzip the file (see Step 8 for an example)
3. Install the DEC C runtime software and follow the on-screen instructions:

   ```
   $ @VMSINSTAL AACRT060 SYS$UPDATE
   ```

Step 4. Ensure no PC Clients are using Easysoft Software

UPGRADE

Ensure that no PC clients are currently connected to the server via the Easysoft ODBC driver. To see a list of active users, start the Host Administrator and then run the SHOW USERS command (called SHOW CLIENTS in version 1.1.6 and below):

```
$ RUN EASYSOFT_SQL_ADMIN
ADMIN> SHOW USERS
User Name               Product Name                       Active
=================================================================
ANDREW                  <product name appears here>        YES
STEWART                 <product name appears here>        YES
```

In this case, users ANDREW and STEWART are active; before proceeding, ensure that they disconnect.

Step 5. Ensure Catalog Uploaded

UPGRADE

If you are upgrading the software and if, as recommended, the Easysoft Administrator will be re-installed on the PC, ensure that the local copy of the Easysoft Catalog on the PC has been uploaded to the server. This guarantees that the central catalog contains the latest information; when you first use the Easysoft Administrator after it has been re-installed, you will be prompted to download the Easysoft Catalog again.

Step 6. Special Instructions for Upgrades - 1.1.6 and Below

UPGRADE (V1.1.6 OR BELOW)

a) Shut Down Licence Server

If you are upgrading from version 1.1.6 or lower, you must shut down the Licence Server. Enter the following at the command prompt:

```
$ @EASYSOFT_SQL_SYSTEM:SHUTDOWN_LICENCE_SERVER
```

No messages are output when you run this command. The Licence Server is not used in this version, and it cannot be restarted.

b) Remove Installed Image

In version 1.1.6 and below it is possible to install Easysoft ODBC for RMS as an installed image. It is necessary to remove an installed image before installing a later version.

To check whether an installed image exists, type:

```
$ INSTALL LIST EASYSOFT_SQL_SERVER
```

```
DISK$NEWDATA:<easysoft.sql.system>.EXE
   SQLSRV;2         Open Hdr Shar
```

If you see any output from this command, then an installed image exists. To remove it type:

```
$ INSTALL REMOVE EASYSOFT_SQL_SERVER
```

Step 7. Transfer Files from Floppy Disk to Server

NEW

UPGRADE

Transfer the Easysoft Server Component archive(s) and the UNZIP.EXE binary (if it is not already on the server) from your PC into the SYS$UPDATE directory on your VAX or Alpha.

Note: This transfer must be done using binary transfer mode.

If you downloaded the software from our web site, then the files will be in some temporary location on your PC. If you received the software on disk, there may be one or more disks in the set of disks for the server, and on each disk there is one zipped file. The UNZIP.EXE file is on the first disk. In either case, the upload process is similar.

Examples of Transfer

Typically, you can use NFT with DECNET (if NFT is installed on your system, detailed instructions are available to you) or you can use FTP with TCP/IP from the command line. A few examples follow.

Transferring Files Using Pathworks

At the DOS \> prompt type:

```
NFT COPY/IMAGE <fname> <node>"<user> <pwd>" ::SYS$UPDATE:<fname>
```

Where
<fname> is the name of the file to transfer
<node> is the DECNET node number (if a name has been defined for the node, you can use the node name)
<user> is the user who is doing the installation (ideally the System Administrator)
<pwd> is the user’s password.

For example, to copy UNZIP.EXE to the node defined as *FREY* where the user is *MIKE* and the password is *BLAH234*, type the following (note there is no space between the node name and the quote mark.)

```
NFT COPY/IMAGE UNZIP.EXE FREY"MIKE BLAH234" ::SYS$UPDATE:UNZIP.EXE
```

Transferring Files Using TCP/IP

Open an FTP session to your server (this example is based on Microsoft FTP), then transfer the files.

```
ftp> CD SYS$UPDATE
ftp> BINARY
ftp> PUT <filename>
```

Repeat this last step (**PUT**) for all of the files.

Step 8. Unzip Files

NEW

UPGRADE

Ensure the default directory is SYS$UPDATE and then unzip the archives.

```
$ SET DEF SYS$UPDATE
$ UNZIP:=$SYS$UPDATE:UNZIP.EXE
$ UNZIP <filename>
```

Repeat this last step for all of the zipped files.

Upgrade note: If an old version of the installation save set resides on the disk, you are asked whether you want to overwrite the files. Answer yes each time this is asked.

```
Archive:  SYS$SYSDEVICE:[EASYSOFT.SQL.SYSTEM]<filename>.ZIP;1
replace <filename>.a? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
```

Installation

You must use the VMSINSTAL command procedure to install the Easysoft Server Component. Throughout, you are asked to supply information or answer questions requested by the installation procedure. In most cases a default value or answer is given and appears in brackets. To select the default value press **RETURN**. To select a different value, type it in and then press **RETURN**. If you do not supply all the parameters required, VMSINSTAL asks for them during the installation.

To cancel the installation procedure at any time, press **CTRL-Y**. If you need help on any of the questions asked, type a question mark (**?**) and VMSINSTAL displays help text for you to read.

If VMSINSTAL detects a problem during the installation procedure, it notifies you and asks you if you want to continue. To stop the installation and correct the problem, enter **NO**. After you have made the correction, you can restart the installation.

Step 9. Install the Easysoft Server Component

NEW

UPGRADE

1. Ensure the default directory is SYS$UPDATE and start the installation as follows:

   ```
   $ SET DEFAULT SYS$UPDATE
   $ @VMSINSTAL <file name> <directory>
   ```

   <file name> is the name of the file which contains the product you are installing. For VAX machines this is RMSVAXnnn and for Alpha (AXP) it is RMSAXPnnn, where nnn is a number which corresponds to the software version number.

   <directory> represents the name of the directory to which the Easysoft files were transferred. If you have followed the previous examples, then <directory> is SYS$UPDATE.

2. VMSINSTAL may then display information relating to the current state of the system. This information usually indicates who is logged into the system. VMSINSTAL asks whether you wish to continue, with the following prompt:

   ```
   * Do you want to continue anyway [NO] ?
   ```

   Enter **YES** if you want to continue (note that you must change the default), or **NO** to exit from the installation. It is acceptable to install Easysoft if other users are logged in to the server, but are not using Easysoft products. However, **if other users are using Easysoft, then you should not continue**.

3. VMSINSTAL then asks you whether you are satisfied with the backup of your system disk. This is a precautionary step, recommended by Digital, aimed at ensuring that you can recover from any unexpected side effects inadvertently caused by the installation procedure. The following message is output:

   ```
   * Are you satisfied with the backup of your system disk [YES] ?
   ```

   Enter **YES** if you want to continue, or **NO** to exit from the installation.

   If you enter **YES**, VMSINSTAL proceeds with the installation, and after a few minutes outputs a message stating the products which are to be processed along with a copyright notice.

4. If the DEC C Runtime components kit is *not* available, then the following message and options are presented.

   ```
   *** DECC Run Time Components Could Not Be Found ***
   The following options are available

     1)  Install the DECC Run Time Kit into the system library
         as described in the Easysoft ODBC for RMS manual
         This is the ideal option and will result in the
         best performance.

     2)  Use the DECC Run Time executable in the
         EASYSOFT_SQL_SYSTEM directory
         This is the best option if you wish to keep your existing
         VAXC runtime library.

     3)  Do not install the DECC Run Time Kit. However, you
         will not be able to run Easysoft ODBC for RMS

     Enter required option (1,2,3)[1]:
   ```

Select option 1 to exit the installation process. Then install the DEC C Runtime components kit (see Step 3). Then repeat this installation step.

Select option 2 to continue. DEC C requirements are automatically dealt with (this is not the same as installing the DEC C Runtime components kit).

Refer to Step 3 for a discussion of the consequences of these two options.

Select option 3 to exit the installation routine.

5. If you are installing the product into the same location as a previous version, VMSINSTAL displays the following prompt:

   ```
   * Do you want to purge the files replaced by this installation [YES] ?
   ```

   If you wish to retain an old version of Easysoft and you are installing this version on the same disk, enter **NO**, otherwise enter **YES**. The recommended answer is the default, **YES**.

6. Before loading the files, VMSINSTAL requests the following information:

   ```
   * Where is Easysoft SQL to be installed
   [SYS$SYSDEVICE:[EASYSOFT.SQL]]:
   ```

   If you do not want the Easysoft files to be located on the default device or directory, enter the name of the required device. For example, $USERDISK, or the required directory, for example, [PROGS.EASYSOFT]. If an earlier version of Easysoft exists, then the default shown is the location of the existing version.

7. VMSINSTAL now asks for a UIC:

   ```
   * What is the Easysoft SQL UIC [[777,1]]:
   ```

   If you do not want the Easysoft SQL files to have the UIC [777,1] enter a different UIC. If the UIC of 777 is already allocated, then a different number must be used.

8. VMSINSTAL now asks for the required network transport:

   ```
   * Which Network Transport are you going to use [DECNET]:
   ```

   This question refers to the network transport that is used by the client to communicate with Easysoft SQL; it may be any one of the following:

DECNET	(DEC PATHWORKS)
UCX	(DEC UCX)
TCPWARE	(Process TCPWARE)
MULTINET	(TGV Multinet)
PATHWAY	(Wollongong Pathway)

UCX, TCPWARE, MULTINET and PATHWAY are all implementations of TCP/IP.

9. VMSINSTAL now asks for a name for the network object to use with the specified protocol:

   ```
   * What is the name of the Remote Service [EASYSOFT]:
   ```

   This question refers to the name of the remote service that is used by the client to communicate with Easysoft on the server. The default value is EASYSOFT, but you can use anything that is convenient.

10. If DECNET was *not* selected as the network transport then the following question is asked, otherwise it is not:

    ```
    * What is the Port Number [7777]:
    ```

    This refers to the port number used by the host transport software to identify Easysoft SQL. The port number is required to set up a successful TCP/IP service on the OpenVMS server; if port 7777 is in use then select an appropriate replacement.

You have now answered all the required questions. During the remainder of the installation other messages may occur detailing the directories used by the installation process. VMSINSTAL now proceeds to restore the remaining files, displaying various messages.

Easysoft Directories on the Server

The following directories are created by VMSINSTAL if they do not already exist. The files are located in sub-directories on the device specified during the installation. The parent directory of these sub-directories is the one supplied during the installation.

Directory	Description	Logical name
[.CATALOG]	Default Easysoft Catalog	EASYSOFT_SQL_CATALOG
[.DEMO.CATALOG]	Demonstration Easysoft Catalog	EASYSOFT_SQL_DEMO_CATALOG
[.DEMO.DATA]	Demonstration system data	-
[.DOC]	On-line documentation	-
[.LICENCE]	Licence information	-
[.LOG]	Log files	-
[.SYSTEM]	System files	EASYSOFT_SQL_SYSTEM
[.TEMP]	Temporary files	-

Step 10. Create the Easysoft Catalog

NEW

For the *first* installation of Easysoft ODBC for RMS on the server you need to create the Easysoft Catalog (it is possible to have more than one) using the Host Administrator CREATE CATALOG command.

1. Start the Host Administrator if it is not already running:

   ```
   $ RUN EASYSOFT_SQL_ADMIN
   ```

   A short message is output, and then the ADMIN> prompt appears.

2.
   ```
   ADMIN> CREATE CATALOG
   _directory : EASYSOFT_SQL_CATALOG
   ```

   You are asked to provide the name of a directory in which the catalog is to be created - it is advisable to use the EASYSOFT_SQL_CATALOG logical for this. This logical is automatically set up during the installation process. (Since there can be only one catalog in any directory, if you want more than one catalog, you must use a different directory).

3.
   ```
   _password :
   _retype password :
   ```

   You are then asked for a password which becomes the catalog administrator's password for this Easysoft catalog. (The catalog administrator's username is automatically set to ADMIN). After you have re-typed the password for confirmation purposes notification of catalog creation is presented (only the first line is shown here).

```
Creating catalog in the directory EASYSOFT_SQL_CATALOG
```

Exit the Administrator by pressing the **ENTER** key.

Step 11. Set Quotas

NEW

For some users the default quota values may not be sufficient. Easysoft recommend that the following quotas are set.

ENQLM is set to at least 2000
PGFLQUOTA is set to at least 20000
CHANNELCNT is set to at least 256

Step 12. Edit the System Startup File NEW

Easysoft software requires a startup line in the system startup file. The name of the system startup file depends on the operating system version. For VMS version 5 the file is called SYS$MANAGER:SYSTARTUP_V5.COM. For OpenVMS version 6 the file is named SYS$MANAGER:SYSTARTUP_VMS.COM.

Set up system-wide logicals by adding the following lines at the end of the file before the line containing $EXIT (the hyphen at the end of the first line is the DCL continuation symbol):

```
$ @SYS$SYSDEVICE:[EASYSOFT.SQL.SYSTEM]STARTUP_SYSTEM -
      SYS$SYSDEVICE:[EASYSOFT.SQL]
```

Note: The directory SYS$SYSDEVICE:[EASYSOFT.SQL] used in this example should be replaced by the directory that contains the Easysoft files. (The directory is specified in response to the installation question which asks where Easysoft SQL is to be installed - see page 21).

Licensing

After the Easysoft software has been installed, it needs to be licensed, for which you need a Licence Key which we will provide.

Each software product requires its own Licence Key. You may obtain the Licence Key by email, fax or phone.

Phone:	+44 (0)1937 863 450
Fax:	+44 (0)1937 863 550
email:	licence@easysoft.com

Contact us *after* you have generated the Site Number (next step).

Note: A Licence Key is required

- for new installations of Easysoft software
- for upgrades of software
- where additional licences are required
- if the hardware configuration of an existing installation changes
- if a licence has expired or is being extended

Step 13. Generate Site Number

NEW

To generate your Site Number, run the Easysoft Server Configuration Routine. You must enter company information and contact details. After doing this the routine displays hardware information followed by your unique Site Number.

```
$ RUN EASYSOFT_SQL_SYSTEM:CONFIG

Easysoft Configuration Routine <version>.
<informational messages>

Company Name          : <your company name>
Address               : <address line 1>
                      :
                      :
Town/City             : <town or city>
County/State          : <county or state>
Postcode/Zip          : <post code or zip code>
Country               : <country>

Contact Name          : <contact name>
Telephone Number      : <country code> <city code> <number>
Fax Number            : <country code> <city code> <number>
Email Address         : <contact’s email address>
```

```
Node Name            : <name of your server>
Hardware Name        : <machine type>
Operating System     : <version number>

Site number          : <unique site number appears here>

Please contact Easysoft Limited with the above information to
receive product Licence Keys.

<Easysoft address details>
```

These lines are generated by Easysoft software.

Step 14. Obtain Licence Key

NEW

UPGRADE

To generate the Licence Key we use your Company Number (a reference number which we provide) and your Site Number (generated by the Server Configuration Routine). Give us all the information you entered in the previous step, plus the information which was automatically generated by the routine, plus the resultant Site Number, and we will give you the product Licence Key which you need for the next step.

Step 15. License Product

NEW

UPGRADE

1. To input the Licence Key, first start the Host Administrator:

   ```
   $ RUN EASYSOFT_SQL_ADMIN
   ```

   A short message is output, and then the ADMIN> prompt appears.

2. Run the ADD LICENCE command and enter your Company Number and Licence Key. Company Number is a reference number which we provide and which we use to identify your company. It only needs to be entered the *first* time you use this command. Licence Keys are in hexadecimal format, so the only valid characters are 0 to 9 and A to F (case is not important).

   ```
   ADMIN> ADD LICENCE

   Site Number            : <site number appears here>
   Company Number         : <company number>

   Enter Licence Key      : <enter the Licence Key here>

     Product Name         : <product name appears here>
     Product Version      : <version number appears here>
     Client Users         : <number of client users appears here>
     Server Users         : <number of server users appears here>
     Expiry Date          : <date>

   Enter Licence Key      :
   ```

If you want to license another Easysoft product, you can do so now by entering the Licence Key for that product at the prompt.

Alternatively, press the **ENTER** key to return to the ADMIN> prompt.

3. Once the Licence Key has been entered correctly, the software is licensed and ready for use. Quit the Host Administrator by using the EXIT command (or press the **ENTER** key):

```
ADMIN> EXIT
$
```

The installation procedures have been completed on the server, and you may now install the Easysoft Client Component and the Easysoft Administrator on the PC platform.

Maintenance

This section explains how to re-license the software and how to change catalog passwords.

Re-licensing

Licence Keys are tied to server specific information. Changes to the server configuration (for example, hardware model number) can result in invalidation of the Licence. Contact Easysoft, and you will be given a new Licence Key.

Licences are specific to a product version, so although you can continue to use a product for as long as the licence is valid, you will need a new licence to upgrade to the next version.

Twenty days (and fewer) before the licence will expire, each time that you connect to a data source a message is generated and shown on the PC stating that the licence is due to expire.

To re-license a product take the following steps:

1. Start the Host Administrator

```
$ RUN EASYSOFT_SQL_ADMIN
```

2. Ensure that no users are actively using a licence by running the SHOW USERS command:

   ```
   ADMIN> SHOW USERS

   User Name           Product Name                        Active
   ============================================================
   FRED BLOGGS         <product name appears here>         YES
   JANE BROWN          <product name appears here>         NO
   ```

   This step is not needed if a licence has already expired.

   To check which licences are available and when they expire, use the SHOW LICENCE command (see chapter 3).

3. Assuming that no users are using the licence you wish to renew, remove the licence by running the REMOVE LICENCE command:

   ```
   ADMIN> REMOVE LICENCE

   No.  Product Name                    Client   Server   Expiry
   =================================================================
   1    <product name appears here>     2        2        <date>
   2    <product name appears here>     2        2        <date>

   Select Licence Number : <select the licence to remove>

   **** WARNING LICENCE WILL BE PERMANENTLY DESTROYED! ****

   Continue with removal? (y/n) : y

   Please note that the following number <removal key>
   must be quoted before a replacement licence can be issued.
   ```

   Before a new licence can be issued the Removal Key must be quoted.

4. Send the Removal Key to Easysoft when you request a new licence key. We will use this when we generate a new Licence Key for the product.

5. Add the new Licence Key using the ADD LICENCE command (see Step 15 for an example).

Changing Catalog Passwords

The Catuser routine is used to change the password of existing Catalog users. It resides in the directory pointed to by the easysoft_sql_system logical. Any catalog user can run this routine, but there is a special option for the catalog administrator.

First, the general case is described, followed by the catalog administrator user options.

```
$ RUN EASYSOFT_SQL_SYSTEM:CATUSER

<informational message>

Catalog Directory : <enter directory name>
Catalog User      : <enter catalog user name>
Catalog Password  :
New Password      :
Verify Password   :

Catalog password successfully updated for user <user name>
```

Catalog Directory refers to the directory in which the catalog resides. Either type the name of the directory, or type a logical that points to the directory.

Catalog User refers to the user whose password you wish to change. If the specified user does not exist, an error message is generated. If the catalog administrator user is entered here, special conditions apply (see below). Catalog User equates to Catalog Login Username in the Easysoft ODBC Setup dialog box (see Appendix A).

Type the current Catalog Password. If the password is incorrect, an error message is generated. Catalog Password equates to Catalog Login Password in the Easysoft ODBC Setup dialog box (see Appendix A).

Enter the New Password. You must verify the new password. If the names do not match, the following message appears: `Passwords do not match` .You must re-enter both the new password and the verification.

You can quit the routine at any stage except the last one by entering a blank line. All the existing values are retained. However, if you get as far as entering a new password, then you must complete the process by entering the verification. If you enter a blank line at this stage, the verify password prompt re-appears.

Catalog Administrator Options

The catalog administrator has the option of changing passwords for all users.

```
Catalog Directory : EASYSOFT_SQL_DEMO_CATALOG

Catalog User      : ADMIN
Catalog Password  :

Administrator Options:

  1)  Change Administrator Password
  2)  Change User Password

  0)  Exit
```

Catalog administrator's name (always "ADMIN") and password.

```
Enter option number [1, 2 or 0] - 2

Catalog User      : <name>
New Password      :
Verify Password   :
```

Name and new password of non-administrator user.

```
Catalog password successfully updated for user <name>

Catalog User      :
```

If, at the Catalog User and Password prompts, the catalog administrator details are entered, three options are presented.

Option 1 behaves exactly as described in the previous section.

Option 2 (shown in the example) allows the catalog administrator to change catalog passwords for any user defined in the catalog. Press the **Enter** key at the Catalog User prompt to quit the routine.

Option 3 (labelled 0) is used to quit the routine at this point.

CHAPTER 3

Host Administrator Reference

This chapter describes the Easysoft Host Administrator, which is used to manage catalogs at a high level and to manage licences.

During installation, a logical, EASYSOFT_SQL_ADMIN, is set up to point to the Host Administrator. To start the Host Administrator type the following at the DCL prompt:

```
$ RUN EASYSOFT_SQL_ADMIN
```

The Host Administrator is executed and following the display of version information and a copyright notice, the `ADMIN>` prompt is displayed.

Each of the Administrator commands is now explained in detail. Apart from the HELP command itself, they are in alphabetic order, not the order as shown on the HELP screen. Where a directory is requested in the command prompts, use double square braces, **`[]`** , if you want to access the current directory.

HELP

This command displays the help page shown below, which lists the available commands and gives a brief description of each command.

```
ADMIN> HELP

COMMANDS

 HELP            - Display this page.
 CREATE CATALOG  - Create system catalogs. A directory name is
required.
                   If no directory name is entered, the catalog will
be
                   created in the current directory.
 UPGRADE CATALOG - Upgrade a catalog to the current version.
 IMPORT CSV      - Import catalogs in the EASYSOFT CSV format.
    /DB=db1[,...]- Import only the specified databases
                   (Default is to import all databases).
 EXPORT CSV      - Export catalogs to the EASYSOFT CSV format.
    /DB=db1[,...]- Export only the specified databases
                   (Default is to export all databases).

 SET COMPANY     - Set the company details.
 SHOW COMPANY    - Display the company details and specific site
number.
```

```
ADD LICENCE     - Licence a new product.
REMOVE LICENCE  - Remove a product licence.
SHOW LICENCES   - Display licensed products.

SHOW USERS      - Display a list of registered users/clients.
REMOVE USER     - Remove a registered user.
EXIT            - Exit the administrator.
```

ADD LICENCE

This command is used to enter new licence information for the machine on which the server software is installed. Licence Keys are in hexadecimal format, so the only valid characters are 0 to 9 and A to F (case is not important).

```
ADMIN> ADD LICENCE

Site Number         : <site number appears here>
Company Number      : <enter company number here>

Enter Licence Key   : <enter the Licence Key here>

  Product Name          : <product name appears here>
  Product Version       : <version number appears here>
  Client Users          : <number of Client users appears here>
  Server Users          : <number of Server users appears here>
  Expiry Date           : <date>

Enter Licence Key   :
```

If more than one Easysoft product is to be licensed, repeat the process, otherwise press the **ENTER** key to return to the ADMIN> prompt.

The first time that you use this command, you must enter the Company Number (a reference number which we provide). On subsequent use of the command, the Company Number is automatically entered.

If an existing licence is being changed then all users using the licence must be inactive (that is, not connected to any data sources). If any user is active then an error message is returned. To see the status of any user run the SHOW USERS command.

The number of client users (also referred to as “clients”) indicates the number of PCs on which the Easysoft ODBC software is installed. The number of server users indicates how many connections can be made to the server. Each time a client connects to a data source on the server, the client’s machine details are saved in the licence file. Once the maximum number of users has been reached any further attempts to connect to the server from other machines are refused. The current status of clients can be seen by running the SHOW USERS command.

CREATE CATALOG

This command creates a catalog which is then used when accessing data files. A catalog can be created in any directory, provided that the directory exists. If it does not exist, a message is generated, and the directory prompt re-appears. The password which is entered is the password which is later used by the ADMIN user when accessing the catalog (see Appendix A). The password has to be retyped to ensure integrity.

```
ADMIN> CREATE CATALOG

_directory : <catalog directory>
_password :
_retype password :

Creating catalog in the directory <catalog directory>
<list of tables created>

Catalog created successfully.
```

EXIT

This command is used to exit the administrator. Alternatively, hitting the **RETURN** key at the `ADMIN>` prompt achieves the same effect.

EXPORT CSV

EXPORT CSV copies the catalog definitions to a text file containing a CSV description of the catalog. The command syntax is:

export csv <file to export> <catalog directory> {/db=<db list>}

<file to export> is the name of the file into which to write the CSV definitions. <catalog directory> is the directory in which the catalog resides.

The /db parameter allows a selection of databases from the catalog to be exported rather than exporting all the databases. The syntax of the parameter is /db=<db list> where <db list> is a list of one or more databases separated by commas. The curly brackets indicate that the parameter is optional.

```
ADMIN> EXPORT CSV DEMOFDX.CSV <catalog directory> /db=demo
_password :
Checking catalog version in directory <catalog directory> ...
Catalog version <version> is valid. Export resuming ...
<informational messages>
```

```
Total number of output lines     - <number>
```

IMPORT CSV

IMPORT CSV is used to import catalog definitions which are contained in a CSV text file. The command syntax is:

import csv <file to import> <catalog directory> {/db=<db list>}

<file to import> is the name of the file which contains the CSV definitions which are to be imported.
<catalog directory> is the directory in which the catalog resides.

The /db parameter allows a selection of databases to be imported rather than importing all the databases defined by the CSV file. The syntax of the parameter is /db=<db list> where <db list> is a list of one or more databases separated by commas. The curly brackets indicate that the parameter is optional. See “EXPORT CSV” for an example of the use of /db.

```
ADMIN> IMPORT CSV DEMOFILE.CSV <catalog directory>
```

```
_password :
Checking catalog version in directory <catalog directory> ...
Catalog version <version> is valid. Import resuming ...
<informational messages>
Database updated successfully
```

Usernames and passwords in the imported catalog may contain spaces. Database names in the imported catalog should not contain the underscore character (_).

On import, the default directory (in the Database Definition) and the file specification (in the Table Definition) are not converted to upper case.

REMOVE LICENCE

This command removes a licence. The following example removes the first licence in the list. **Note**. Ensure you keep a record of the Removal Key that is generated by this command, as a new Licence Key will not be given without this.

```
ADMIN> REMOVE LICENCE
```

```
No.  Product Name                    Client   Server   Expiry
================================================================
1    <product name appears here>     2        2        <date>
2    <product name appears here>     2        2        <date>
```

```
Select Licence Number : 1
```

```
**** WARNING LICENCE WILL BE PERMANENTLY DESTROYED! ****

Continue with removal? (y/n) : y

Please note that the following number <removal key> must be quoted
before a replacement licence can be issued.
```

A licence can only be removed when all users using the licence are inactive (that is, not connected to any data sources). If a user is active then an error message is returned. To see the status of any user use the SHOW USERS command.

REMOVE USER

This command removes a user to free a client licence slot. If all slots have been taken by users then no more users can access server data via Easysoft ODBC. If a licence slot is cleared by removing a user who is no longer using Easysoft ODBC then another user can access the required data. The following example removes user “JOE SMITH”.

```
ADMIN> REMOVE USER

No.  User Name          Product Name                       Active
=================================================================
1    FRED BLOGGS        <product name appears here>        NO
2    JOE SMITH          <product name appears here>        NO
3    JANE BROWN         <product name appears here>        NO

Select User Number : 2
```

Only inactive (that is, not connected to any data sources) users can be removed. An attempt to remove an active user results in an error message.

SET COMPANY

This command is used to keep a record of company details, some of which are used in the licensing operation.

```
ADMIN> SET COMPANY

Company Name          : <your company name>
Address               : <address line 1>
                      : <address line 2>
                      : <address line 3>
Town/City             : <town or city>
County/State          : <county or state>
Postcode/Zip          : <postcode or zipcode>
Country               : <country>
```

```
Contact Name          : <contact name>
Telephone Number      : <phone number>
Fax Number            : <fax number>
Email Address         : <email address>
```

SHOW COMPANY

This command displays company details (entered using the SET COMPANY command) and hardware information. Site number is automatically generated by our software. The output of this command should be sent to Easysoft when you require a Licence Key.

```
ADMIN> SHOW COMPANY

Company Number        : <your company number>
Company Name          : <company name>
Address               : <address line 1>
                      : <address line 2>
                      : <address line 3>
Town/City             : <town or city>
County/State          : <county or state>
Postcode/Zip          : <postcode or zipcode>
Country               : <country>

Contact Name          : <contact name>
Telephone Number      : <phone number>
Fax Number            : <fax number>
Email Address         : <email address>

Node Name             : <node name>
Hardware Name         : <hardware name>
Operating System      : <operating system>

Site Number           : <unique site number>
```

SHOW LICENCES

This command displays licence information for the machine on which the server software is installed.

```
ADMIN> SHOW LICENCES

Product Name                          Client   Server   Expiry
================================================================
<product name appears here>           <n>      <n>      <date>
<product name appears here>           <n>      <n>      <date>
```

<n> indicates the maximum number of clients and servers which can be used with the licence.

SHOW USERS

This command displays details of users licensed on the machine on which the server software is installed. The Active field indicates whether a user is active or inactive.

```
ADMIN> SHOW USERS

User Name               Product Name                       Active
=================================================================
FRED BLOGGS             <product name appears here>        YES
JOE SMITH               <product name appears here>        NO
JANE BROWN              <product name appears here>        NO
```

A single client machine, which is identified by its machine name and node address, can have multiple connections open to the server which are classed as a single concurrent connection. For example, a user on a client machine could be using a database, a spreadsheet and a reporting tool simultaneously accessing three different data sources on the server but only taking up one concurrent connection.

Sometimes a client is connected to the server even though no data is being read or written; the client is taking up a concurrent connection even though no work is being done. This scenario arises when a client has connected to a data source on the server, retrieved or updated some information, and has not closed the client application. To enable other clients to use the concurrent connection the client must close the document (for example, report, database, spreadsheet) which accessed the data thus closing the concurrent connection on the server. Whether or not a connection is closed after an ODBC request is application dependent.

When a client connects to Easysoft SQL, connection details are passed from the client machine to the host server. Before the host server can access any data the connection has to be verified. Unique information is taken from the PC and is used in allocating a client licence slot on the server.

UPGRADE CATALOG

This command upgrades catalogs created with older versions of the software to the latest version. This ensures that data integrity is maintained and that the Easysoft SQL engine works correctly with the catalog.

When you upgrade the software on the server, the default catalog is automatically upgraded; this command only needs to be used in cases where either more than one catalog exists, or where the catalog is not in its default location (as defined by the EASYSOFT_SQL_CATALOG logical).

```
ADMIN> UPGRADE CATALOG

_directory : <catalog directory>
_password :

Checking catalog version in directory <catalog directory> ...
Catalog is version <version>, this will be upgraded to Version
<version>
Catalog upgraded successfully.
```

CHAPTER 4

ODBC Driver Installation

This chapter describes the installation of the Easysoft ODBC Driver.

Easysoft ODBC requires Windows 95 or above and unless there are differences which need to be considered, these platforms are collectively referred to as “Windows”.

Easysoft ODBC storage requirement:	5 megabytes
Easysoft ODBC memory requirement:	4 megabytes

To upgrade from a previous Easysoft version, follow the procedure outlined for installing the software.

Note: When you upgrade the ODBC driver, you must also upgrade the Easysoft Server Component on the server and the version number of these two products must be identical.

Starting the Driver Installation

There are three ways to obtain the Easysoft ODBC Driver software:

- The Easysoft web site is available 24 hours a day at http://www.easysoft.com for downloads of definitive releases and documentation. Select Download from the Easysoft ODBC-RMS Driver section of the web site. Log in. (If you have not yet done so, you need to register first. On the registration form, an asterisk (*) indicates that a field is mandatory.) From the download page, choose the client platform release that you require.
- The Easysoft FTP server is available 24 hours a day at ftp://ftp.easysoft.com, containing free patches, upgrades, documentation and beta releases of Easysoft products, as well as definitive releases. Change to the pub/download/client/ directory and then choose the platform release that you require.
- You can order Easysoft software on CD by email, telephone or post (see Contacting Easysoft).

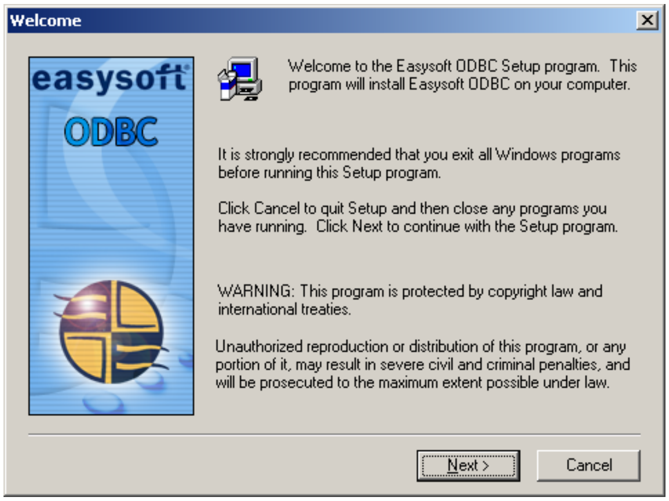
The name of the Easysoft ODBC Driver distribution file is of the form:

- odbc-*x*.*y*.*z*-windows-x86.exe

where "*x*" is the major version number, "*y*" is the minor version number and "*z*" is the build index.

Driver Installation Steps

Execute the file distribution that you obtained from one of the sources described earlier in this chapter. There will be a short delay while Setup prepares the wizard to guide you through the installation procedure. Then the Easysoft ODBC (Welcome) dialog box appears (Figure 9).



Welcome to the Easysoft ODBC Setup program. This program will install Easysoft ODBC on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

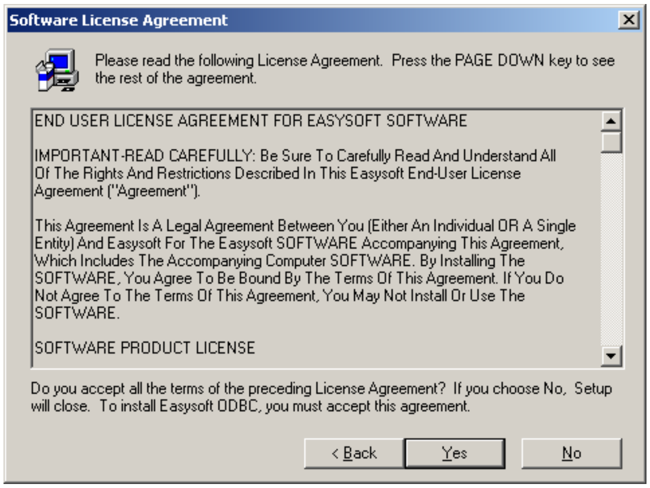
WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Figure 9.

ODBC Setup (Welcome) dialog box

Click Next to continue. The Software License Agreement dialog box is displayed (Figure 10).



Please read the following License Agreement. Press the PAGE DOWN key to see the rest of the agreement.

END USER LICENSE AGREEMENT FOR EASYSOFT SOFTWARE

IMPORTANT-READ CAREFULLY: Be Sure To Carefully Read And Understand All Of The Rights And Restrictions Described In This Easysoft End-User License Agreement ("Agreement").

This Agreement Is A Legal Agreement Between You (Either An Individual OR A Single Entity) And Easysoft For The Easysoft SOFTWARE Accompanying This Agreement, Which Includes The Accompanying Computer SOFTWARE. By Installing The SOFTWARE, You Agree To Be Bound By The Terms Of This Agreement. If You Do Not Agree To The Terms Of This Agreement, You May Not Install Or Use The SOFTWARE.

SOFTWARE PRODUCT LICENSE

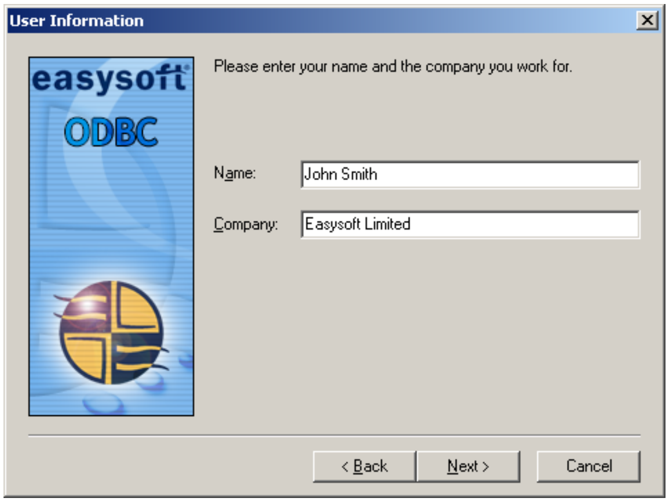
Do you accept all the terms of the preceding License Agreement? If you choose No, Setup will close. To install Easysoft ODBC, you must accept this agreement.

Figure 10.

Software License Agreement dialog box

After you have read the License Agreement, click Yes to confirm your acceptance of its terms. If you do not accept the terms of the agreement, click No and then click Exit Setup to exit the installation.

The User Information dialog box (**Figure 11**) is displayed.



User Information

Please enter your name and the company you work for.

Name: John Smith

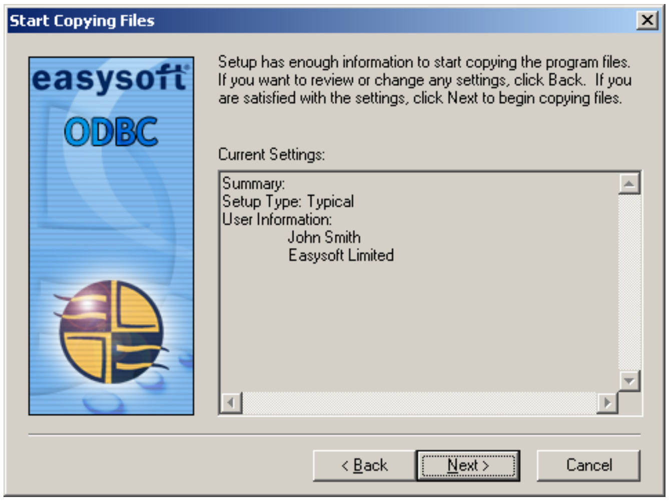
Company: Easysoft Limited

Figure 11.

User Information dialog box

Type your name and company in the spaces provided. Click Next.

In the Start Copying Files dialog box (Figure 12), click Next.



Start Copying Files

Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

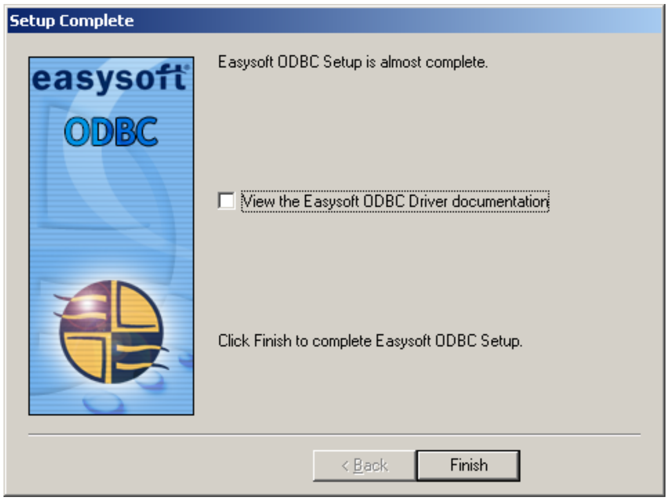
Current Settings:

Summary:
Setup Type: Typical
User Information:
John Smith
Easysoft Limited

Figure 12.

Start Copying Files dialog box

In the Setup Complete dialog box (**Figure 13**), click Finish to return to Windows. To display the Easysoft ODBC Driver online Help, click the check box.



Setup Complete

Easysoft ODBC Setup is almost complete.

□ View the Easysoft ODBC Driver documentation

Click Finish to complete Easysoft ODBC Setup.

Figure 13.

Setup Complete

The installation successfully terminates, and the default Microsoft ODBC Administrator icon is created. Typically, the ODBC Administrator icon can be found in the Control Panel, or under the Start menu.

Use ODBC Administrator to add your ODBC data sources. For more information, see Chapter 5, “Managing Data Sources.”

CHAPTER 5

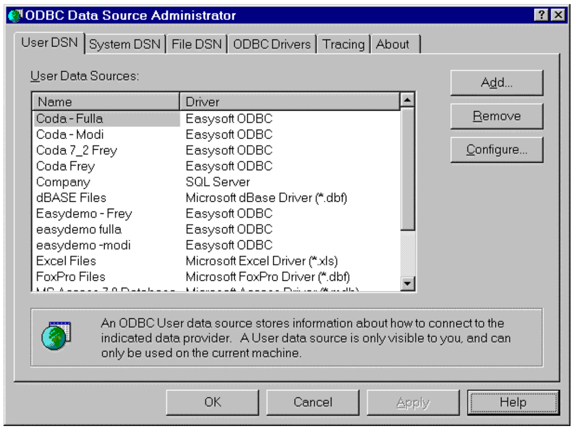
Managing Data Sources

This chapter explains how to configure the data sources you intend to use. A detailed walk-through of most options is shown, the data you need to input to the system is explained, and some of the more common problems encountered are explained. The working example shows how to create a data source called EASYDEMO which for testing purposes can be used to access Easysoft demonstration data that is set up during the installation on the server.

To configure data sources the Microsoft ODBC Administrator must be running.

Microsoft ODBC Administrator

This section describes the Microsoft ODBC Administrator.



User Data Sources:

Name	Driver
Coda - Fulla	Easysoft ODBC
Coda - Modi	Easysoft ODBC
Coda 7_2 Frey	Easysoft ODBC
Coda Frey	Easysoft ODBC
Company	SQL Server
dBASE Files	Microsoft dBase Driver (*.dbf)
Easydemo - Frey	Easysoft ODBC
easydemo fulla	Easysoft ODBC
easydemo -modi	Easysoft ODBC
Excel Files	Microsoft Excel Driver (*.xls)
FoxPro Files	Microsoft FoxPro Driver (*.dbf)
MS Access 7.0 Database	Microsoft Access Driver (*.mdb)

An ODBC User data source stores information about how to connect to the indicated data provider.   A User data source is only visible to you, and can only be used on the current machine.

Figure 14.

User DSN tab

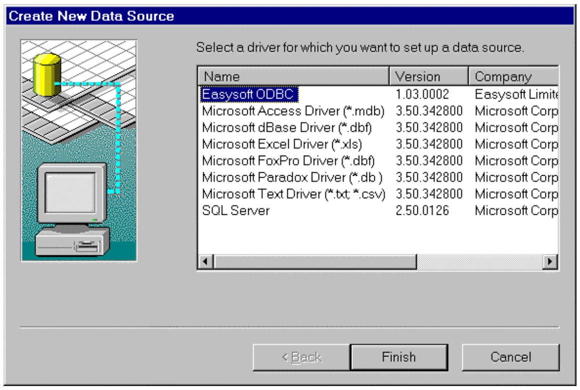
The following tabs are available on the Microsoft ODBC Data Source Administrator:

Tab	Function
User DSN	Add, remove and configure user data sources. A user data source is available to a specific user on a machine (compare System DSN).
System DSN	Add, remove and configure system data sources. A system data source is available to all users on a machine (compare User DSN).
File DSN	These are file-based data sources that can be shared between all users that have the same drivers installed. This option is not applicable to Easysoft ODBC for RMS and was not available in previous versions of the Microsoft ODBC Administrator.
ODBC Drivers	Displays information about the installed drivers.
Tracing	Set trace options for the ODBC Driver Manager.
About	Information about the core components of the Microsoft ODBC. For information about drivers, look under the ODBC Drivers tab (Figure 21). This tab is not described in this chapter.

Adding a Data Source

The operation of the User Data Source dialog box and the System Data Source dialog box is identical. Here we describe how to create a user data source. To create a system data source, complete a similar process using the System DSN tab.

Click the **User DSN** tab (Figure 14) on the Microsoft ODBC Data Source Administrator and then click the **Add...** button. The Create New Data Source dialog box (Figure 15) is displayed.



Create New Data Source

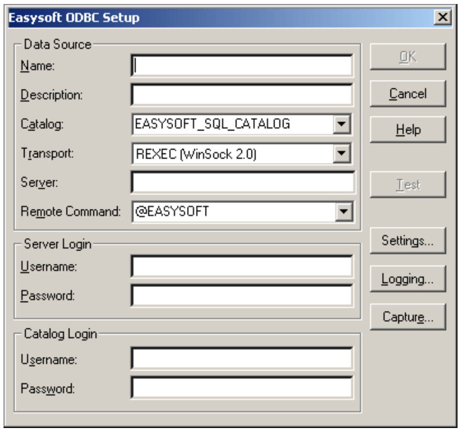
Select a driver for which you want to set up a data source.

Name	Version	Company
Easysoft ODBC	1.03.0002	Easysoft Limite
Microsoft Access Driver (*.mdb)	3.50.342800	Microsoft Corp
Microsoft dBase Driver (*.dbf)	3.50.342800	Microsoft Corp
Microsoft Excel Driver (*.xls)	3.50.342800	Microsoft Corp
Microsoft FoxPro Driver (*.dbf)	3.50.342800	Microsoft Corp
Microsoft Paradox Driver (*.db)	3.50.342800	Microsoft Corp
Microsoft Text Driver (*.txt; *.csv)	3.50.342800	Microsoft Corp
SQL Server	2.50.0126	Microsoft Corp

Figure 15.

Create New Data Source dialog box

The list box shows all the installed drivers. Select Easysoft ODBC and click the **Finish** button. The Easysoft ODBC Setup dialog box (Figure 16) appears.



Easysoft ODBC Setup

Data Source
Name:
Description:
Catalog: EASYSOFT_SQL_CATALOG
Transport: REXEC (WinSock 2.0)
Server:
Remote Command: @EASYSOFT

Server Login
Username:
Password:

Catalog Login
Username:
Password:

Figure 16.

Easysoft ODBC Setup dialog box

Transport.

This field name changes, depending upon the option chosen for Transport.

The **OK** and **Test** buttons are initially greyed out. **OK** is enabled when data is entered in the Name and Server entry fields. **Test** is enabled when Catalog, Transport, Server and Remote Service/command/object are specified. It is possible to use the **Test** button even if **OK** is greyed out.

The Remote entry field name defaults to one of the following, depending upon the Transport previously selected:

- Remote Service
- Remote Command
- Remote Object

Table 1 on page 49 describes the information you have to provide in order to successfully configure the data source and the Easysoft demonstration defaults are given in Table 2 on page 50 so you can have a guaranteed working example to test.

Table 1. Information required for the Easysoft ODBC Setup

Name The name of a data source. This name appears in the ODBC connection dialog box on connection to a data source. The name may contain spaces (Table 5 on page 156 lists invalid characters).

Description A more descriptive name for the data source (optional).

Catalog The directory where the Easysoft Catalog resides on the server. This can be specified using either a logical or a full path name (including device name).

The Easysoft ODBC setup on the PC is independent of the Server Component installation, and it caters for all of the products which can be installed on any server. Therefore, some of the options may not be valid for your server.

The Easysoft ODBC for RMS Server Component installation sets up two logicals:
`EASYSOFT_SQL_CATALOG` points to the default catalog and
`EASYSOFT_SQL_DEMO_CATALOG` points to the catalog for the demonstration data

Transport The network transport to be used. You are presented with the options available to your PC.

connection method	protocol
REXEC	Windows sockets 2.0
TCP/IP	Windows sockets 2.0
DECNET	Pathworks 4.1 upwards

Server The name of the server on which the source data resides.

Remote Service / Command / Object The name of the service to connect to on the server. Remote Service and Remote Object map to a command to run the software. Remote Command is the command to run in order to access the software.

Server Login Username The user name used to connect to the server. Case is ignored.

Server Login Password The password associated with the server login username. Case is ignored.

Catalog Login Username The user name that provides access to the catalog. Case is ignored.

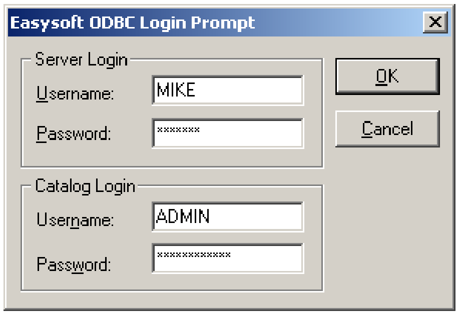
Catalog Login Password The password associated with the Catalog login user name. Case is ignored.

Table 2. Easysoft Demonstration Defaults

Name	EASYDEMO
Description	Easysoft demonstration (You can type anything you like)
Catalog	EASYSOFT_SQL_DEMO_CATALOG (Select this from the available options)
Transport	REXEC (Recommended option - select from available options)
Server	The name of the server
Remote Service /Command/Object	Select one of the available options
Server Login Username	Your user name for the server
Server Login Password	Your password for the server
Catalog Login Username	ADMIN This is provided automatically after you select the demonstration catalog
Catalog Login Password	ADMIN This is provided automatically after you select the demonstration catalog

Complete the Easysoft ODBC Setup dialog box using Table 2 as a guide. Click the **Test** button to validate the information, including the server username and password. The Easysoft ODBC Login Prompt (Figure 17) appears by default; if you don't want it to appear in future, you can change the setting (details are in the "Settings" section on the following page). Click **OK** to continue with the test. If the test is successful, a dialog box appears stating this; click **OK** to continue. If the test is unsuccessful an appropriate message is generated.

After the test is successful, click on **OK** to save the new data source and return to the Data Sources dialog box.



Server Login
Username: MIKE
Password: *******

Catalog Login
Username: ADMIN
Password: ************

Figure 17.

Easysoft ODBC Login Prompt

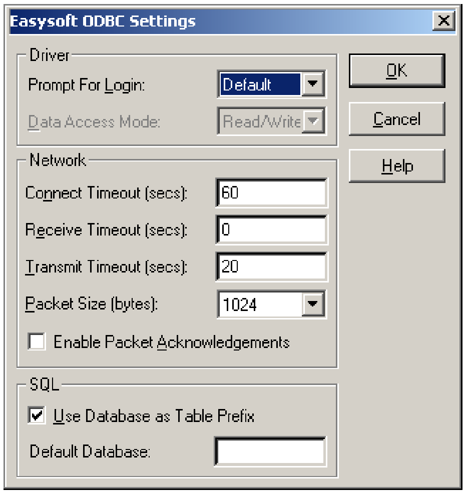
Server and catalog usernames and passwords can be different.

The **OK** button is greyed out if either of the Usernames is not specified.

For the purposes of setting up the Easysoft demonstration, you do not need to deal with the **Settings...**, **Logging...** or **Capture...** options.

Settings

The Easysoft ODBC Settings dialog box (Figure 18) is accessed by selecting **Settings...** on the Easysoft ODBC Setup dialog box (Figure 16). It is used to set driver, network and SQL options.



Easysoft ODBC Settings

Driver

Prompt For Login: Default

Data Access Mode: Read/Write

Network

Connect Timeout (secs): 60

Receive Timeout (secs): 0

Transmit Timeout (secs): 20

Packet Size (bytes): 1024

□ Enable Packet Acknowledgements

SQL

☑ Use Database as Table Prefix

Default Database:

Figure 18.

Easysoft ODBC Settings dialog box

Table 3. Information required for Easysoft ODBC Settings

Prompt For Login	Controls the display of the Easysoft ODBC Login Prompt when a connection is made to a data source. **Default**: display is defined by the application. **Never**: dialog box is never displayed. For successful operation, ensure that login information is supplied in the Easysoft ODBC Setup dialog box. **Always**: dialog box is always displayed even if login information is contained in Easysoft ODBC Setup dialog box.
Data Access Mode (not yet supported)	Allows data to be either read/write or read only.

Connect Timeout The attempt to connect is cancelled if it is not successful within the specified time.

Receive Timeout The connection is cancelled if a reply is not received within the specified time. **Note**: the value zero means wait forever.

Transmit Timeout The connection is cancelled if the network layer cannot send the packet in the specified time.

Note: the timeouts for each option are identical for both client and server.

Packet Size Size of the data packets used to transmit data over the network. Valid options are 128, 256, 512, 1024 (default), 2048, 4096, 8192, 16384, 32768.

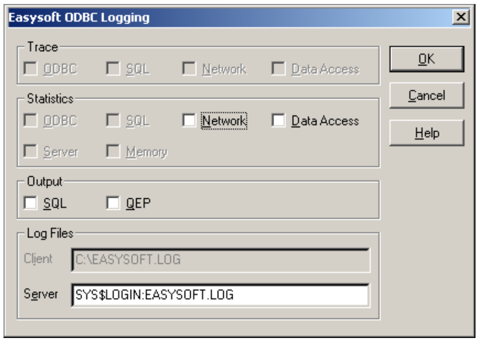
Enable Packet Acknowledgements This option checks to see if data has been received correctly. The default is OFF. If network problems are experienced it can be turned ON. Using this option slows down the transmission of the data.

Use Database as Table Prefix In the case where a catalog contains information on two or more databases, and where those databases contain tables with the same name, this option allows you to differentiate the tables. For example, if there are two SALES tables, one in a database called MINE and one in a database called THEIRS, then you are able to connect to both of these using the data source. You would see the tables as MINE_SALES and THEIRS_SALES.

Default Database If the Use Database as Table Prefix option is not selected, then a default database must be specified, even if only a single database exists. If there is more than one database in the catalog, only the one specified with this option will be accessible.

Logging Options

The Easysoft ODBC Logging dialog box (Figure 19) is accessed by selecting **Logging...** on the Easysoft ODBC Setup dialog box (Figure 16). It is used to control the logging options that are described in Table 4.



Trace: □ ODBC □ SQL □ Network □ Data Access

Statistics: □ ODBC □ SQL □ Network □ Data Access □ Server □ Memory

Output: □ SQL □ QEP

Log Files

Client: C:\EASYSOFT.LOG

Server: SYS$LOGIN:EASYSOFT.LOG

Figure 19.

Easysoft ODBC Logging dialog box

Table 4. Easysoft ODBC Logging Information

(Only available options are described)

Trace ODBC, Trace Network These can be enabled using the EASYSOFT.INI file. Refer to the [LOGGING] option in “Explanation of Sections”, page 158

Network Statistics about data transmitted and received.

Data Access Lists the files accessed for a given query and the number of I/Os to files for that query.

SQL Lists the SQL query which is passed to the server.

QEP Lists the Query Execution Plan.

Log file (Server) Path and name of server file to which logging information is sent. Default can be changed.

Using logging can reduce performance considerably, so in general, you would only enable the log file options if there were problems accessing data.

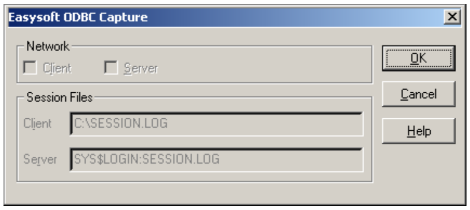
For each ODBC session, the log contains information relating to the querying of the Easysoft catalog. Then, each time a query is sent to the server, the log is appended until you disconnect from the data source, so if you send many queries, you will finish with a long log containing entries for many queries. To aid clarity, delete the Easysoft log file (by default called EASYSOFT.LOG) before sending the query you wish to inspect. After the query has been sent, the log file relates to just that single query.

The log files on the server are not purged, so if you use logging regularly, you should remove old unwanted log files.

Examples of log files are shown in the appendix entitled, “QEP Scoring Mechanism”, page 198.

Capture Information

The Easysoft ODBC Capture dialog box (Figure 20) is accessed by selecting **Capture...** on the Easysoft ODBC Setup dialog box (Figure 16). Capture is an Easysoft diagnostic tool which logs all data sent and received. The resulting log files can be used to replay a given scenario. Checking **Client** enables a log for the client and checking **Server** enables a log for the server. Session Files for client and server are the paths and names of the log files. File paths and names are given by default - you can change them if you wish. The capture functions are not yet available.



□ Client □ Server

Client: C:\SESSION.LOG

Server: SYS$LOGIN:SESSION.LOG

Figure 20.

Easysoft ODBC Capture dialog box

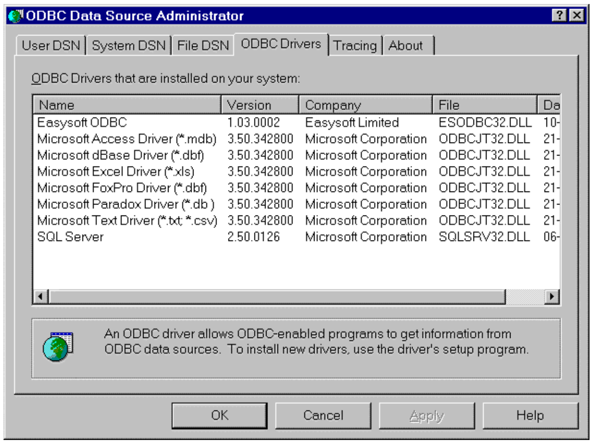
Removing or Modifying a Data Source

To remove a data source, highlight it and click **Remove**. A dialog box appears asking for confirmation. Click **Yes** to delete the data source, **No** to cancel the operation.

To modify a data source, highlight it and click **Configure...**.

Drivers Tab

The ODBC Drivers tab (Figure 21) gives information about drivers.



ODBC Drivers that are installed on your system:

Name	Version	Company	File	Da
Easysoft ODBC	1.03.0002	Easysoft Limited	ESODBC32.DLL	10-
Microsoft Access Driver (*.mdb)	3.50.342800	Microsoft Corporation	ODBCJT32.DLL	21-
Microsoft dBase Driver (*.dbf)	3.50.342800	Microsoft Corporation	ODBCJT32.DLL	21-
Microsoft Excel Driver (*.xls)	3.50.342800	Microsoft Corporation	ODBCJT32.DLL	21-
Microsoft FoxPro Driver (*.dbf)	3.50.342800	Microsoft Corporation	ODBCJT32.DLL	21-
Microsoft Paradox Driver (*.db)	3.50.342800	Microsoft Corporation	ODBCJT32.DLL	21-
Microsoft Text Driver (*.txt; *.csv)	3.50.342800	Microsoft Corporation	ODBCJT32.DLL	21-
SQL Server	2.50.0126	Microsoft Corporation	SQLSRV32.DLL	06-

An ODBC driver allows ODBC-enabled programs to get information from ODBC data sources. To install new drivers, use the driver's setup program.

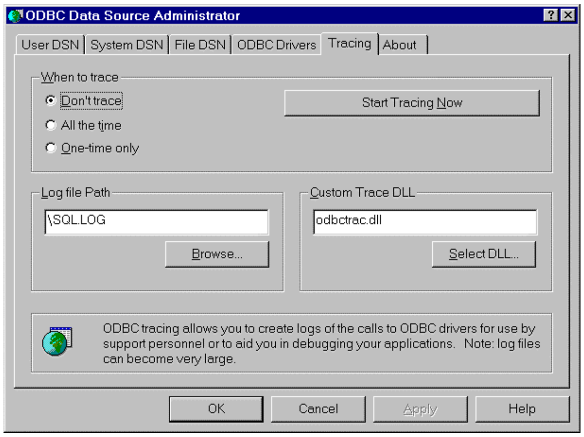
Figure 21.

ODBC Drivers tab

Each line in the list box shows the driver name, followed by the version, company, name of the driver file and the date it was produced.

Tracing Tab

The Tracing tab (Figure 22) is used to specify how the ODBC Driver Manager traces calls to ODBC functions.



When to trace

- ● Don't trace
- ○ All the time
- ○ One-time only

Log file Path: \SQL.LOG

Custom Trace DLL: odbctrac.dll

ODBC tracing allows you to create logs of the calls to ODBC drivers for use by support personnel or to aid you in debugging your applications. Note: log files can become very large.

Figure 22.

Tracing tab

When to trace

These options can only be set when there is no connection.

Don’t trace: disables tracing.

All the time: tracing is performed for all connections at all times.

One-time only: tracing is performed for the next connection, then disabled.

Start Tracing Now

Enables dynamic tracing, which is performed as long as the dialog box is open. Dynamic tracing can be enabled whether or not a connection is open. When this option is selected, the button is replaced with Stop Tracing Now. When Stop Tracing Now is selected, or when the ODBC Administrator dialog box is closed, dynamic tracing is disabled.

Log file Path

Displays the path and file where the log information will be stored. You can change the path and file name by editing the entry box, or by using the **Browse** button.

Custom Trace DLL

If you prefer to use your own custom DLL to perform the tracing operation, replace the default file with the name of your file.

Click the **Apply** button to accept changes without closing the dialog box, or click **OK** to make changes and close the dialog box.

CHAPTER 6

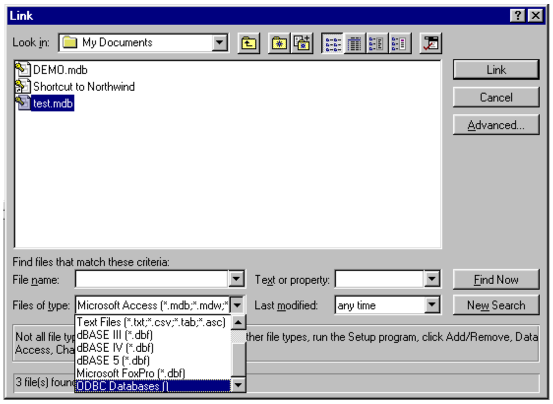
Using Easysoft ODBC

Using the EASYDEMO data source, this chapter gives a step-by-step example of how to access server data using Microsoft Access 7.0, which is an ODBC compliant application. To work through this example, both the Easysoft Server Component and the Easysoft Client Component must be installed.

(After you have worked through the Easysoft Administrator tutorial (chapter 8) you can use the table that you defined there (MYPROD); this is not done here, as we wish to show an example that all users can work through).

1. Start Microsoft Access and either open an existing database or create a new one.

2. From the **File** menu select **Get External Data** followed by **Link Tables...** The Link dialog box (Figure 23) appears.



Look in: My Documents

Files of type: Microsoft Access (*.mdb;*.mdw;*

Last modified: any time

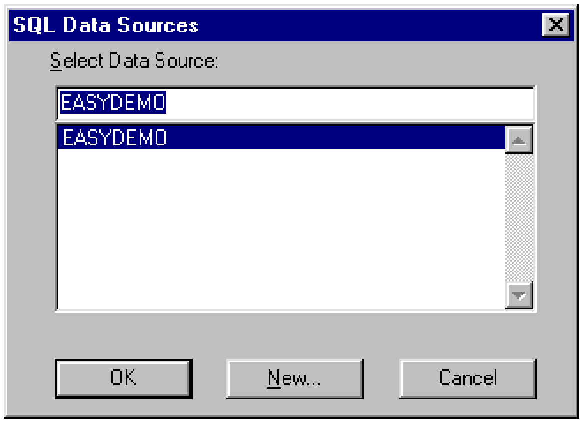
Not all file typ ther file types, run the Setup program, click Add/Remove, Data
Access, Cha

3 file(s) foun

Figure 23.

Link dialog box

3. Select the database to use, then from the Files of type list box select **`ODBC Databases()`**. Click the **Link** button. The SQL Data Sources dialog box (Figure 24) appears.



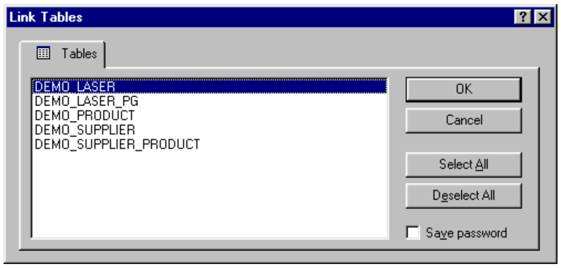
Select Data Source:
EASYDEMO

EASYDEMO

Figure 24.

SQL Data Sources dialog box

4. Select **EASYDEMO** as the data source and click **OK**. (If the Easysoft ODBC Login Prompt dialog box appears, click **OK**). The Link Tables dialog box (Figure 25) appears.

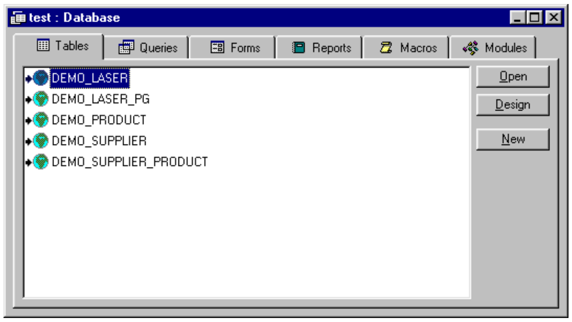


DEMO_LASER
DEMO_LASER_PG
DEMO_PRODUCT
DEMO_SUPPLIER
DEMO_SUPPLIER_PRODUCT

Figure 25.

Link Tables dialog box

5. Select one or more tables to link, and click **OK**. The tables are linked, and when the operation is complete the Database dialog box (Figure 26) contains a list of linked tables.

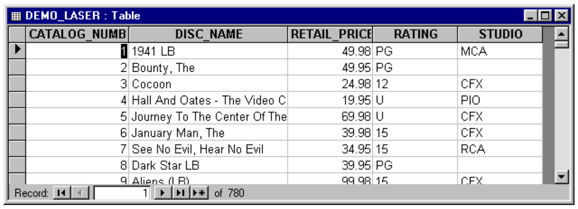


DEMO_LASER
DEMO_LASER_PG
DEMO_PRODUCT
DEMO_SUPPLIER
DEMO_SUPPLIER_PRODUCT

Figure 26.

Database dialog box

7. Double clicking on a table displays the data. You can select subsets of the data and add and delete records. The changes you make are updated to the file on the server which relates to the table you are changing. Double click one of the tables to see Easysoft demonstration data in Microsoft Access (Figure 27).



CATALOG_NUMB	DISC_NAME	RETAIL_PRICE	RATING	STUDIO
1	1941 LB	49.98	PG	MCA
2	Bounty, The	49.95	PG	
3	Cocoon	24.98	12	CFX
4	Hall And Oates - The Video C	19.95	U	PIO
5	Journey To The Center Of The	69.98	U	CFX
6	January Man, The	39.98	15	CFX
7	See No Evil, Hear No Evil	34.95	15	RCA
8	Dark Star LB	39.95	PG	
9	Aliens (LB)	99.98	15	CFX

Figure 27.

Demonstration data in Access

CHAPTER 7

Easysoft Excel Macro for ODBC

The Easysoft Excel Macro for ODBC (also known as the “RMS macro”) allows you to download data, create reports and upload data to RMS with the minimum of effort.

The software is available for all currently supported PC versions of Microsoft Excel.

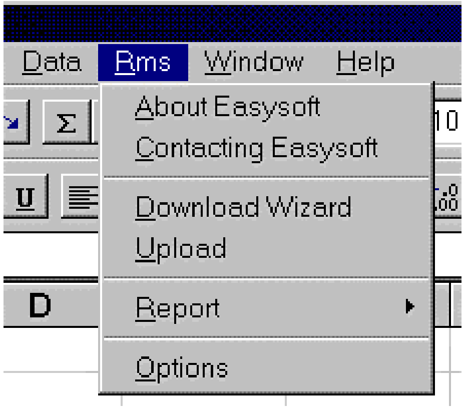


Figure 28.

ODBC macro menu options

Use this the first time you open the macro.

The **Rms** menu option on the menu bar gives access to the top level menu shown here.

Before the macro can be used, it must be installed, and this is described in the next section. Then the following topics are discussed: downloading data, uploading data, creating reports and finally, troubleshooting.

Use the **About Easysoft** option to obtain version information about the macro.

Use the **Contacting Easysoft** option to list the support contact options.

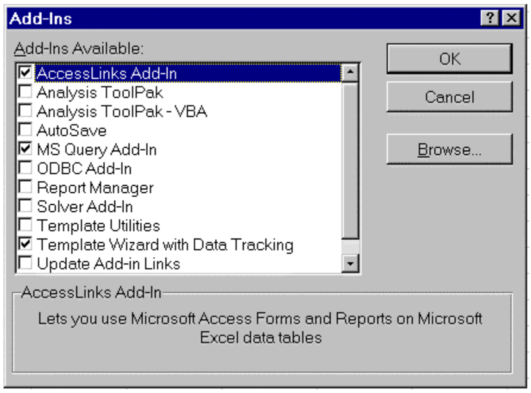
Macro Installation

To install the macro, you must add the software as an Excel add-in. Take the following steps:

1. Copy the macro to a convenient location (the recommended location is c:\easysoft\excel).

   If you obtain the software from the Easysoft web site your web browser should allow you to download and save the file in the newly created directory. If you obtain the software on floppy disk, copy it to the directory.

2. Start Microsoft Excel. Select **Add-Ins...** from the **Tools** menu. The Add-Ins dialog box appears.



☑ AccessLinks Add-In
☐ Analysis ToolPak
☐ Analysis ToolPak - VBA
☐ AutoSave
☑ MS Query Add-In
☐ ODBC Add-In
☐ Report Manager
☐ Solver Add-In
☐ Template Utilities
☑ Template Wizard with Data Tracking
☐ Update Add-in Links

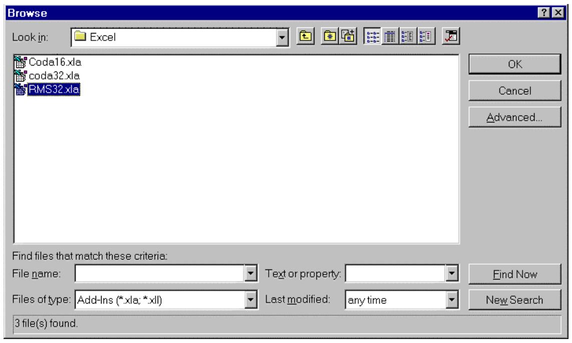
AccessLinks Add-In

Lets you use Microsoft Access Forms and Reports on Microsoft Excel data tables

Figure 29.

Add-ins dialog box

3. Click the **Browse...** button, navigate to the c:\easysoft\excel directory, and highlight the .XLA file. Click the **OK** button.



Look in: Excel

Coda16.xla
coda32.xla
RMS32.xla

File name:

Text or property:

Files of type: Add-Ins (*.xla; *.xll)

Last modified: any time

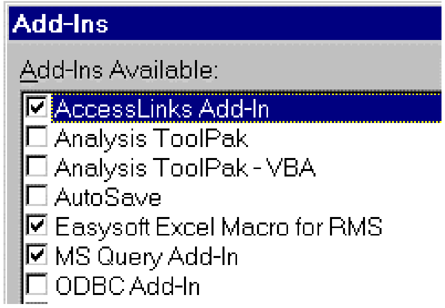
3 file(s) found.

Figure 30.

Browse dialog box

4. The Add-in now appears in the Add-Ins Available list box.

The Easysoft macro is now accessible.



Add-Ins

Add-Ins Available:

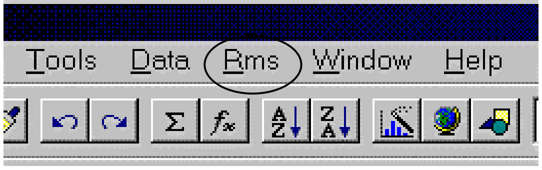
☑ AccessLinks Add-In
□ Analysis ToolPak
□ Analysis ToolPak - VBA
□ AutoSave
☑ Easysoft Excel Macro for RMS
☑ MS Query Add-In
□ ODBC Add-In

Select the RMS add-in.

Note: if there are any other Easysoft macros available, ensure that they are not selected. Conflicts can arise is you attempt to work with more than one Easysoft macro at the same time.

5. Click the **OK** button on the Add-Ins dialog box.

6. The macro is now available for use by clicking the newly added **Rms** menu which appears on the Excel menu bar.

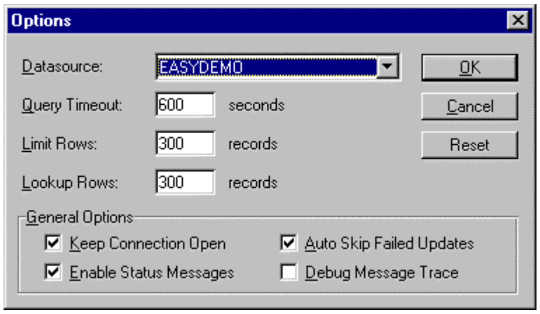


*New **Rms** option*

Initialisation

The first time the macro is used on a PC this initialisation step must be completed. It only has to be done once (unless you want to change the options); the information is saved, and when you, or another user, uses the macro in future, the information is already available for use.

1. To set the initialisation options, select **Options** from the **Rms** menu. The Options dialog box appears. The various settings you can choose are described in the remainder of this section.



Datasource: EASYDEMO

Query Timeout: 600 seconds

Limit Rows: 300 records

Lookup Rows: 300 records

General Options

☑ Keep Connection Open ☑ Auto Skip Failed Updates

☑ Enable Status Messages ☐ Debug Message Trace

Figure 31.

Options dialog box

2. **Datasource**: select the required data source from the dropdown list box (in these examples we use the EASYDEMO data source which contains demonstration data). Only data sources which use Easysoft drivers are shown in the dropdown list.

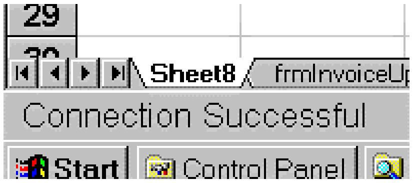
3. **Query Timeout** refers to the maximum length of (clock) time that the query is allowed to take up. If the query is not completed within this time, it will be cancelled. The default value of 600 seconds can be changed if required.

4. **Limit Rows** is used to limit the number of rows initially returned by the query. If the query result contains more rows than the number specified, you will be asked whether or not you wish to see these. The default value of 300 rows can be changed if required.

5. **Lookup Rows** is not enabled in this version of the macro. The number in the box has no effect on the operation of the macro.

6. **Keep Connection Open** is used to keep the connection to the data source in an open state. This can speed up the accessing of data, but the trade-off is that if many users want to access the RMS data, there may not be enough licence slots available. By default, this option is selected.

7. **Enable Status Message** is used to show messages at the bottom of the worksheet, for example:



Connection Successful

This is where you will see messages.

By default, this option is selected.

8. **Auto Skip Failed Updates** is used to prevent on-screen error messages which result from attempts at invalid updates being shown. However, at the end of the update operation, all rows which the user wanted to update, but which were not updated will be flagged on the worksheet. By default, this option is selected.

9. **Debug Message Trace**. If problems are being experienced, this option should be selected. It results in messages being displayed on screen. By default, this option is not selected.

10. When all the information has been entered, click the **OK** button to accept it. Control passes back to Microsoft Excel, and you can now use the macro functions, which are described in the following sections.

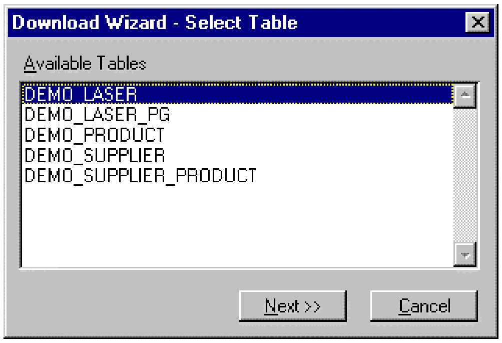
    Click the **Cancel** button to return to Excel without saving any changes you may have made. If you have not previously initialised the macro, you will not be able to use it.

    Reset is not enabled in this version of the macro. Clicking the button has no effect on the operation of the macro.

Download Data

The Download Wizard consists of a series of dialog boxes which you should work through in sequence. When you have completed each dialog box, you should click the **Next>>** button to move onto the next one. If you change your mind about an earlier entry, then click the <<**Back** button(s) to move back to an earlier dialog box. This step-by-step guide shows you then entire process. At the end of this section there are explanations of the more advanced options which are available.

1. To download data, select **Rms**, **Download Wizard**. The Table Download dialog box appears. Select the table you wish to download (this example uses the LASER table), then click the **Next>>** button..

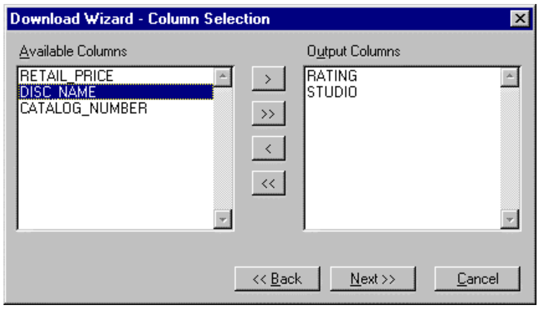


DEMO_LASER
DEMO_LASER_PG
DEMO_PRODUCT
DEMO_SUPPLIER
DEMO_SUPPLIER_PRODUCT

Figure 32.

Select Table dialog box

2. The Column Selection dialog box appears. From the Available Columns list, highlight each column you wish to see, then click the > button (you cannot select multiple columns). If you want to see all the columns, then just click >> without highlighting them. The column(s) will be moved to the Output box. Click the **Next>>** button.



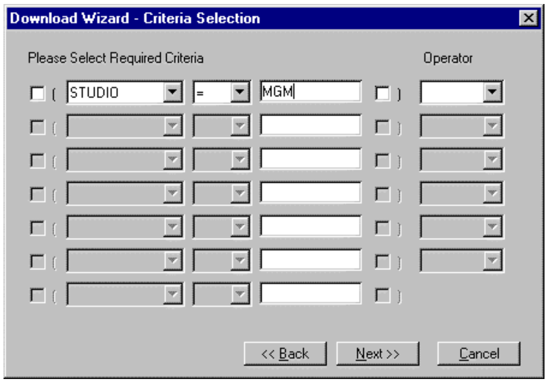
Available Columns: RETAIL_PRICE, DISC_NAME, CATALOG_NUMBER

Output Columns: RATING, STUDIO

Figure 33.

Column Selection dialog box

3. The Criteria Selection dialog box appears. Enter the criteria you require (if any), then click the **Next>>** button (complex criteria are discussed on page 70). Here we select laser discs from the MGM studio. Columns in the selection criteria do not need to have been chosen in the Column Selection dialog box.



Download Wizard - Criteria Selection

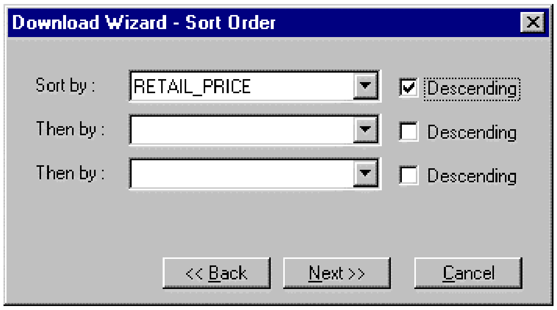
Please Select Required Criteria

□ (STUDIO = MGM □)

Figure 34.

Criteria Selection dialog box

4. The Sort Order dialog box appears. Use this to order the data that is retrieved (this is optional; you do not have to set a sort order). You can select up to 3 columns on which to order the result. For each column, the sort order may be ascending or descending. The default ordering is ascending, and is not indicated. To sort data in descending order, click the check box to the right of the column name. Click the **Next>>** button when you have made your selection.



Download Wizard - Sort Order

Sort by : RETAIL_PRICE ☑ Descending

Then by : □ Descending

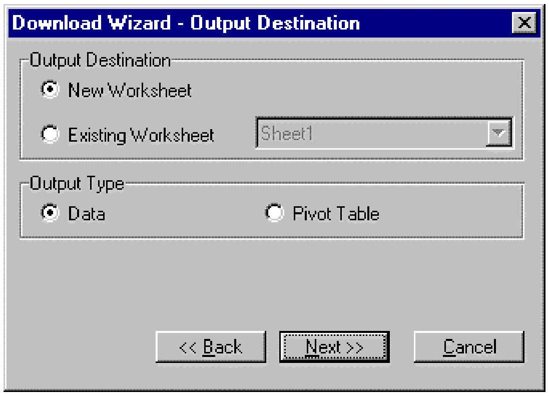
Then by : □ Descending

Figure 35.

Sort Order dialog box

5. The Output Destination dialog box appears. By default, the data will be sent to a new worksheet - the worksheet will be selected automatically. If you want to send the data to an existing worksheet, then select that option, and select the name of the worksheet from the dropdown dialog box.

   You can either output the data directly to the worksheet (this is the default option) or you can create a pivot table (see page 70). Click the **Next>>** button when you have made your selection.



Download Wizard - Output Destination

Output Destination

● New Worksheet

○ Existing Worksheet Sheet1

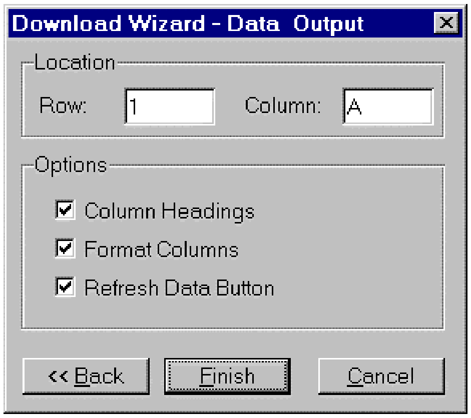
Output Type

● Data ○ Pivot Table

Figure 36.

Output Destination dialog box

6. Assuming that you selected the output type “Data” in the previous dialog box, the Data Output dialog box appears. This controls the location of the data that is downloaded. Row specifies the first row on the worksheet in which the data will be placed, and Column specifies the first column in which the data will be placed.



Download Wizard - Data Output

Location

Row: 1 Column: A

Options

☑ Column Headings

☑ Format Columns

☑ Refresh Data Button

Figure 37.

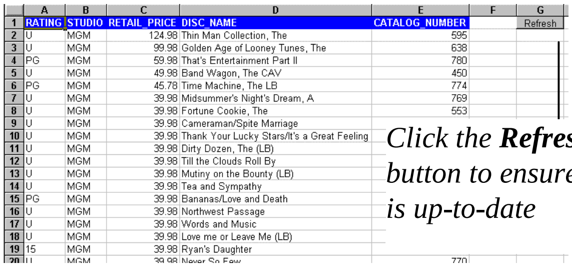
Data Output dialog box

Column Headings: specifies whether or not the names of the columns will be shown on the worksheet. By default, this option is selected.

Format Columns: sets the widths of the columns which contain downloaded data so that the data can be easily seen. Also, if the Column Headings option is selected, then this formatting option sets colour of the cells. By default, this option is selected.

Refresh Data Button: places a button (named Refresh) on the worksheet. The button is created in the second empty column after the last column of data. Clicking this button after the data has been downloaded will refresh the data (that is, it will download the data again, according to the options you selected when you first downloaded it). Thus, if the data changes frequently, you can always be sure of having the latest version, without having to work through the entire Download Wizard each time you want the latest information. By default, this option is selected.

Click the **Finish** button to download the data.



	A	B	C	D	E	F	G
1	RATING	STUDIO	RETAIL_PRICE	DISC_NAME	CATALOG_NUMBER		Refresh
2	U	MGM	124.98	Thin Man Collection, The	595		
3	U	MGM	99.98	Golden Age of Looney Tunes, The	638		
4	PG	MGM	59.98	That's Entertainment Part II	780		
5	U	MGM	49.98	Band Wagon, The CAV	450		
6	PG	MGM	45.78	Time Machine, The LB	774		
7	U	MGM	39.98	Midsummer's Night's Dream, A	769		
8	U	MGM	39.98	Fortune Cookie, The	553		
9	U	MGM	39.98	Cameraman/Spite Marriage			
10	U	MGM	39.98	Thank Your Lucky Stars/It's a Great Feeling			
11	U	MGM	39.98	Dirty Dozen, The (LB)			
12	U	MGM	39.98	Till the Clouds Roll By			
13	U	MGM	39.98	Mutiny on the Bounty (LB)			
14	U	MGM	39.98	Tea and Sympathy			
15	PG	MGM	39.98	Bananas/Love and Death			
16	U	MGM	39.98	Northwest Passage			
17	U	MGM	39.98	Words and Music			
18	U	MGM	39.98	Love me or Leave Me (LB)			
19	15	MGM	39.98	Ryan's Daughter			
20	U	MGM	39.98	Never So Few	770		

*Click the **Refresh** button to ensure data is up-to-date*

Figure 38.

Example output using macro

Note: the validation of the SQL that is generated from the dialog boxes in the Wizard does not occur until after the **Finish** button on the Data Output dialog box has been clicked.

Complex Criteria

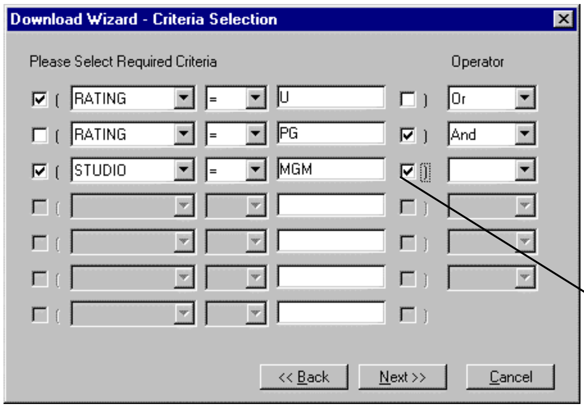
You can set complex criteria using the Criteria Selection dialog box. The columns that you use do not need to appear in the output columns. We will use the LASER table as an example.

Select the laser discs which are produced by the MGM studio and which have a rating of either U or PG.

The SQL for the corresponding selection criteria is:

WHERE (RATING = ‘U’ OR RATING = ‘PG’) AND STUDIO = ‘MGM’

To set this criteria, enter the following values in the dialog box. Pay particular attention to the start and end parentheses (brackets). A tick in the check box next to a parenthesis indicates that one will be inserted into the SQL.



Download Wizard - Criteria Selection

Please Select Required Criteria

					Operator
☑ (	RATING	=	U	□)	Or
□ (	RATING	=	PG	☑)	And
☑ (	STUDIO	=	MGM	☑)	
□ (				□)	
□ (				□)	
□ (				□)	
□ (				□)	

Figure 39.

Data Output dialog box

End

Start

Optional, but can aid clarity

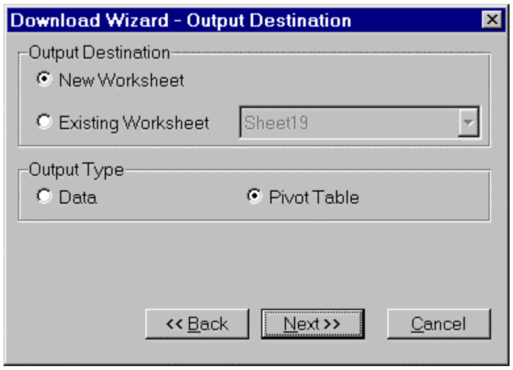
Pivot Table

Pivot tables are part of Microsoft Excel. They allow you to view data in many different ways, and you can change the way you view the data with ease. Here you will see how to initially set up a pivot table. Once you have created a pivot table, you can use the facilities that are available in Excel to modify the pivot table.

For exemplary purposes, say that for each studio we want to see the total cost of the discs in each rating category.

1. Select **Rms**, **Download Wizard**. You will be presented with the Table Download dialog box. Select the LASER table, and then work through the dialog boxes that follow to select all the columns; do not set any selection criteria and do not sort the data.

2. The Output Destination dialog box appears. Select a new worksheet as the output destination. Send the data to a pivot table by selecting the Pivot Table option. Click the **Next>>** button.


Download Wizard - Output Destination

Output Destination

● New Worksheet

○ Existing Worksheet Sheet19

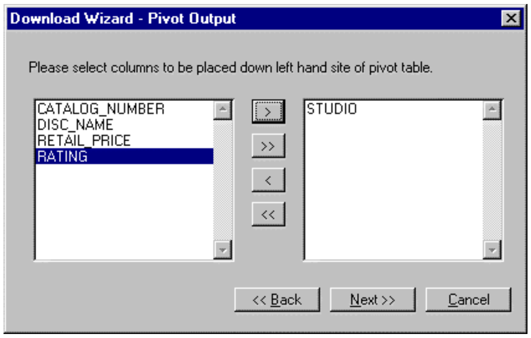
Output Type

○ Data ● Pivot Table

Figure 40.

Output Destination dialog box

3. The first of the Pivot Output dialog boxes appears. Highlight STUDIO and click the > button (you cannot select multiple columns). Then click the **Next>>** button.


Download Wizard - Pivot Output

Please select columns to be placed down left hand site of pivot table.

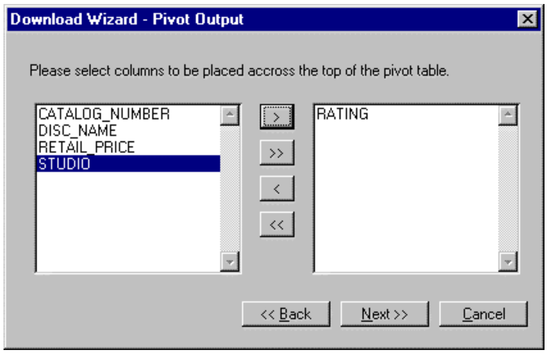
CATALOG_NUMBER
DISC_NAME
RETAIL_PRICE
RATING

STUDIO

Figure 41.

Pivot Output (1) dialog box

4. The second of the Pivot Output dialog boxes appears. Highlight RATING and click the > button (you cannot select multiple columns). Then click the **Next>>** button.



Download Wizard - Pivot Output

Please select columns to be placed accross the top of the pivot table.

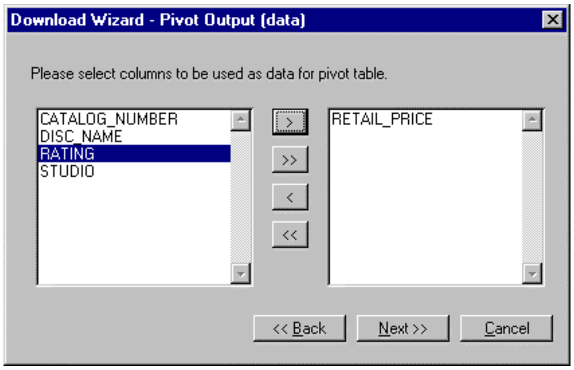
CATALOG_NUMBER
DISC_NAME
RETAIL_PRICE
STUDIO

RATING

Figure 42.

Pivot Output (2) dialog box

5. The Pivot Output (data) dialog box appears. Highlight RETAIL_PRICE and click the > button (you cannot select multiple columns). Then click the **Next>>** button.



Download Wizard - Pivot Output (data)

Please select columns to be used as data for pivot table.

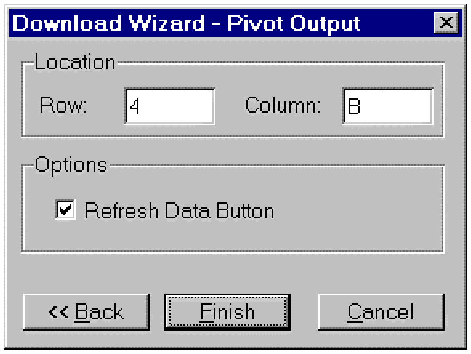
CATALOG_NUMBER
DISC_NAME
RATING
STUDIO

RETAIL_PRICE

Figure 43.

Pivot Output (2) dialog box

6. The last of the Pivot Output dialog boxes appears. Accept the default options and then click the **Finish** button.



Download Wizard - Pivot Output

Location

Row: 4   Column: B

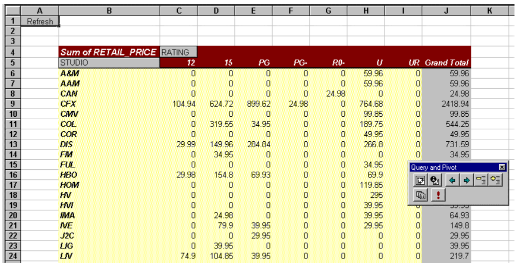
Options

☑ Refresh Data Button

Figure 44.

Pivot Output (2) dialog box

7. The pivot table is created, and you will see something like this:

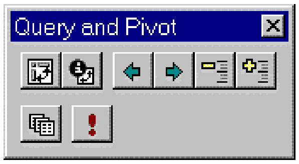


Sum of RETAIL_PRICE	RATING							
STUDIO	12	15	PG	PG-	R0-	U	UR	Grand Total
A&M	0	0	0	0	0	59.96	0	59.96
AAM	0	0	0	0	0	59.96	0	59.96
CAN	0	0	0	0	24.98	0	0	24.98
CFX	104.94	624.72	899.62	24.98	0	764.68	0	2418.94
CMV	0	0	0	0	0	99.85	0	99.85
COL	0	319.55	34.95	0	0	189.75	0	544.25
COR	0	0	0	0	0	49.95	0	49.95
DIS	29.99	149.96	284.84	0	0	266.8	0	731.59
FM	0	34.95	0	0	0	0	0	34.95
FUL	0	0	0	0	0	34.95		
HBO	29.98	154.8	69.93	0	0	69.9		
HOM	0	0	0	0	0	119.85		
HV	0	0	0	0	0	295		
HVI	0	0	0	0	0	39.95		
IMA	0	24.98	0	0	0	39.95	0	64.93
IVE	0	79.9	39.95	0	0	29.95	0	149.8
J2C	0	0	29.95	0	0	0	0	29.95
LIG	0	39.95	0	0	0	0	0	39.95
LIV	74.9	104.85	39.95	0	0	0	0	219.7

Figure 45.

Pivot Table output example

8. To modify the pivot table, you will need to use the Excel Pivot Table Wizard. Either ensure that the Query and Pivot toolbar is available (**View**, **Toolbars…** and then select the Query and Pivot option), or use the **Data**, **Pivot Table Wizard** menu options.



Pivot Table Wizard on Query and Pivot toolbar.

Upload Data

To upload data, the first row of the worksheet must contain the column names for the table you wish to upload. The data must appear on sequential lines following this. The upload stops at the first blank line. If you have previously used the Download Wizard to obtain data (or even just the column names), then you can be confident that the correct names are being used in a later upload.

After the upload has been performed, the colour of the primary key fields of the uploaded rows will change:

row colour	**indicates**
red	update or insertion failed
blue	row inserted successfully
green	row updated successfully

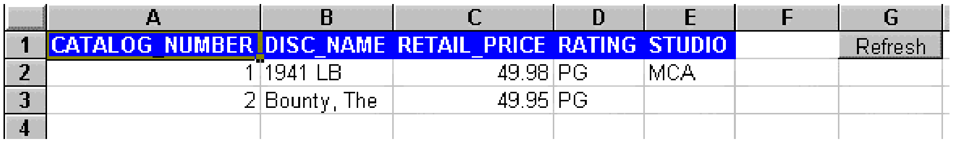
The rules for updates are as follows:

First attempt an update (green)
If the update is unsuccessful, then attempt to insert the row (blue)
If the attempted insertion is unsuccessful, then indicate failure (red)

Upload Example

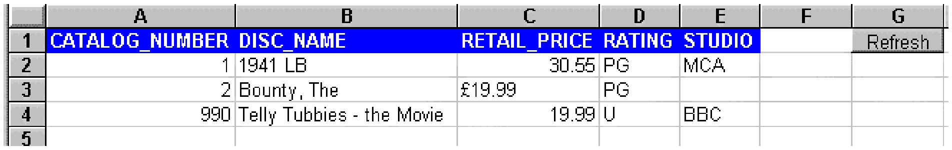
This example is for an upload of laser disc data. Say that we want to update the price of two rows and insert one new row.

1. Insert the column names into the worksheet. In this example, we did this by downloading the data for the rows we will update and requesting that the column headings be shown.



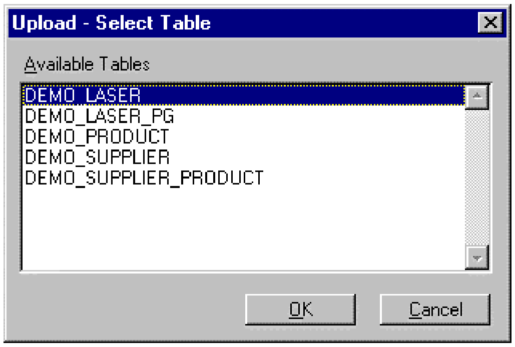
	A	B	C	D	E	F	G
1	CATALOG_NUMBER	DISC_NAME	RETAIL_PRICE	RATING	STUDIO		Refresh
2	1	1941 LB	49.98	PG	MCA		
3	2	Bounty, The	49.95	PG			
4							

2. Enter the data for the rows you want to insert or update. Ensure that there are no blank lines between the rows. In this example, we want to update the price of the first two records and insert a new record (number 990). Note there is a data type error in the second row (RETAIL_PRICE is numeric column).



	A	B	C	D	E	F	G
1	CATALOG_NUMBER	DISC_NAME	RETAIL_PRICE	RATING	STUDIO		Refresh
2	1	1941 LB	30.55	PG	MCA		
3	2	Bounty, The	£19.99	PG			
4	990	Telly Tubbies - the Movie	19.99	U	BBC		
5							

3. Start the upload by selecting **Rms**, **Upload**. The Upload - Select Table dialog box appears.



Upload - Select Table

Available Tables

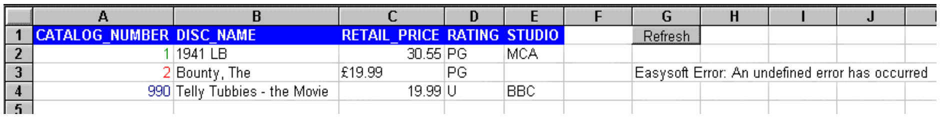
DEMO_LASER
DEMO_LASER_PG
DEMO_PRODUCT
DEMO_SUPPLIER
DEMO_SUPPLIER_PRODUCT

Figure 46.

Upload - Select Table dialog box

4. Select the table that will be updated and click the **OK** button. In this example, we used the LASER table.

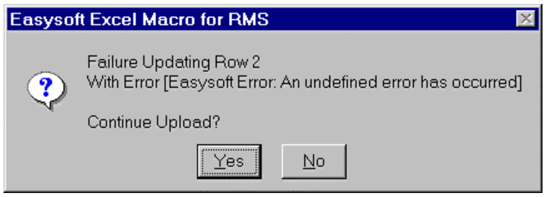
5. The data is uploaded, and the worksheet contains information on the status of the upload.



	A	B	C	D	E	F	G	H	I	J
1	CATALOG_NUMBER	DISC_NAME	RETAIL_PRICE	RATING	STUDIO		Refresh			
2	1	1941 LB	30.55	PG	MCA					
3	2	Bounty, The	£19.99	PG			Easysoft Error: An undefined error has occurred			
4	990	Telly Tubbies - the Movie	19.99	U	BBC					
5										

After the update was performed, the colours on the worksheet changed. The key field (CATALOG_NUMBER) on the first row is green, indicating successful update. The key field on the second row is red, indicating that the attempted update was not successful and an error message is presented. The key field on the last row is blue, indicating successful insertion of the new data.

Note: If the Auto Skip Failed Updates option is not selected in the Options dialog box (Figure 31), then each time that an error occurs, a message will be presented asking whether you wish to continue the update.

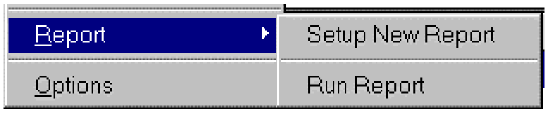


Failure Updating Row 2
With Error [Easysoft Error: An undefined error has occurred]

Continue Upload?

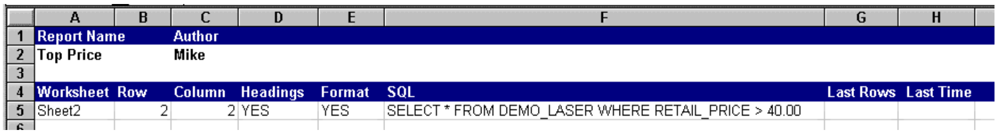
Report

This option allows you to create reports tailored to your exact needs. There are two options, **Setup New Report**, which is used to define the report, and **Run Report**.



Setup New Report

In this example, we select data from the LASER table where the price of the disc is more than 40.00.



	A	B	C	D	E	F	G	H
1	Report Name		Author					
2	Top Price		Mike					
3								
4	Worksheet	Row	Column	Headings	Format	SQL	Last Rows	Last Time
5	Sheet2	2	2	YES	YES	SELECT * FROM DEMO_LASER WHERE RETAIL_PRICE > 40.00		

1. Enter values for **Report Name** and **Author** at the top of the report. These are optional, and do not affect the output.

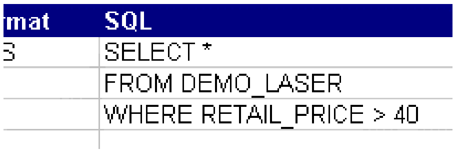
2. **Worksheet Row**. This specifies the location of the output. It must be a worksheet that currently exists, and it must not be the worksheet on which the report is defined. Remember to enter the name of the worksheet exactly as it appears on the tabs at the bottom of the screen.

3. **Row** and **Column** specify the first row and column respectively on which the data is to be returned.

4. **Headings**. If **YES** is specified, then column headings will be included in the report. This option may be left blank (so that headings are not shown).

5. **Format**. If **YES** is specified, then the width of each column is adjusted to fit the data returned. This option may be left blank, in which case the column width will not be adjusted.

6. **SQL**. This is the SQL that is sent to the server. Multiple rows of SQL are allowed, for example:

rmat	SQL
S	SELECT *
	FROM DEMO_LASER
	WHERE RETAIL_PRICE > 40

   There must not be any empty lines between the lines of SQL - the first empty line is taken to be the end of the query. Individual words cannot be split over lines.

7. Save the workbook if you want to use the report in the future. The report is now ready to run.

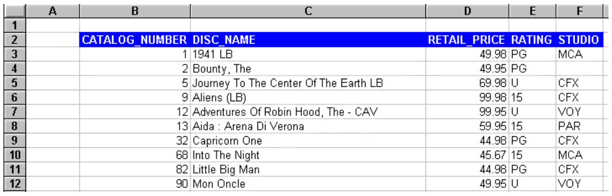
Note: **Last Rows** and **Last Time** are completed automatically when the report is run. They specify the number of rows returned and the time taken to complete the report.

Run Report

1. Select the worksheet that contains the report you wish to run.

2. Run the report (**Rms**, **Report**, **Run Report**). You should this message indicating success: `Report Completed Successfully`. Click **OK** on the dialog box.

3. Switch to the worksheet on which you asked for the data to be placed. Example output:

	A	B	C	D	E	F
1						
2		CATALOG_NUMBER	DISC_NAME	RETAIL_PRICE	RATING	STUDIO
3		1	1941 LB	49.98	PG	MCA
4		2	Bounty, The	49.95	PG	
5		5	Journey To The Center Of The Earth LB	69.98	U	CFX
6		9	Aliens (LB)	99.98	15	CFX
7		12	Adventures Of Robin Hood, The - CAV	99.95	U	VOY
8		13	Aida : Arena Di Verona	59.95	15	PAR
9		32	Capricorn One	44.98	PG	CFX
10		68	Into The Night	45.67	15	MCA
11		82	Little Big Man	44.98	PG	CFX
12		90	Mon Oncle	49.95	U	VOY

Macro Troubleshooting

- Error: Column x does not exist

  This can occur in a custom report if the following conditions apply:

  - the query contains a condition of the form:
    WHERE <column> <operator> <value>
  - <value> is a character datatype
  - the data that you entered for <value> was not enclosed within character literals, that is, single quote marks.

  For example if you wrote: WHERE RATING = U
  instead of WHERE RATING = ‘U’

  Then the following error message would occur:
  `Column U does not exist in table`.

  The reason for this is that the syntax of the first where clause is correct; it is possible to compare values from two columns. Since there is no column called “U” the query returns the error message.

- There are two (or more) Refresh buttons on the worksheet

  In the general case, this is not a problem; each different query has an associated **Refresh** button, and provided that the column headings or the result sets do not overlap, no confusion arises.

  If you want to have more than one query on the same worksheet, then ensure you plan the layout carefully. If you want to clear a worksheet of a button, then select the area to delete, and use **Edit**, **Delete** from the main menu (**Clear** removes data, but it will not remove a button).

- Number of rows retrieved are different for Excel and Easysoft

  Once the number of rows retrieved matches the number set in the Limit Rows option on the Options dialog box (Figure 31), you are asked whether or not you wish to retrieve more records. There may be an apparent conflict between the number shown on the Row Limit Exceeded dialog box and the Status bar in Excel. This is because the macro does not consider column headings in its calculation of rows retrieved.

- Error: RPC message is incomplete or corrupt

  You may see this error message if there is a timeout. You will have to re-start Excel.

CHAPTER 8

Easysoft Administrator Installation

This chapter describes the installation of the Easysoft Administrator. If you are familiar with installing Windows software you may wish to skip the details of this chapter and move to the next chapter which describes how to use the Administrator. The installation does not require much user input and should take no more than ten minutes to complete.

The Easysoft Administrator requires Windows 95 or above.

If you need to un-install the Easysoft Administrator, use the Add/Remove Programs utility in the Windows Control Panel.

Obtaining the software

The software is an executable file named EasysoftODBCAdministrator.exe. You can obtain this file using any of the following methods:

- The Easysoft web site is available 24 hours a day at http://www.easysoft.com for downloads of definitive releases and documentation. Select Download from the Easysoft ODBC-RMS Driver section of the web site. Log in. (If you have not yet done so, you need to register first. On the registration form, an asterisk (*) indicates that a field is mandatory.) From the download page, choose the Easysoft ODBC-RMS Driver client platform release that you require. The Easysoft Administrator is displayed in the Additional Downloads section of the page.
- The Easysoft FTP server is available 24 hours a day at ftp://ftp.easysoft.com, containing free patches, upgrades, documentation and beta releases of Easysoft products, as well as definitive releases. Change to the pub/download/client/ directory.
- You can order Easysoft software on CD by email, telephone or post (see Contacting Easysoft).

Installation

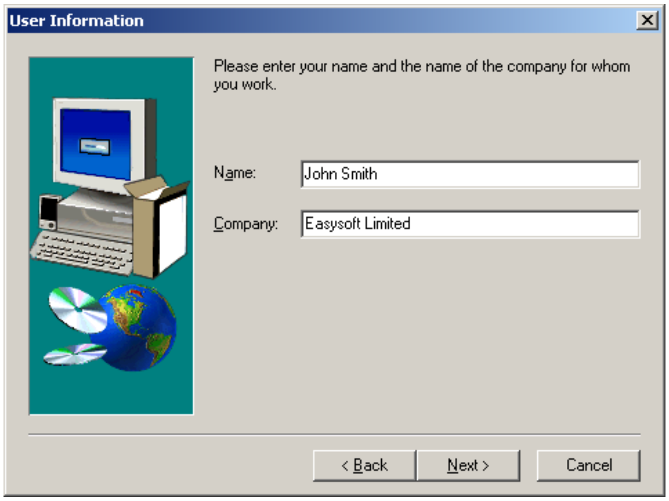
Once you have obtained the file EasysoftODBCAdministrator.exe, follow the procedure below to install the Easysoft Administrator.

1. Close any Windows applications that are running.

2. Double-click the EasysoftODBCAdministrator.exe file. The files are unpacked and after a few moments a Welcome screen is displayed. Click Next to continue with the installation.

3. The next screen displays the license agreement. Click Yes to accept the license agreement and continue with the installation. If you do not accept the license agreement, click No to exit the installation.

4. On the next screen (Figure 47), type your name and your company name. (Refer to **Table 5** in Appendix A for details of valid characters.) Click Next when you have typed in your details.



Please enter your name and the name of the company for whom you work.

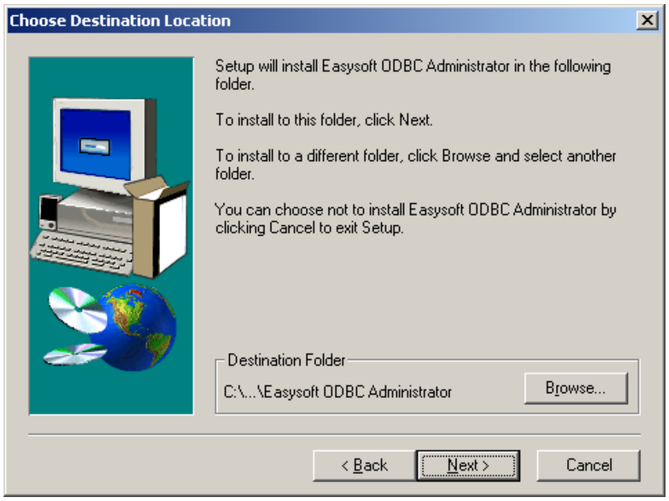
Name: John Smith

Company: Easysoft Limited

Figure 47.

Customer Information dialog box

5. The next screen (Figure 48) asks you to choose where you want to install the software. The default directory is C:\Program Files\Easysoft\Easysoft ODBC Administrator. If you want to specify an alternative directory, click Browse and choose a directory. Click Next when you have specified a suitable directory.



Setup will install Easysoft ODBC Administrator in the following folder.

To install to this folder, click Next.

To install to a different folder, click Browse and select another folder.

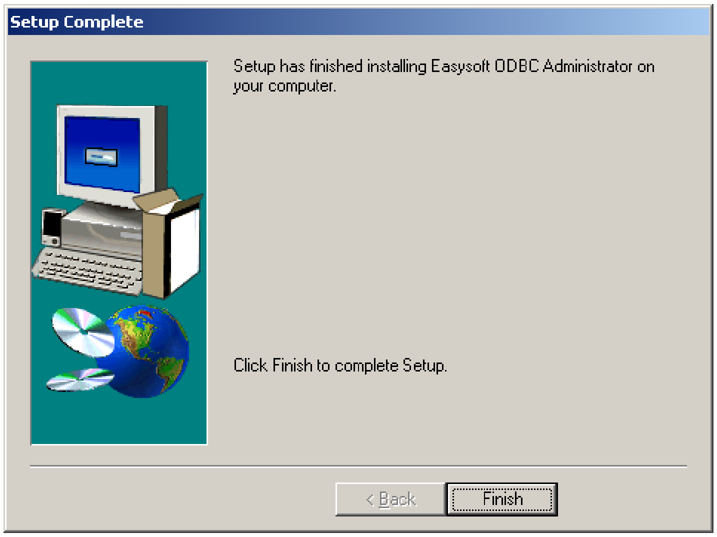
You can choose not to install Easysoft ODBC Administrator by clicking Cancel to exit Setup.

Destination Folder: C:\...\Easysoft ODBC Administrator

Figure 48.

Destination Location dialog box

The Easysoft Administrator files are now copied to the destination directory. When the installation has completed successfully, the final dialog box (Figure 49) is shown. Click **Finish** to return to Windows.



Setup has finished installing Easysoft ODBC Administrator on your computer.

Click Finish to complete Setup.

Figure 49.

Installation Complete dialog box

The following chapter is a detailed tutorial on using the Easysoft Administrator.

CHAPTER 9

Easysoft Administrator Tutorial

This chapter explains how to start the Easysoft Administrator then gives step-by-step instructions on how to use the Easysoft Administrator to define a data file which resides on the server. Once the tutorial is completed you will be able to use the defined file. The file is located in the demonstration data directory on the server and was created when the Easysoft Server Component was installed. To gain access to this file, you will use the EASYDEMO data source.

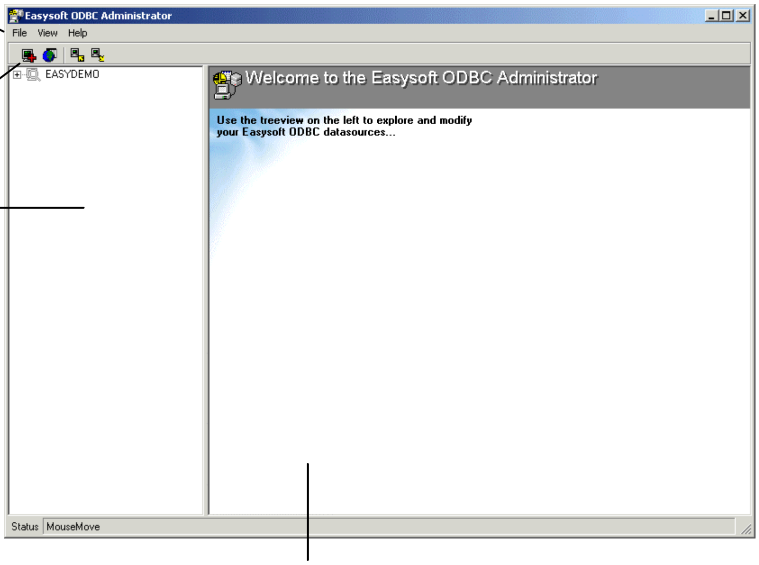
Note: The Easysoft Catalog associated with the demonstration data already has some definitions which should not be changed.

The tutorial should take about half an hour to complete. You can stop part of the way through, and resume later if you wish. You do not need to save your changes to the catalog because they are saved directly on the server.

Starting the Easysoft Administrator

To start the Easysoft Administrator, select **Start – Programs – Easysoft ODBC Administrator** then select the Easysoft Administrator program icon. (Note that your program folder might be named other than **Easysoft ODBC Administrator** if you did not accept the default during the installation procedure.)

When you start the Easysoft Administrator, its main screen (Figure 50) is displayed:


Easysoft ODBC Administrator
File
View
Help
EASYDEMO
Welcome to the Easysoft ODBC Administrator
Status
MouseMove

Use the treeview on the left to explore and modify your Easysoft ODBC datasources...

menu bar

toolbar

tree view displaying the objects in your DSNs

data pane displaying details of whatever object is selected in the tree view.

Figure 50.

Administrator main screen

The tree view lists any data sources that are set up on this machine to connect to catalogs on your server machine(s). If no data sources are listed, see Adding a Data Source on page 46 for details of how to set one up. Typically you will have at least the Easysoft demonstration data source (EASYDEMO) set up.

Stages of Defining a Server File

This section lists the stages of defining a server file for use through the Easysoft ODBC Driver. Before you can define a file on the server a catalog should be created on the server and a data source associated with that catalog should be defined using the Easysoft ODBC Setup dialog box. If the EASYDEMO data source is set up you can continue with this tutorial. If the EASYDEMO data source is not set up, see Adding a Data Source on page 46 for details of how to set up a data source.

Once you connect to the Easysoft Catalog you have a ‘live’ connection to the definitions on the server and any changes you make are saved directly on the server. For this reason, we recommend that only one person is responsible for managing the Easysoft Catalog, and that the Easysoft Administrator is installed on only one machine. Any changes can only be made by the Admin Catalog user.

1	Connect to the Catalog	Connect to the Easysoft Catalog on the server
2	File Definition	Describes the structure of a file on the server
3	Field Definitions	Describes the fields within a file on the server
4	Database Definition	If a database does not exist, then create one
5	Table Definition	SQL table name to map on to a File Definition. Assign table security if required
6	Column Definition	SQL columns to map on to fields of a file on the server
7	User Definition	Create users and assign passwords
8	Create Database Privileges	Assign user security for a Database
9	Disconnect from the Catalog	Disconnect from the Easysoft Catalog on the server

Note: the terms “Easysoft Catalog” and “System Catalog” are synonymous.

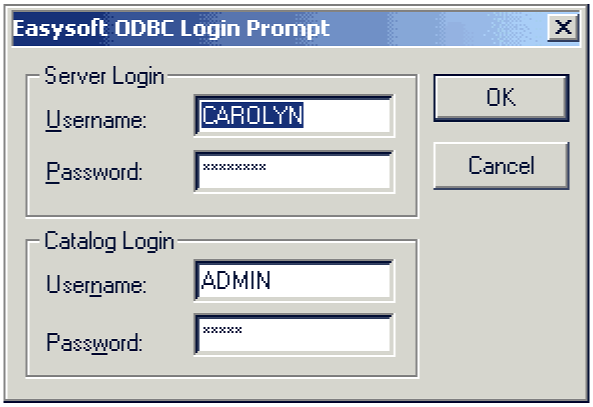
Once these steps have been performed, the data can be accessed using an ODBC-compliant application.

The following sections take you through defining a file on a server.

Step 1. Connect to the Catalog

Select the EASYDEMO data source then either right-click and select **Connect**, or click the + alongside EASYDEMO in the tree view. This automatically logs you onto the Server and connects to the Catalog using the log on information contained within the data source.

If you configured the EASYDEMO data source to ‘prompt for login’, the **Easysoft ODBC Login Prompt** dialog (Figure 51) will be displayed. (Setting the **Prompt For Login** option is explained on page 51.) If this dialog appears, ensure that the Username and Password details are complete, then click OK.



Server Login

Username: CAROLYN

Password: *********

Catalog Login

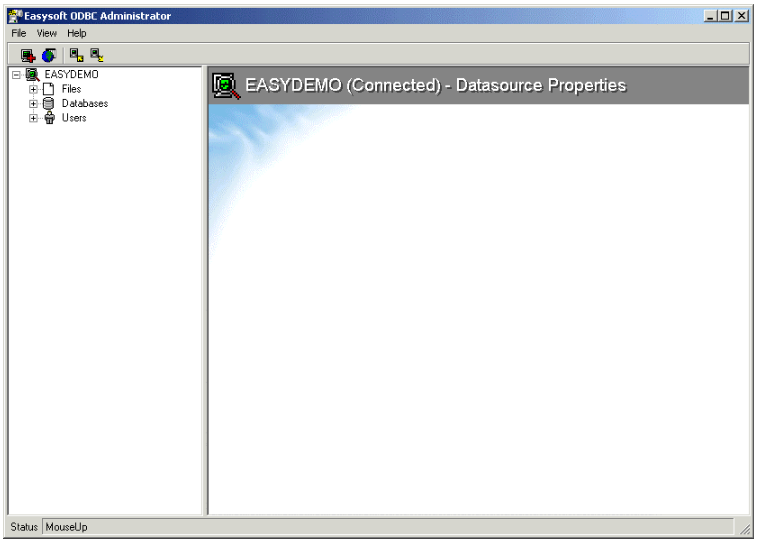
Username: ADMIN

Password: *****

Figure 51.

Easysoft ODBC Login Prompt

Once you are connected to the Catalog, the tree view expands to list Files, Databases and Users objects. (Figure 52)



EASYDEMO (Connected) - Datasource Properties

Figure 52.

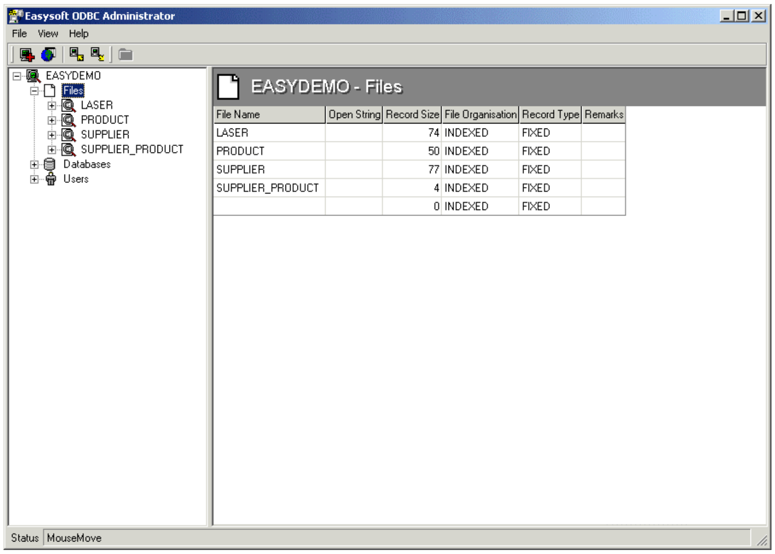
Objects listed in the tree view

Note: If you are running Microsoft Windows 95, NT or 98 (first edition) and you have MSCOMCTL.DLL version 6.1.83.41, then you will need to double-click on objects in the tree view when normally you should only need to click once.

Step 2. File Definitions

A file definition is essentially a description of the record structure of a physical file. See “Defining Tables” on page 8 for a discussion of files and file definitions.

Highlight Files in the tree view, then either double-click or click the adjacent +. A list of current file definitions is displayed - these were created automatically for demonstration and testing purposes. An overview of the file definitions is displayed in the data pane. (Figure 53)



File Name	Open String	Record Size	File Organisation	Record Type	Remarks
LASER		74	INDEXED	FIXED	
PRODUCT		50	INDEXED	FIXED	
SUPPLIER		77	INDEXED	FIXED	
SUPPLIER_PRODUCT		4	INDEXED	FIXED	
		0	INDEXED	FIXED	

Figure 53.

File definitions in EASYDEMO

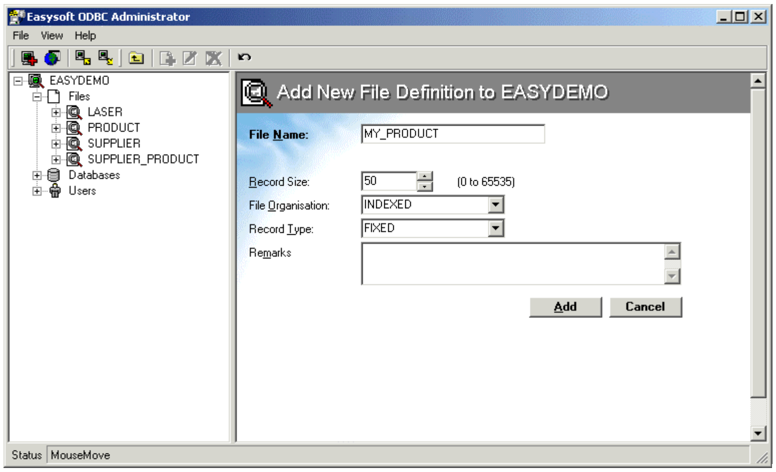
You can view the details of an individual file definition by right-clicking the file and selecting **Edit File**.

For the purposes of this tutorial, we will create a new file definition for the physical file PRODUCT.DAT. This is an indexed, fixed record file with a length of 50 bytes. We will call the definition MY_PRODUCT.

Right-click on Files and choose **Add New File**. The **Add New File Definition** screen is displayed in the data pane. Complete the screen with the following details, as shown in Figure 54.

File Name	`MY_PRODUCT`
Record Size	`50`

File Organisation	`INDEXED`
Record Type	`FIXED`



Add New File Definition to EASYDEMO

File Name: MY_PRODUCT

Record Size: 50 (0 to 65535)

File Organisation: INDEXED

Record Type: FIXED

Remarks

Figure 54.

Add New File Definition screen

When your screen matches the settings shown above, click **Add**. The tree view refreshes to add the new file definition to its list of objects. Notice that a Fields icon is indented below the MY_PRODUCT file definition.

Step 3. Field Definitions

Once you have created a new file definition, you can define the individual fields within that file. Right-click the Fields icon below the MY_PRODUCT file definition, then select **Add New Field**. The **Add New Field** screen is displayed in the data pane.

The physical file PRODUCT.DAT contains three fields:

PRODUCT	WORD
PRODUCT_NAME	STRING (Length = 40)
PRICE	DOUBLE

Create the first field by filling in the following details on the **Add New Field** screen. (The completed screen is illustrated in Figure 55.)

1. Type **`PRODUCT`** in the Field Name box.

   Field Number determines the field order within the file definition.

2 Select **WORD** from the Data Type drop-down list box. Typing the first letter(s) causes the nearest matches to be shown in the drop-down list box.

Offset is the starting position of the field with respect to the start of the file. It is calculated automatically using the values in the Length and Precision fields.

Length and Precision are defaults which depend on the data type; if they cannot be changed they are greyed out.

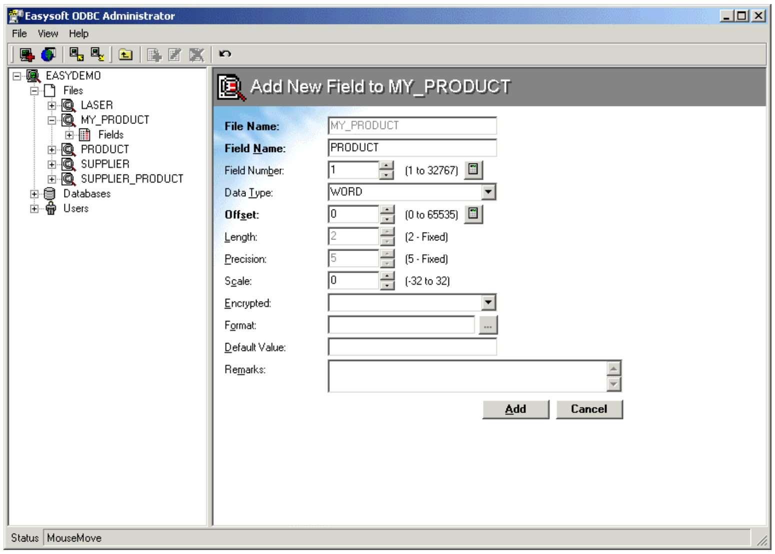
Scale refers to the maximum number of places to the right of the decimal point. This is only valid for numeric fields.

Encrypted is not relevant.

Format lets you specify the format of a date or time field stored on the Server.

Default Value lets you specify a default value which is written to the field when a record is inserted.

3 Once all the field details are correct click the **Add** button.



Add New Field to MY_PRODUCT

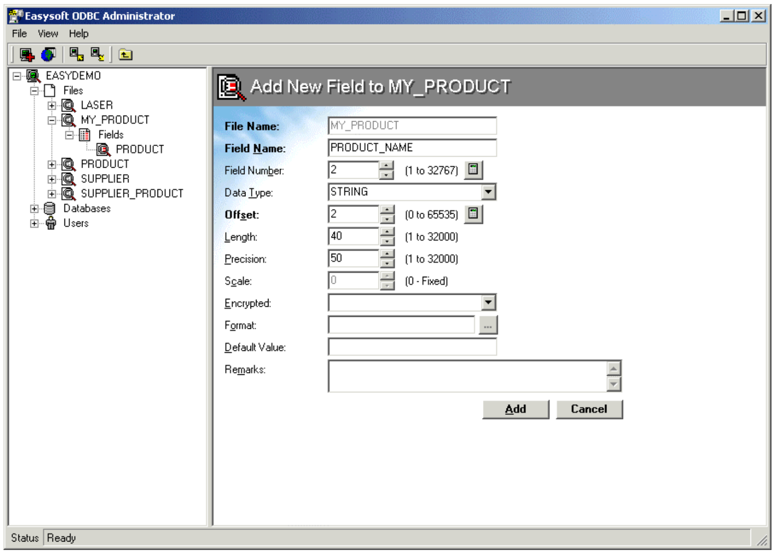
File Name: MY_PRODUCT
Field Name: PRODUCT
Field Number: 1 (1 to 32767)
Data Type: WORD
Offset: 0 (0 to 65535)
Length: 2 (2 - Fixed)
Precision: 5 (5 - Fixed)
Scale: 0 (-32 to 32)
Encrypted:
Format:
Default Value:
Remarks:

Figure 55.

Add New Field screen

The tree view refreshes to add the new field to the list of objects, and the **Add New Field** screen is cleared so that you can add another field to the MY_PRODUCT file definition.

Create the PRODUCT_NAME and PRICE fields in turn by completing the **Add New Field** screen as illustrated below in Figure 56 and Figure 57 respectively.

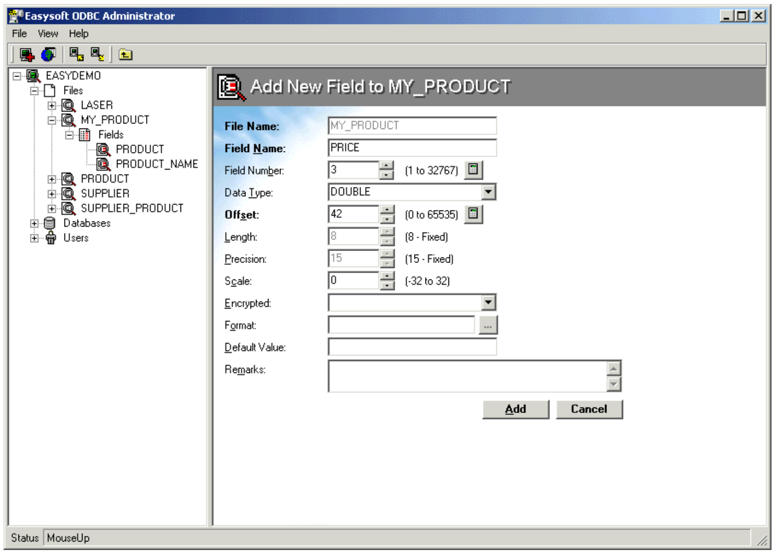


Add New Field to MY_PRODUCT

File Name: MY_PRODUCT
Field Name: PRODUCT_NAME
Field Number: 2 (1 to 32767)
Data Type: STRING
Offset: 2 (0 to 65535)
Length: 40 (1 to 32000)
Precision: 50 (1 to 32000)
Scale: 0 (0 - Fixed)
Encrypted:
Format:
Default Value:
Remarks:

Figure 56.

Creating the PRODUCT_NAME field



Add New Field to MY_PRODUCT

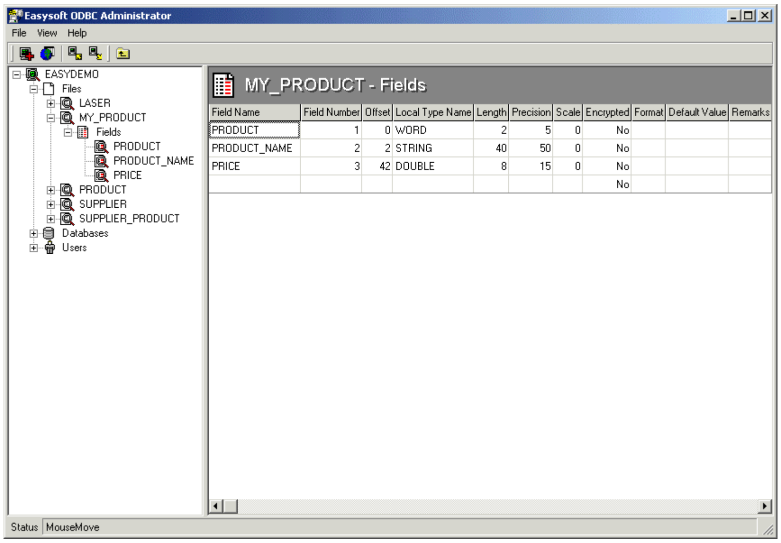
File Name: MY_PRODUCT
Field Name: PRICE
Field Number: 3 (1 to 32767)
Data Type: DOUBLE
Offset: 42 (0 to 65535)
Length: 8 (8 - Fixed)
Precision: 15 (15 - Fixed)
Scale: 0 (-32 to 32)
Encrypted:
Format:
Default Value:
Remarks:

Figure 57.

Creating the PRICE field

If you notice a mistake in any of the field details, you can correct it by right-clicking the field in the tree view and selecting **Edit Field**. The data pane then displays the **Edit Field Properties** screen where you can make any corrections then click **Update**.

When you have created all the fields for the MY_PRODUCT file definition, the Easysoft Administrator screen will look something like Figure 58:



MY_PRODUCT - Fields

Field Name	Field Number	Offset	Local Type Name	Length	Precision	Scale	Encrypted	Format	Default Value	Remarks
PRODUCT	1	0	WORD	2	5	0	No			
PRODUCT_NAME	2	2	STRING	40	50	0	No			
PRICE	3	42	DOUBLE	8	15	0	No			
							No			

Figure 58.

The field definitions for MY_PRODUCT

Step 4. Database Definitions

A database is a collection of tables. You must define a database prior to defining tables. Click the + adjacent to Databases in the tree view and notice that the DEMO database is already set up. For the purposes of this tutorial you will define a new database.

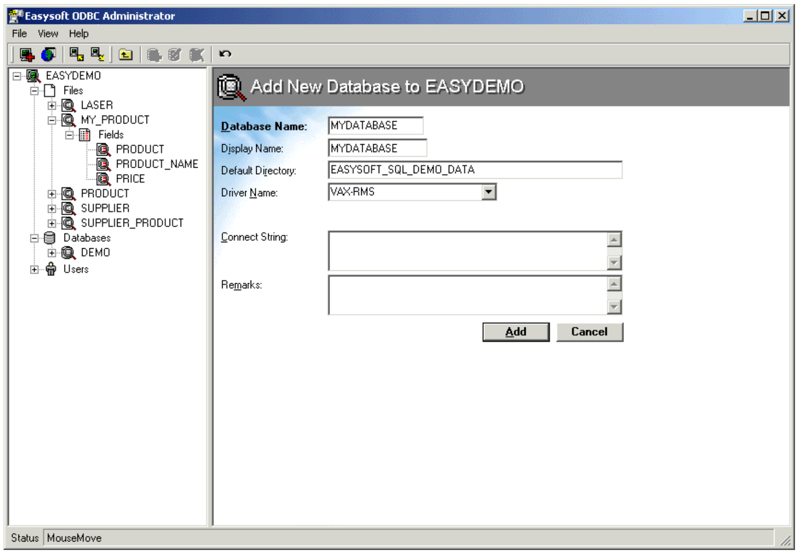
Right-click on Databases in the tree view, then select **Add New Database**. The data pane refreshes to display the **Add New Database** screen.

Enter the following details, then click **Add**. (The completed screen is illustrated in Figure 59.)

Database Name	`MYDATABASE`
Display Name	`MYDATABASE`
Default Directory	`EASYSOFT_SQL_DEMO_DATA`
Driver Name	`VAX-RMS`

Connect String is not used in this tutorial.

EASYSOFT_SQL_DEMO_DATA references the directory for the demonstration data and is created during the installation of the Easysoft Server Component.



Add New Database to EASYDEMO

Database Name: MYDATABASE
Display Name: MYDATABASE
Default Directory: EASYSOFT_SQL_DEMO_DATA
Driver Name: VAX-RMS
Connect String:
Remarks:

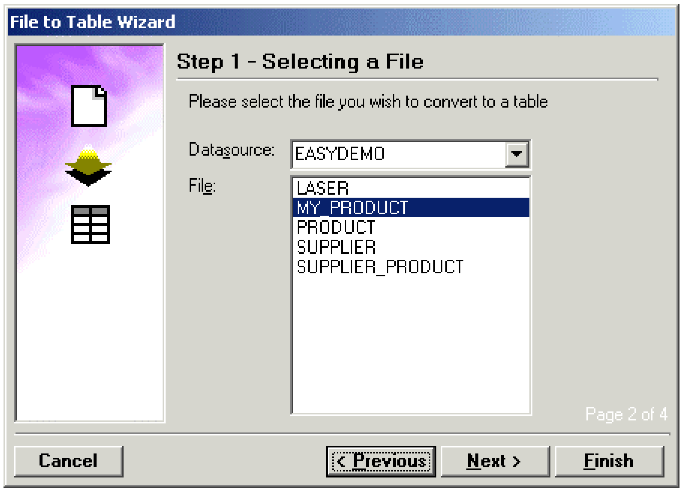
Figure 59.

Creating a new database

Step 5. Table Definitions

The easiest way to create a Table Definition from an existing File Definition is to use the **File to Table wizard**. Alternatively you can right-click on Tables for the appropriate database then select **Add New Table** and manually define the columns. In this section you will use the **File to Table** wizard.

Right-click the MY_PRODUCT file definition and select **File to Table Wizard**. The first screen of the wizard is displayed. Click Next when you have read this introductory screen. On the screen for Step 1 of the wizard (Figure 60), select the MY_PRODUCT file definition from the EASYDEMO data source, then click Next.



File to Table Wizard

Step 1 - Selecting a File

Please select the file you wish to convert to a table

Datasource: EASYDEMO

File:

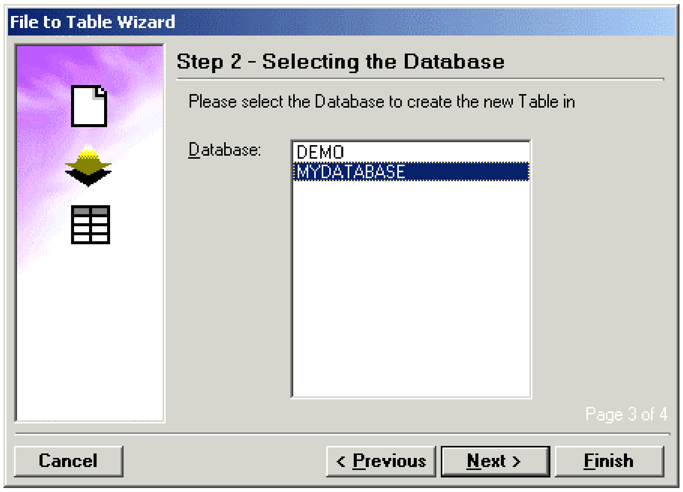
- LASER
- MY_PRODUCT
- PRODUCT
- SUPPLIER
- SUPPLIER_PRODUCT

Page 2 of 4

Figure 60.

Step 1 of the File to Table Wizard

On Step 2 of the wizard, select MYDATABASE (Figure 61). The new table will be created in this database. Click Next to continue.



File to Table Wizard

Step 2 - Selecting the Database

Please select the Database to create the new Table in

Database:

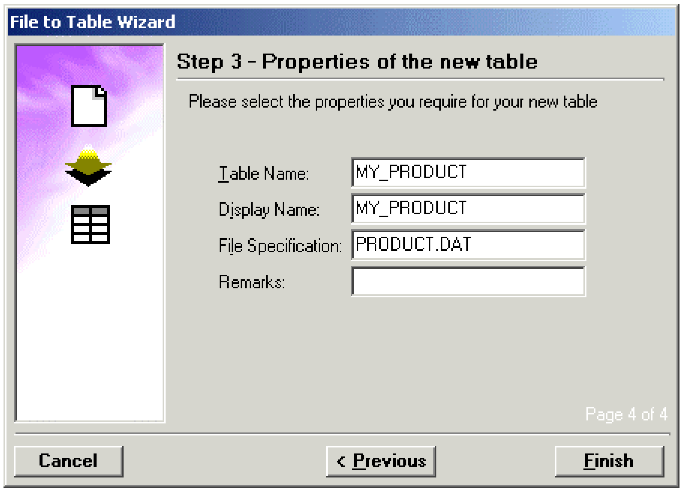
- DEMO
- MYDATABASE

Page 3 of 4

Figure 61.

Step 2 of the File to Table Wizard

On Step 3 of the wizard, specify the table properties as shown in Figure 62, then click **Finish** to create the table.



Step 3 - Properties of the new table

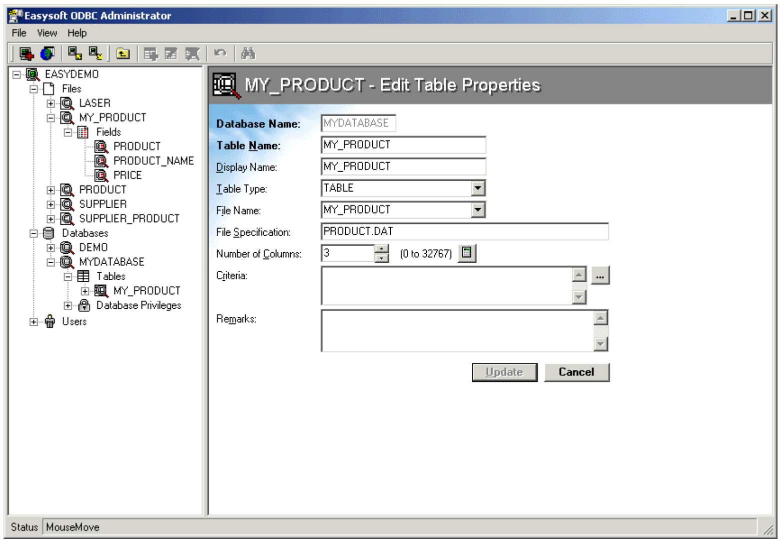
Please select the properties you require for your new table

Table Name:	MY_PRODUCT
Display Name:	MY_PRODUCT
File Specification:	PRODUCT.DAT
Remarks:	

Figure 62.

Step 3 of the File to Table Wizard

After clicking the **Finish** button, a message box is displayed informing you that the table has been created. **OK** this message box. The Administrator refreshes to display the tables in MYDATABASE. Click on the MY_PRODUCT table in the tree view to view the table's properties as illustrated in Figure 63.



MY_PRODUCT - Edit Table Properties

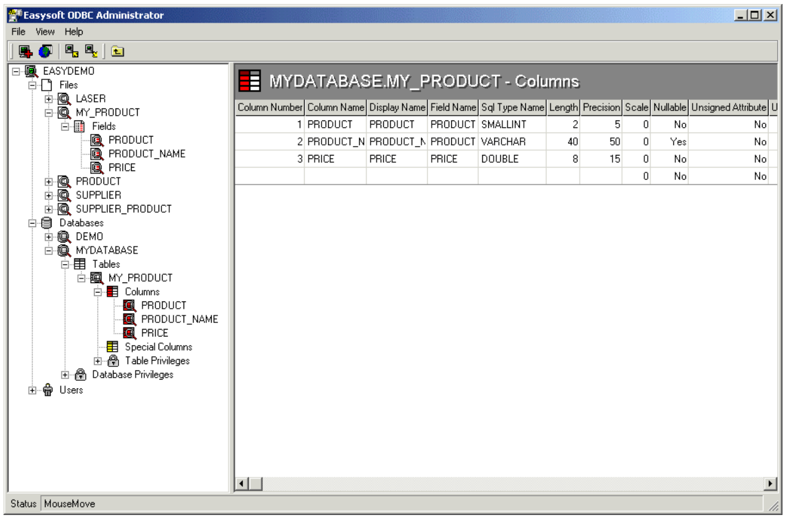
Database Name:	MYDATABASE
Table Name:	MY_PRODUCT
Display Name:	MY_PRODUCT
Table Type:	TABLE
File Name:	MY_PRODUCT
File Specification:	PRODUCT.DAT
Number of Columns:	3 (0 to 32767)
Criteria:	
Remarks:	

Figure 63.

Table Properties screen

Step 6. Column Definitions

To view the column definitions in the MY_PRODUCT table in MYDATABASE, expand the MY_PRODUCT table by clicking its +, then select the Columns object. The columns in this table are listed in the data pane (Figure 64).



MYDATABASE.MY_PRODUCT - Columns

Column Number	Column Name	Display Name	Field Name	Sql Type Name	Length	Precision	Scale	Nullable	Unsigned Attribute	U
1	PRODUCT	PRODUCT	PRODUCT	SMALLINT	2	5	0	No	No	
2	PRODUCT_N	PRODUCT_N	PRODUCT	VARCHAR	40	50	0	Yes	No	
3	PRICE	PRICE	PRICE	DOUBLE	8	15	0	No	No	
							0	No	No	

Figure 64.

Columns defined in the MY_PRODUCT table

To view an individual column definition, expand the Columns object by clicking the adjacent +, then select the column whose details you want to view. The column definition is displayed in the data pane. If you choose to create a table using the **Add New Table** command rather than the **File to Table Wizard**, once the new table has been defined you must define its columns using the **Add New Column** screen. For full details of defining columns and special columns, see *Easysoft Administrator Reference* on page 99.

The definition of the **`PRODUCT.DAT`** file is now complete, all that remains to do is to create and assign privileges to a user to access the new database.

Step 7. User Definitions

To give a user access to a database, you need to define a user, then specify that user’s privileges.

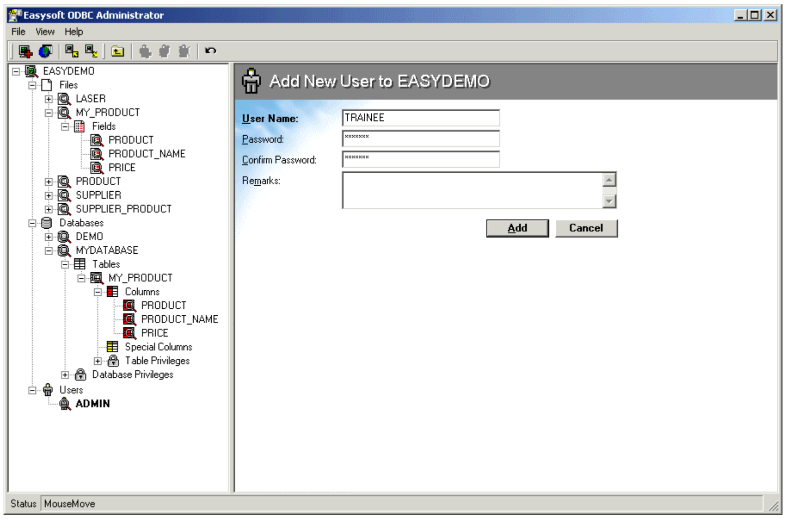
Expand the Users object by clicking its + in the tree view. The current list of users is displayed; this list contains only ADMIN if no other users have been entered.

Right-click on Users then select **Add New User**. The **Add New User** screen is displayed in the data pane. Enter the following details:

User Name	**`TRAINEE`**
Password	**`TRAINEE`**
Confirm Password	**`TRAINEE`**

The user may be required to specify their username and password when logging onto the Catalog.

The completed screen is illustrated in Figure 65.



User Name: TRAINEE
Password: *******
Confirm Password: *******
Remarks:

Figure 65.

Add New User screen

When you have entered these user details, click **Add**. The tree view refreshes to include TRAINEE in the list of users.

Step 8. Create Database Privileges

A user record alone is not enough to be able to access tables within the Easysoft SQL Engine. A user must be granted permissions to a particular database. Easysoft recommend that the ADMIN user deals only with controlling the catalog and does not have access to the server data; therefore, the ADMIN user would not normally have any database privileges. (Note that the EASYDEMO data source is a special case, and the ADMIN user does have full privileges on the demonstration data.)

To set up privileges for the TRAINEE user you have created:

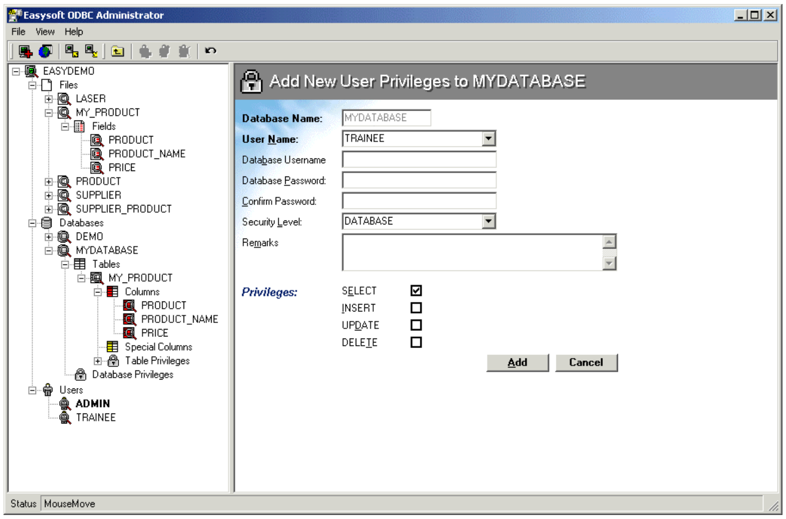
1. Expand MYDATABASE if it is not already expanded, then right-click Database Privileges and select **Add New User Privileges**. The **Add New User Privileges** screen is displayed in the data pane.

2. Select TRAINEE from the User Name drop-down list box.

3. Select DATABASE from the Security Level drop-down list box. (The other security level option allows you to set privileges per table. See the following chapter for details of how to do this.)

4. Click the SELECT box to allow this user to select data in MYDATABASE.

The completed screen is illustrated in Figure 66.



Add New User Privileges to MYDATABASE

Database Name: MYDATABASE
User Name: TRAINEE
Database Username
Database Password:
Confirm Password:
Security Level: DATABASE
Remarks

Privileges:
SELECT ☑
INSERT □
UPDATE □
DELETE □

Figure 66.

Add New User Privileges screen

Click **Add** to create these user privileges. A message appears, informing you that the password will be set to blank unless you specify a new one. Click No to proceed without specifying a password for the TRAINEE user to connect to MYDATABASE.

Step 9. Disconnect from the Catalog

To disconnect from the catalog and finish your session on the Server, right-click the data source in the tree view, then select **Disconnect**. You will need to reconnect if you need to make further changes such as define more files on the Server or set up additional users.

By connecting to the EASYDEMO data source you can now use an ODBC-compliant application such as Microsoft Excel to read and write to the **`PRODUCT.DAT`** file which resides on the server.

Note: Before being able to use the EASYDEMO data source you must change it so that it knows about the new user defined here (TRAINEE). Do one of the following using the Easysoft ODBC Setup dialog box (Figure 16):

- Make the Catalog Login Username and Catalog Login Passwords blank. When you connect to a data source you will be prompted for these by the Easysoft ODBC login prompt (Figure 51).

- Set the Catalog Login Username and Catalog Login Passwords to the TRAINEE user you created in Step 7.

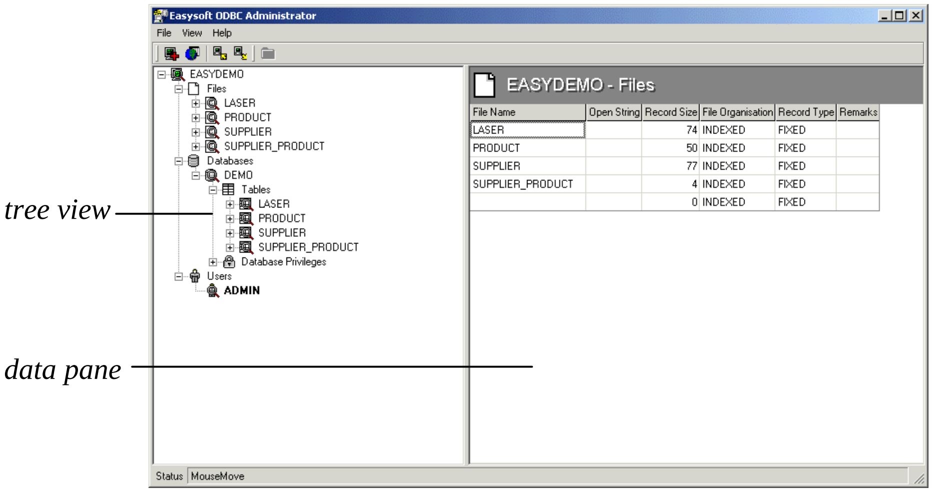
CHAPTER 10

Easysoft Administrator Reference

This chapter describes in detail the components within the Easysoft Administrator. Two areas are covered: operations which deal with definitions, and operations which can be considered utility operations, such as updating the Easysoft Catalog.

Note: The Easysoft PC Administrator is designed to be used with any of the Easysoft Server Components that might be installed on the server platforms supported by Easysoft, and therefore, some of the options may not be applicable to Easysoft ODBC for RMS. The characters that can be input into the Administrator are a subset of the characters that can be used in the catalog; if you want to use additional characters, modify the CSV catalog definitions using a text editor.

Figure 67 shows the Easysoft Administrator screen; the tree view lists the objects in each data source, and the data pane displays details of whatever is selected in the tree view. This illustration shows the files in the EASYDEMO data source.



File Name	Open String	Record Size	File Organisation	Record Type	Remarks
LASER		74	INDEXED	FIXED	
PRODUCT		50	INDEXED	FIXED	
SUPPLIER		77	INDEXED	FIXED	
SUPPLIER_PRODUCT		4	INDEXED	FIXED	
		0	INDEXED	FIXED	

Figure 67.

Easysoft Administrator

To exit the Easysoft Administrator, select **Exit** from the **File** menu.

Objects

Three types of objects are displayed in the tree view:

Files Files define the physical file and field layouts of the files stored on the server. *All* files defined in a single Easysoft Catalog are shown.

Databases Databases hold details of the SQL tables and SQL columns that map onto the information specified in the Files. *All* databases defined in a single Easysoft Catalog are shown.

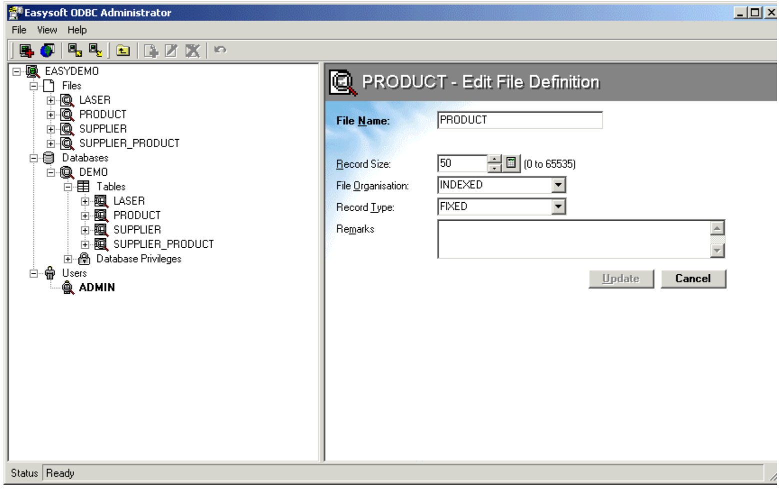
Users Users hold username and password information. A user can be assigned privileges to a whole database or to individual tables. *All* users defined in a single Easysoft Catalog are shown.

Defining Remote Files

This section explains how to use the Easysoft Administrator to define data files which reside on the server. A complete definition of a remote file comprises of file definition and field definitions. (If the organisation is indexed, the system automatically defines indexes.)

File Definition

To define a new file on the server, right-click on Files in the tree view, then select **Add New File**. The data pane refreshes to display the **Add New File Definition** screen. You can edit an existing file definition by right-clicking on it in the tree view then selecting **Edit File**. The data pane then displays the **Edit File Definition** screen (Figure 68). The options on the **File Definition** screen are basically the same whether you are creating a new file definition or editing an existing file definition.



PRODUCT - Edit File Definition

File Name: PRODUCT

Record Size: 50 (0 to 65535)

File Organisation: INDEXED

Record Type: FIXED

Remarks:

Figure 68.

File Definition screen

File Name is a unique name by which the selected file definition can be accessed from within the table definition screen. Here we are defining physical file structures. File names must be *unique* within an Easysoft Catalog. Valid characters you can input here are: a to z, A to Z, underscore (_), hyphen (-) .

Record Size defines the size of the records stored within the file. Valid record size is from 1 to 65,535. If the record type is VARIABLE, then the record size must be set to the size of the largest record in the file.

Organisation is used to specify the type of file being defined. Select the file organisation from the drop-down list.

Record Type defines the type of records stored within the file. Select FIXED or VARIABLE from the drop-down list box.

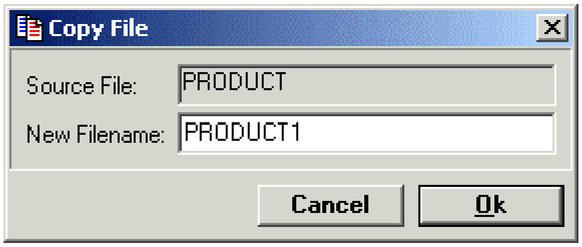
Click **Add** to create a new file definition, or **Update** to save changes to an existing file definition.

Field definitions cannot be entered until a valid file definition exists.

To remove a file definition, right-click on it in the tree view then select **Delete File**. You are asked to confirm the command before the file definition is deleted. If the file definition is being used by a table, you are warned of this before the file definition is deleted.

Note: Take care when deleting objects because they are immediately deleted from the Catalog on the Server.

You can copy a file definition by right-clicking the file in the tree view, then selecting **Copy.** The Copy File dialog box Figure 69 is displayed. Enter a name for the new file definition; it must be unique and cannot be left blank.



Source File: PRODUCT

New Filename: PRODUCT1

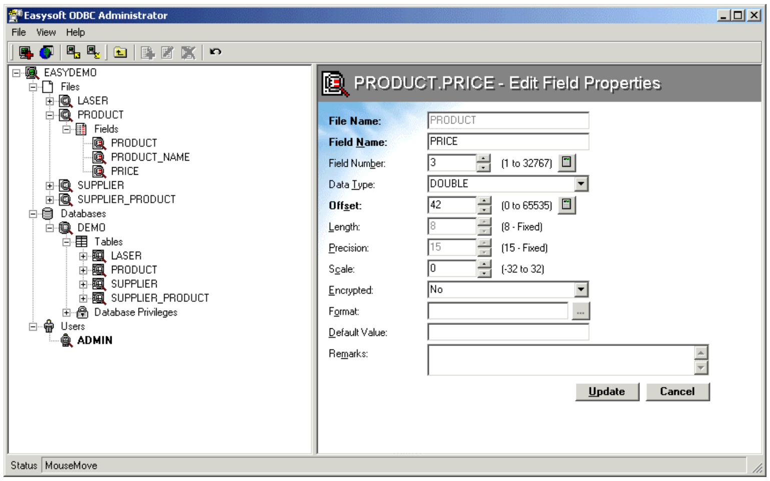
Figure 69.

Copy File dialog box

When you copy a file definition, the fields and indexes associated with the file definition are also copied.

Field Definition

A file definition defines a specific file on the server. Once you have created a file definition, you can define the fields in that file. To define a field within a file definition, expand the file definition by clicking its + in the tree view, then right-click on Fields and select **Add New Field**. The data pane refreshes to display the **Add New Field Definition** screen. You can edit an existing field definition by right-clicking the field in the tree view, then selecting **Edit Field**. The data pane then refreshes to display the **Edit Field Properties** screen. The options on the **Add New Field Definition** screen and the **Edit Field Properties** screen are basically the same, though some options may be unavailable depending on the field’s data type.



PRODUCT.PRICE - Edit Field Properties

File Name: PRODUCT

Field Name: PRICE

Field Number: 3 (1 to 32767)

Data Type: DOUBLE

Offset: 42 (0 to 65535)

Length: 8 (8 - Fixed)

Precision: 15 (15 - Fixed)

Scale: 0 (-32 to 32)

Encrypted: No

Format:

Default Value:

Remarks:

Figure 70.

Edit Field Properties screen

Field Name a name which is used to label a field within a file definition - the name must be unique within the file definition. Valid characters that you can enter here are: A to Z, underscore (_), hyphen (-).

Field Number determines the field order within the file definition.

Data Type is the storage format of the field in the file on the server. For a list of currently supported data types refer to Appendix D, “Data Types”.

Length is the maximum number of bytes of storage used by the field. For certain data types the value of this field is fixed and entry of a different value is disabled.

Offset is the position of the field within the file. It is calculated automatically using the Length and Precision values. You can override this to place fields at the same offset if multiple record types are present in the same file on the server.

Scale refers to the maximum number of digits to the right of the decimal point (numeric fields only). For approximate floating point data types, the scale is 0 by default. For Integer fields the scale can be defined by the user to give an implied decimal position. Values for scale range from -32 to + 32. (for example, an integer field of value 12345 with a scale of 2 will return 123.45 whereas an integer field of value 12345 with a scale of -2 will return 1234500).

Precision the precision of a numeric field refers to the maximum number of digits used by the data type. The precision of a non-numeric field refers to either the maximum length or defined length of the field.

Format specifies the format of a date or a time field which is stored as a string on the server. Click the ... button to specify the date or time format.

Default Value specifies a default value for a field. The default value is overwritten when data is inserted into the field.

Encrypted encrypts the field. This is used to encrypt passwords that are saved in the Catalog. When creating new field definitions you could choose to encrypt them if you intend to only access them using Easysoft ODBC; other applications will not be able to access these encrypted fields.

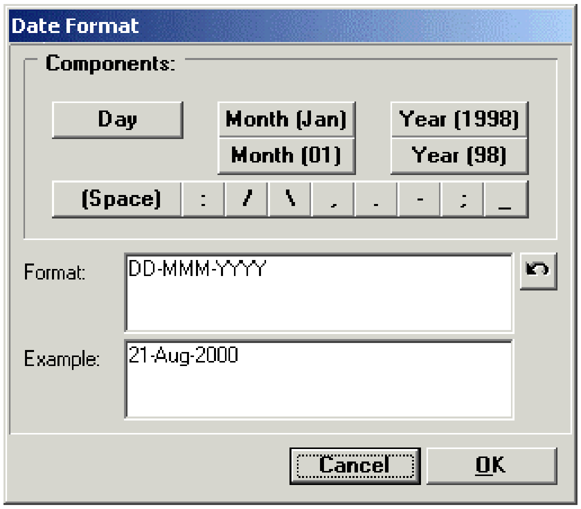
Click **Add** to create a new field definition, or **Update** to save changes to an existing field definition. If you edit a field definition that is being used by a column, when you click **Update** the column changes to reflect the updated field.

To remove a field definition, right-click on it in the tree view then select **Delete Field**. You are asked to confirm the command before the field definition is deleted. If the field definition is being used by a column, you are warned of this before the field definition is deleted.

Note: Take care when deleting objects because they are immediately deleted from the Catalog on the Server.

Date Format

When you create a field definition using the Date data type, you can specify the date format for that field. On the **Add New Field Definition** screen or the **Edit Field Properties** screen, click the (...) button to access the **Date Format** dialog box.



Format: DD-MMM-YYYY

Example: 21-Aug-2000

Figure 71.

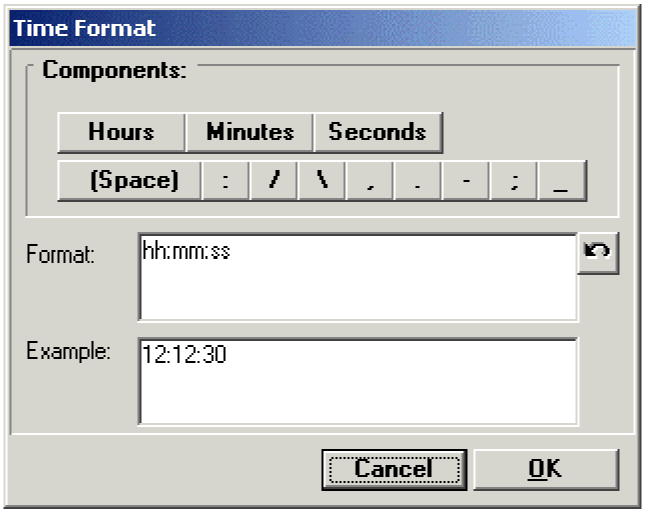
Date Format dialog box

Click buttons in the **Components** section to build up the date format you want. For example, click **Day / Month (01) / Year (98)** to display dates as 31/06/01, or click **Day - Month (Jan) - Year (1998)** to display dates as 31-Jun-2001. Click ↶ to clear the current format one component at a time, so that you can specify an alternative.

Note: Ensure that the date format you specify is valid; the software does not validate it for you.

Time Format

When you create a field definition using the Time data type, you can specify the time format for that field. On the **Add New Field Definition** screen or the **Edit Field Properties** screen, click the (...) button to access the **Time Format** dialog box.



Format: hh:mm:ss

Example: 12:12:30

Figure 72.

Time Format dialog box

Click buttons in the **Components** section to build up the time format you want. For example, click **Hours : Minutes : Seconds** to display times as 09:30:23. Click ↶ to clear the current format one component at a time, so that you can specify an alternative.

Note: Ensure that the time format you specify is valid; the software does not validate it for you.

Assigning Tables To File Definitions

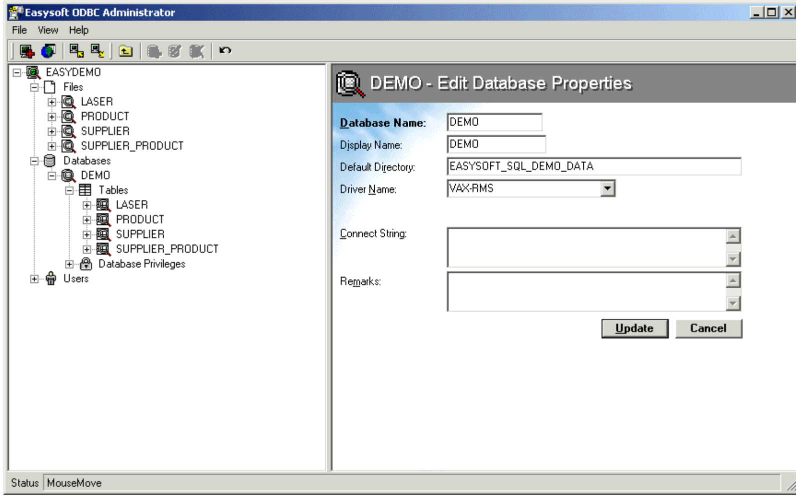
This section explains how to define tables and columns which map onto the file definitions of the files residing on the server. Tables, which must be placed within a database, consist of the following definitions: Table Definition, Table Criteria, Table Security and Column Definitions.

Database Definition

A database is a collection of tables; you create databases to store groups of tables. To create a new database, right-click the Databases object in the tree view then select **Add New Database**. The data pane refreshes to display the **Add New Database** screen. You can edit an existing database by right-clicking it in the tree view, then selecting **Edit Database**. The data pane then refreshes to display the **Edit Database**

Properties screen. The options on the **Add New Database** screen and the **Edit Database Properties** screen are basically the same.

Click **Add** to create a new database, or **Update** to save changes to an existing database.



Database Name: DEMO
Display Name: DEMO
Default Directory: EASYSOFT_SQL_DEMO_DATA
Driver Name: VAX-RMS
Connect String:
Remarks:

Figure 73.

Edit Database Properties screen

Database Name is the name of a database. Database names must be unique within each Easysoft Catalog. A database name can be a maximum of 10 characters.
Valid characters that you can enter here are: a to z, A to Z only. Typically, when viewed through an ODBC-compliant application, the database name prefixes table names; the **Use Database as Table Prefix** option in the **Easysoft ODBC Settings** dialog box (Figure 18) can be used to hide the database name.

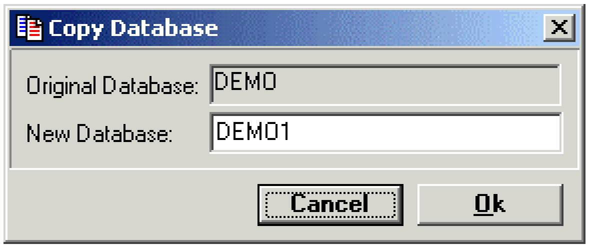
Display Name This must be the same as the **Database Name** but can be UPPERCASE, lowercase or MiXEd CaSE. This is the database name as it appears to other applications.

Default Directory This is the default location of the files stored on the server. It may be a logical or a full directory path name. In this example it is a logical which is automatically defined in the Easysoft Server Component installation. Normally, you have to define your own logicals.

Driver Name The driver name is the driver the Easysoft SQL engine uses to access the files on the server. Driver names can only be selected from the drop-down list; they cannot be typed in the text box.

Connect String This option is not applicable to Easysoft ODBC for RMS.

Database definitions can be copied by right-clicking on the database in the tree view, then selecting **Copy** button. The **Copy Database** dialog box (Figure 74) is displayed. Enter the name of the new database. It must be unique and it should not be blank.



Original Database: DEMO

New Database: DEMO1

Figure 74.

Copy Database dialog box

All the *definitions* associated with the database (database, database privileges, tables, table privileges, columns) are copied, but the data itself is *not* copied.

Table Definition

A table maps to a file definition of a physical file on the Server. A table definition consists of the definition itself, table security (if required), column details and criteria (if required).

You can define a table in two ways:

- by using the **File to Table Wizard**
- by manually defining the table and its columns

Defining a table using the File to Table Wizard

To define a table using the **File to Table Wizard**:

1. Right-click the file that you want to define a table for, then select **File to Table Wizard**.

2. Click **Next** after reading the introductory screen.

3. On the screen for Step 1 (Figure 60), select the file that you want to define a table for, then click **Next**.

4. On the screen for Step 2 (Figure 61), select the database in which you want to define the table, then click **Next**.

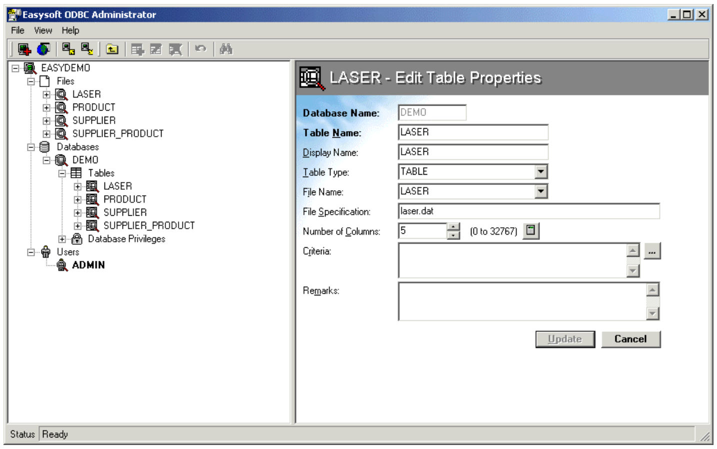
5. On the screen for Step 3 (Figure 62), enter the name of the physical file on the server, e.g. *FILENAME.DAT*. You may include the physical/logical path in the **File Specification** box. (The **Display Name** and **Remarks** settings are optional.) Click **Finish** to create the table definition.

After a few moments, a message confirms that the table has been created. **OK** this message box to continue.

Defining a table manually

To create a table definition without using the **File to Table Wizard**, right-click the Tables object within the appropriate database, then select **Add New Table**. The data pane refreshes to display the **Add New Table** screen. You can edit an existing table definition by right-clicking it in the tree view, then selecting **Edit Table**. The data pane then refreshes to display the **Edit Table Properties** screen. The options on the **Add New Table** screen and the **Edit Table Properties** screen are basically the same.

Click **Add** to create a new table definition, or **Update** to save changes to an existing table. (After creating a table definition, you will need to define the columns within that table. Column definitions are explained on page 113.)



LASER - Edit Table Properties

Database Name: DEMO

Table Name: LASER

Display Name: LASER

Table Type: TABLE

File Name: LASER

File Specification: laser.dat

Number of Columns: 5 (0 to 32767)

Criteria:

Remarks:

Figure 75.

Edit Table Properties screen

To insert a new SQL table enter the new Table Name, File Name (file definition to map onto) and the file specification.

Database Name The name of the database containing the table. The name is defined using the Database Definitions dialog box.

Table Name Table names must be unique within a database.
Valid characters that you can enter here are: a to z, A to Z, 0 to 9, underscore (_).

Display Name This must be the same as the **Table Name** but can be UPPERCASE, lowercase or MiXEd CaSE. This is the table name as it appears to other applications.

Table Type This field specifies the type of SQL table. The only valid entry at present is TABLE.

File Name The name of the File Definition to map to.

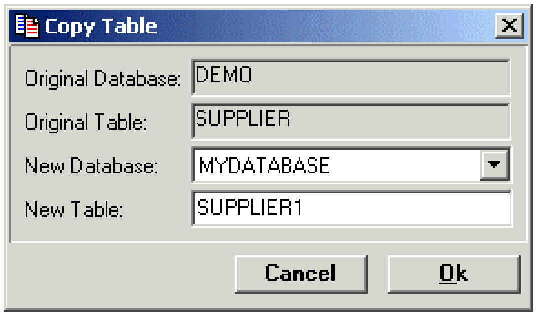
File Specification This is the file specification of the file which the File Definition maps to on the server. The file specification is placed within Table Definitions to allow a single file definition to be mapped onto multiple tables possibly in different locations. A file specification can include a path; if used, a pathname overrides the default path, that is, the **Default Directory** setting on the **Add New Database/Edit Database Properties** screen.

Number of Columns The number of columns in this table.

Criteria Criteria to eliminate records from the table. (See the following section for further information.)

Remarks Any comments you want to save with the table.

A table definition can be copied by right-clicking it and selecting **Copy**. The Copy Table dialog box (Figure 76) is displayed. Select the database in which the table definition should be stored, then enter the name of the new table definition. It must be unique within a database and it should not be blank.



Copy Table

Original Database: DEMO

Original Table: SUPPLIER

New Database: MYDATABASE

New Table: SUPPLIER1

Figure 76.

Copy Table dialog box

All the *definitions* associated with the table definition (table, table privileges, columns) are copied, but data itself is *not* copied.

Table Criteria

You can set criteria on a table definition to eliminate records from the table. For example, in a table mapped to a product file, you might want to exclude information about products above a particular price; you can do this by setting criteria on the table definition.

To set criteria on a table, display the **Add New Table/Edit Table Properties** screen (Figure 75), locate the **Criteria** box then click the (...) button. The **Table Criteria Builder** is displayed.

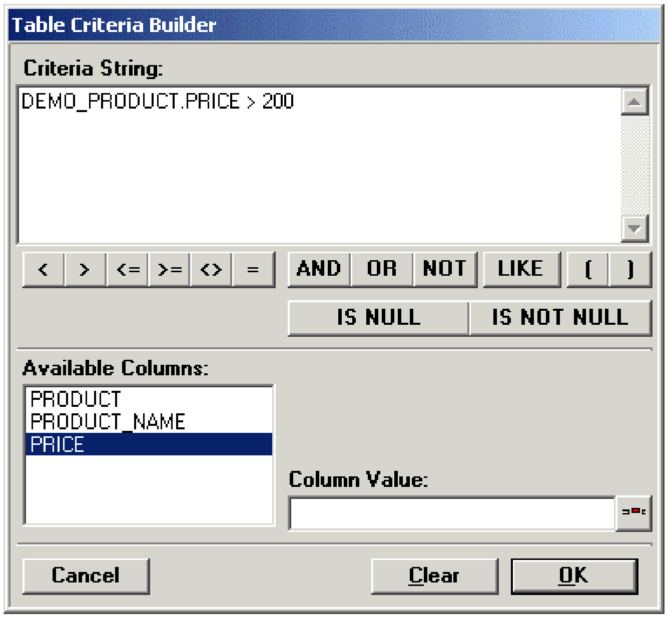


Table Criteria Builder

Criteria String:

DEMO_PRODUCT.PRICE > 200

Available Columns:

PRODUCT
PRODUCT_NAME
PRICE

Column Value:

Figure 77.

Table Criteria Builder

Criteria String shows the criteria string which can be a maximum length of 254 characters.

Available Columns is a list of columns for which criteria can be specified. Double clicking the column inserts it into the criteria string. Note that only the column name is shown in the list, although the full description of the column is added to the criteria string. The full syntax for defining a column is
<database name>_<table name>.<column name>
You can also type a value directly into the Criteria String field.

Column Value use of this entry field is optional. Enter a value in this list box then click the ( ) button. The value is inserted into the criteria string; if appropriate, quotes are added to the value. The advantage of using this box is that it automatically knows whether the value should be a character.

To enter a criteria string you can either type the string directly into the **Criteria String** box or you can use the buttons to build up the string. You could also use a combination of both these methods.

Figure 78 shows the tokens available in the string generator. Clicking one of these buttons adds the item to the criteria string. The item is added to the end of the string, even if the cursor is elsewhere.

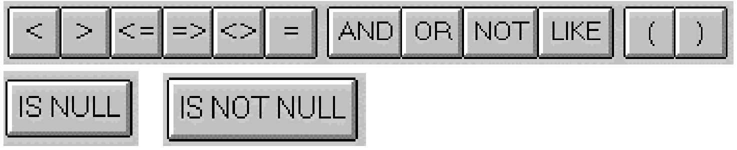


Figure 78.

Table Criteria tokens

To clear entire string click the **Clear** button.

Click **OK** to save the criteria string and exit the dialog box.

Click **Cancel** to abandon the operation and exit the dialog box.

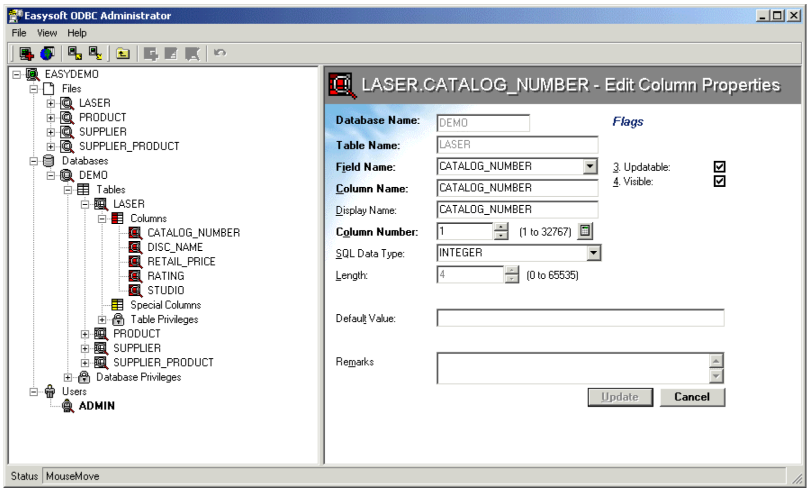
Note: There is no validation of the criteria string; please check the SQL syntax carefully. If the criteria string is invalid, when the data is accessed, the server returns a message, `Syntax error.`

Column Definitions

After creating a table definition to map to a file definition, you need to create the column definitions for that table definition. Column definitions map to the fields in a file definition.

To create columns for a table definition, expand the table definition by clicking its + in the tree view. Right-click the Columns object for that table definition, then select **Add New Column**. The data pane refreshes to display the **Add New Column** screen. You can edit an existing column definition by right-clicking it in the tree view, then selecting **Edit Column**. The data pane then refreshes to display the **Edit Column Properties** screen. The options on the **Add New Column** screen and the **Edit Column Properties** screen are basically the same.

Click **Add** to create a new column definition, or **Update** to save changes to an existing column.



LASER.CATALOG_NUMBER - Edit Column Properties

Database Name: DEMO

Table Name: LASER

Field Name: CATALOG_NUMBER

Column Name: CATALOG_NUMBER

Display Name: CATALOG_NUMBER

Column Number: 1 (1 to 32767)

SQL Data Type: INTEGER

Length: 4 (0 to 65535)

Default Value:

Remarks:

Flags

3. Updatable: ☑

4. Visible: ☑

Figure 79.

Edit Column Properties screen

Database Name the database in which this column is being defined.

Table Name the table in which this column is being defined.

Field Name the field that this column maps to.

Column Name each column must have a unique name within a table. The column may be renamed by typing the new column name and pressing the **Save** button.
Valid characters that you can enter here are: a to z, A to Z, 0 to 9, underscore (_).

Display Name the column name as it appears to other applications

Column Number the location of this column in the table.

SQL Data Type the SQL data type that this column definition maps onto. Select the value from the drop-down list box.

Length specifies the length of the SQL data type - it can only be altered if the SQL data type allows variable lengths.

The length of a column is the maximum number of bytes returned to an application when data is transferred to it. For character data, the length does not include the null termination byte. The length of a column may be different from the number of bytes required to store the data on the data source.

Default Value specifies a default value for a column field. Default values are overwritten when data is inserted into the field. Default values set on column definitions override default values set on field definitions. There is no provision to prevent default values being longer than the specified length of the data type.

Remarks Any comments you want to save with the column.

You can specify whether a column is updateable or visible by setting the following flags:

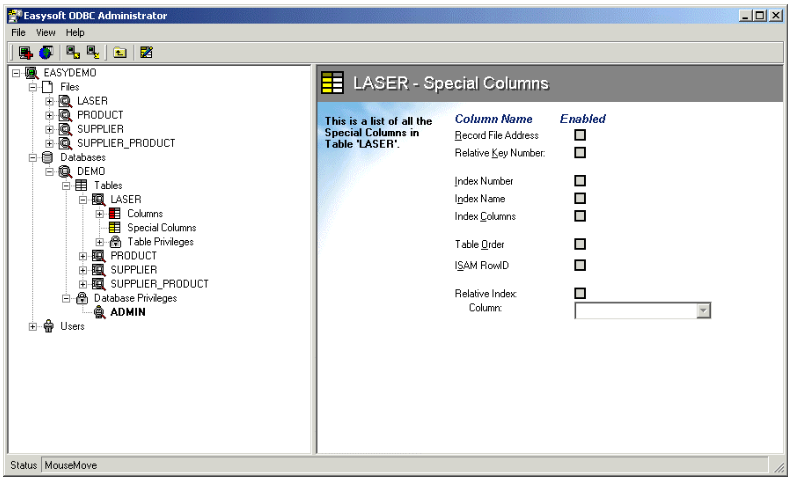
Updateable whether the column can be updated (only relevant if the logged on user has UPDATE privileges to this table or the whole database).

Visible if this box is checked, the column is visible to the user.

To delete a column definition, right-click on it in the tree view then select **Delete Column**. You are asked to confirm the command before the column definition is deleted.

Special columns

Each table can have a number of special columns. To set up special columns for a table, click the Special Columns object for the relevant table. The data pane refreshes to display the **Special Columns** screen.



LASER - Special Columns

This is a list of all the Special Columns in Table 'LASER'.

Column Name	Enabled
Record File Address	□
Relative Key Number:	□
Index Number	□
Index Name	□
Index Columns	□
Table Order	□
ISAM RowID	□
Relative Index:	□
Column:	

Figure 80.

Special Columns screen

To change the special columns you must first click the button. The boxes then become available for editing. Once you have made your changes, click the **Update** button to save them to the server.

Add a special column to the table by putting a check mark in the relevant box. Since the Easysoft Administrator is not tied to any specific server platform, not all the special columns apply in any one situation. If a special column is not valid for your server installation, its box is greyed-out on the **Special Columns** screen.

The function of each special column is described below.

Record File Address (RFA) Shows the RFA of RMS records in the “RFA” column. This option is applicable to indexed and relative files only, but not sequential files. See also the RFA data type (Appendix D).

Relative Key Number Returns a column containing the relative key in the results set. This is only valid for RMS files when the File Organisation is set to Relative. (Figure 68)

INDEX Number Shows the number of the index that is used to resolve the SQL query. The QEP scoring mechanism (see Appendix J) automatically selects the index to use. You can override this by specifying your chosen index in the selection criteria of the SQL query. How you do this is application-dependent.

INDEX Name Shows the name of the index that is used to resolve the SQL query.

INDEX Columns Shows the names of the columns that are used by the index.

TABLE_ORDER Selector For multi-table queries, this shows the order of evaluation of the tables.

ISAM RowID Returns a column containing the unique ISAM record number in the results set. This is only valid for ISAM files.

Relative Index/Column These settings can be used together to speed up access to RMS files which do not have indexes. See the following paragraph for an explanation of these settings.

This paragraph explains how to use the **Relative Index** and **Column** settings on the **Special Columns** screen.

Let us suppose that you have an ORDERS table containing order header information, and an ORDER_LINES table containing the lines in each order. The ORDER_LINES table is stored in an RMS file with no indexes. If a customer places an order for 20 different items, that order will have 20 lines in the ORDER_LINES table and each of these lines will have the same order number (let’s assume the order number is stored in the ORDER_NO column).

When the **Relative Index** and **Column** settings are not used, your query has to read every line of the ORDER_LINES table to find all the lines with a particular order number, which could take a long time depending on the number of rows in the table.

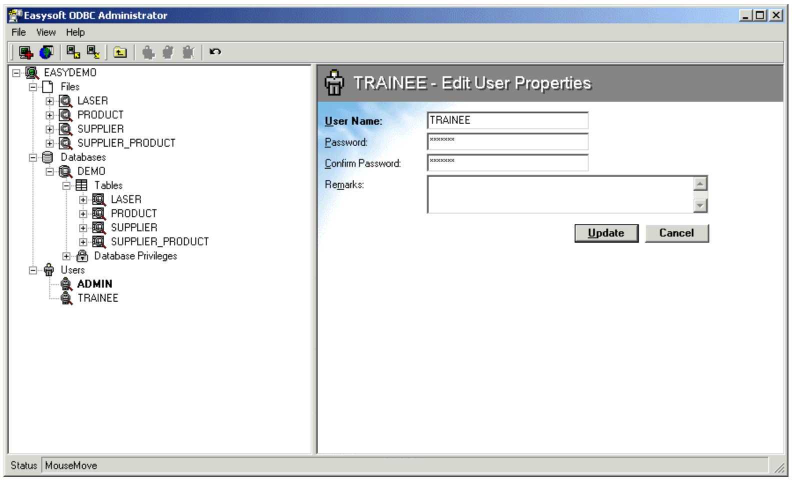
To speed up the data retrieval, you can use the **Relative Index** and **Column** settings for the ORDER table: select the **Relative Index** box so that the table uses the relative key numbers in tables, then from the **Column** drop-down list box select the ORDER_NO column in the ORDER_LINES table.

When you then run a query to find all the order lines with a particular order number, the query is able to go straight to the first row in the ORDER_LINES table with the specified order number (because relative key numbers are being used) and it then reads all the order lines until the value in the ORDER_NO column changes. So, if you query for all the order lines with one order number then as soon as the order number changes the query stops reading from the ORDER_LINES table, making data retrieval much faster.

Defining Users

You can create users then give them access to data in the Catalog. You can give a user access to a whole database, or to individual tables in a database.

To create a user, right-click the Users object in the tree view then select Add New User. The data pane refreshes to display the Add New User screen. You can edit an existing user by right-clicking on the user in the tree view, then selecting **Edit User**. The data pane then refreshes to display the **Edit User Properties** screen. The options on the **Add New User** screen and the **Edit User Properties** screen are basically the same.



User Name: TRAINEE
Password: *******
Confirm Password: *******
Remarks:

Figure 81.

Edit User Properties screen

Enter the user name and a password for that user to log on to the Catalog. Re-enter the password in the **Confirm Password** box. See Appendix A for details of restrictions on user passwords. Click **Add** to create a new user, or **Update** to save changes to an existing user.

Creating a user does not automatically give that user access to databases and tables in the Catalog. You must set up database or table privileges to grant the user access rights.

To delete a user, right-click the user in the tree view then select **Delete User**. You will be asked to confirm the command before the user is deleted.

Note: The ADMIN user cannot be deleted.

Changing Administrator Password

For security reasons you should change the password for the Admin user immediately after installation.

To change the password for the Admin user, right-click the Admin user in the tree-view then select **Edit User**. Enter a new password in the **Password** box, then re-enter it in the **Confirm Password** box. Click **Update** to save the new password. See Appendix A for details of password restrictions.

Database and table privileges

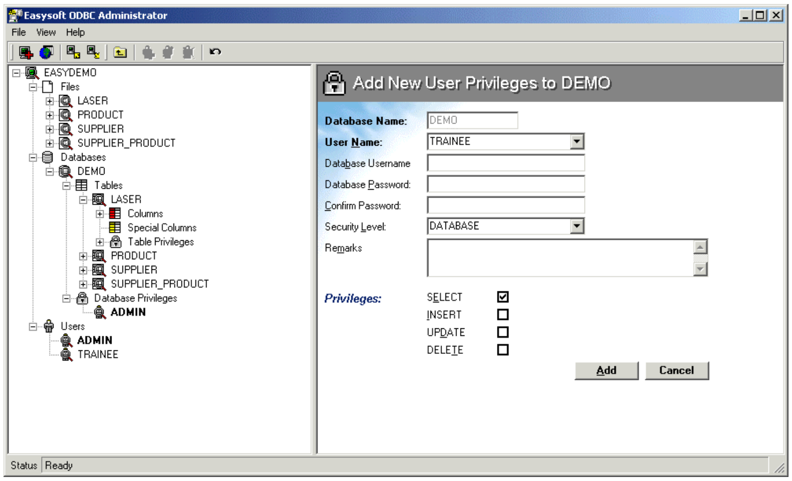
When you create a user, that user does not automatically have access to databases and tables in the Catalog; to grant a user access you must give them database or table privileges.

Easysoft recommend that the ADMIN user deals only with controlling the Catalog and does not have access to the server data. Therefore, this user is not usually given any database or table privileges.

Database Privileges

When you grant a user privileges to a database, that user can access any table within the database.

To grant a user access to a database, right-click the Database Privileges object in the tree view then select **Add New User Privileges**. The data pane refreshes to display the **Add New User Privileges** screen. You can edit a user's existing database privileges by right-clicking on the user within the Database Privileges object, then selecting **Edit User Privileges**. The data pane then refreshes to display the **Edit User Privileges** screen. The options on the **Add New User Privileges** screen and the **Edit User Privileges** screen are basically the same.



Add New User Privileges to DEMO

Database Name: DEMO

User Name: TRAINEE

Database Username

Database Password:

Confirm Password:

Security Level: DATABASE

Remarks

Privileges:

SELECT ☑

INSERT ☐

UPDATE ☐

DELETE ☐

Status Ready

Figure 82.

Add New User Privileges screen

Database Name the name of the database to which you are granting user privileges

User Name the user for whom you are granting privileges. Select a user name from the drop-down list box.

Database Username used to pass a name for a database where applicable. Not applicable to Easysoft ODBC for RMS.

Database Password used to pass a password for a database where applicable. Not applicable to Easysoft ODBC for RMS.

Confirm Password Confirms the Database Password where applicable. Not applicable to Easysoft ODBC for RMS.

Security Level determines whether the user's privileges apply at database or table level. When table level is selected, you can grant different privileges for individual tables in the database.

Remarks Any comments you want to save with these user privileges

SELECT Allows the user to read records from the specified database

INSERT Allows the user to insert records into the specified database

UPDATE Allows the user to update existing records within a database

DELETE Allows the user to delete existing records within a database

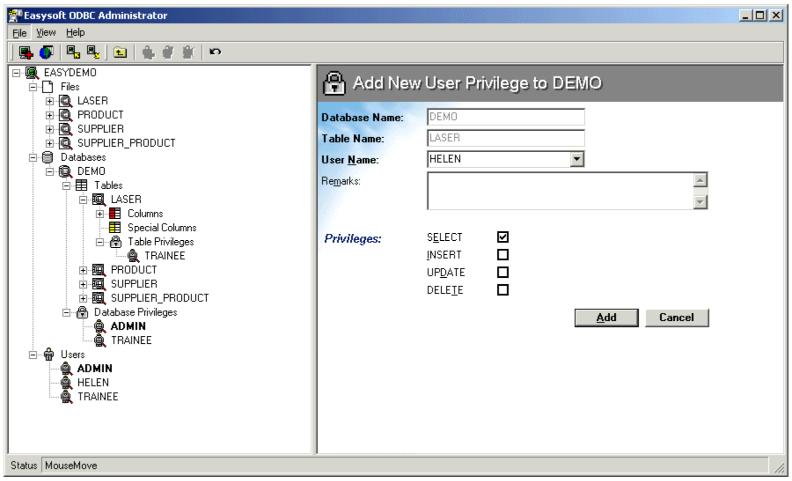
To delete a user’s database privileges, right-click the user within the relevant Database Privileges object then select **Delete User’s Privileges**. You will be asked to confirm the command before the user’s privileges are deleted.

Note: Database privileges override table privileges. For example, if a user has UPDATE privileges to a specific table but has only SELECT privileges to the database, the user will not be able to update the table.

Table Privileges

You can grant a user different privileges to individual tables within the database. For example, you might want a Personnel Manager to have full access to an EMPLOYEE table but only read access to a PRODUCT table.

To grant a user access to a specific table, expand the table by clicking its + in the tree view, then right-click on its Table Privileges object and select **Add New User Privileges**. The data pane refreshes to display the table-specific **Add New User Privileges** screen. You can edit a user’s existing table privileges by right-clicking on the user within the Table Privileges object, then selecting **Edit User Privileges**. The data pane then refreshes to display the table-specific **Edit User Privileges** screen. The options on the table-specific **Add New User Privileges** screen and the **Edit User Privileges** screen are basically the same.



Add New User Privilege to DEMO

Database Name: DEMO

Table Name: LASER

User Name: HELEN

Remarks:

Privileges:
SELECT ☑
INSERT □
UPDATE □
DELETE □

Figure 83.

Add New User Privileges screen for a specific table

Database Name the name of the database in which you are granting table privileges

Table Name the name of the table to which you are granting a user privileges

User Name the name of the user to whom you are granting privileges to the selected table. Select a user name from the drop-down list box.

Remarks Any comments you want to save with these user privileges

SELECT Allows the user to read records from the specified table

INSERT Allows the user to insert records into the specified table

UPDATE Allows the user to update existing records within the specified table

DELETE Allows the user to delete existing records within the specified table

To delete a user’s table privileges, right-click the user within the relevant Table Privileges object then select **Delete User’s Privileges**. You will be asked to confirm the command before the user’s privileges are deleted.

Note: Table privileges are overridden by database privileges. For example, if a user has UPDATE privileges to a specific table but has only SELECT privileges to the database, the user will not be able to update the table.

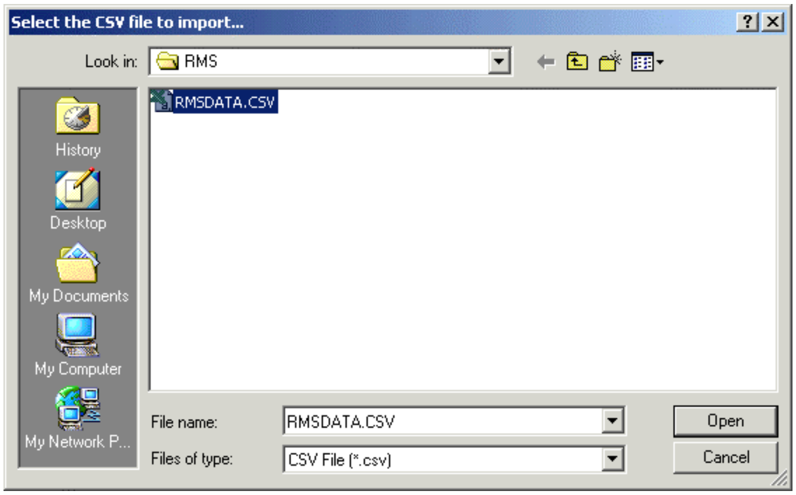
Importing and Exporting Definitions

This section explains how object definitions that can be created within the Administrator (for example, database, user, table) can be imported and exported to a text file. This enables easy transfer of definitions between systems as well as giving developers a facility to store different versions of systems.

Currently supported import/export formats are Comma Separated Values (CSV) and Fixed Width Text. Definitions for these can be found in Appendix F, “Import Export Formats”.

Importing Definitions

The import definition facility (Figure 84) allows a user to import definitions from a text file. To import a file, click **Import Catalog** from the **File** menu or click the **Import Catalog** button ( ) in the toolbar, then click the **Import** button displayed in the data pane. A standard Windows file selector (Figure 84) is displayed for you to select the file you want to import.



Select the CSV file to import...

Look in: RMS

File name: RMSDATA.CSV

Files of type: CSV File (*.csv)

Figure 84.

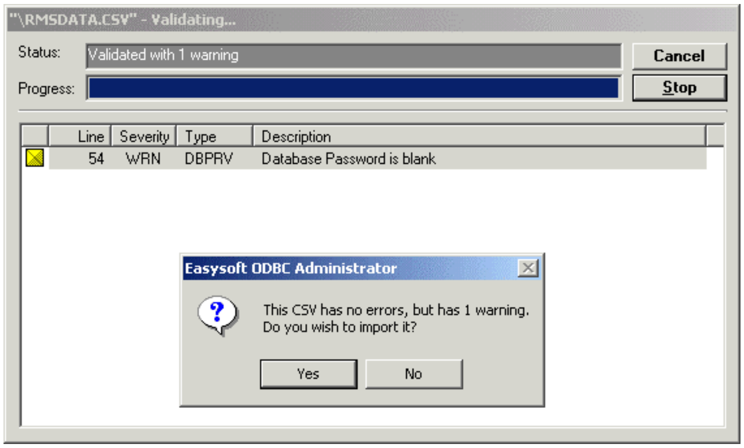
Import dialog box

Look In use this drop-down list box to locate the folder where your CSV or TXT file has been saved.

File Name type the name of the file you want to import, or click on the file.

Files of Type select the format of the file you want to import. Currently supported import formats are comma separated values (CSV) and fixed width text.

Click on **Open** to import the selected file. The file is validated before it is imported. A dialog is displayed and you must click its Start button to proceed with the validation process. If any items in the file fail the validation, errors or warnings are listed on this dialog. If any errors are reported, you must fix them before the file can be imported; if any warnings are reported, the import can proceed but a message is displayed asking if you want to continue. There will always be a warning that the database password is blank (Figure 85); this is because the password is not exported. If this is the only warning given, click **Yes** to continue with the import.



"\RMSDATA.CSV" - Validating...

Status: Validated with 1 warning

Progress:

	Line	Severity	Type	Description
	54	WRN	DBPRV	Database Password is blank

Easysoft ODBC Administrator

This CSV has no errors, but has 1 warning.
Do you wish to import it?

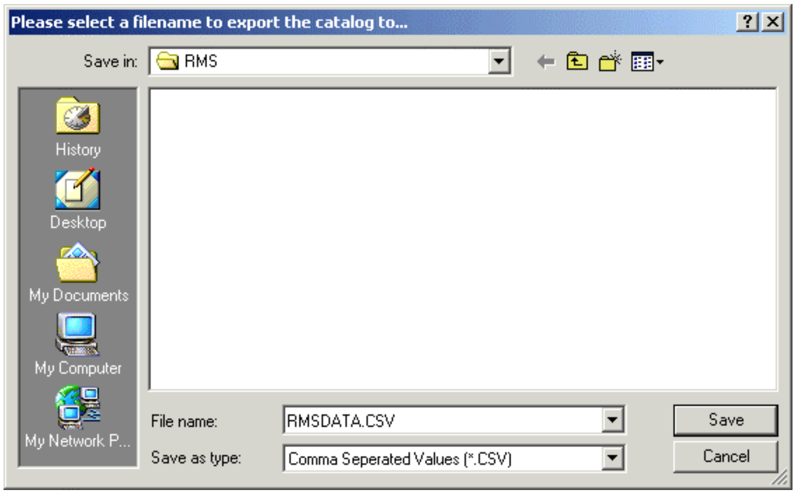
Figure 85.

Validating the import file

The import progress is displayed on the validation dialog. A message box is displayed when the import is complete. **OK** this message box to continue. You can then work with the file definitions, field definitions and other objects that have been imported.

Exporting Definitions

The export facility allows you to export all definitions stored within a catalog to comma separated or fixed width text files. To export a catalog, right-click on the appropriate data source in the tree view and select **Export Catalog** or click the **Export** button ( ) on the toolbar, then click the **Export** button displayed in the data pane. A standard Windows file selector (Figure 84) is displayed for you to specify a location and filename for the export file.



File name: RMSDATA.CSV

Save as type: Comma Seperated Values (*.CSV)

Figure 86.

Export dialog box

Save In use this drop-down list box to locate the folder where you want to save the export file.

File Name type a filename for the export file.

Files of Type select the format you want to use for the export file. Currently supported export formats are comma separated values (CSV) and fixed width text.

Click **Save** to proceed with the export. A message box is displayed when the export is complete. **OK** this message box to continue. You can then open the file in another system or import it back into another database using the **Import Catalog** command.

Clear Catalog

You can completely clear a Catalog of all its file definitions, field definitions, tables and other objects that have been defined. For example, you might want to do this before importing a CSV file from another source.

To clear a Catalog, right-click the data source in the tree view then select **Clear Catalog**. A message is displayed asking you to confirm that you want to completely clear the catalog. Click **Yes** to clear the Catalog or **No** to abandon the command.

CHAPTER 11

Converters

This chapter describes the utilities which convert native data definitions into a form suitable for the Easysoft catalog. There are four such converters:

- COBOL
- Powerhouse PDL (also known as QSHOW)
- UserBase
- IQ

COBOL Converter

The Easysoft COBOL Converter is used to convert the file definitions contained in the FILE SECTION of COBOL files to the CSV form (Comma Separated Value) required by the PC-based Easysoft Administrator and the server-based Administrator.

The Easysoft Cobol Converter conforms to the following COBOL standards:
ANSI '74
ANSI '85 (ANSI X3.23 - 1985)
X/Open
VAX COBOL 5.3
DEC COBOL 2.4

This text is based on the assumption that you know about the form and function of the FILE SECTION of COBOL files.

To use the COBOL converter, you will need write privileges to the server.

Using the COBOL Converter

This section describes the options available; examples for each option are presented after this.

The entire process of conversion and using the generated converted file is:

1. Convert the COBOL file (all file and record definitions in the FD section are converted)

2. Either
   a) Import the CSV file directly into the Easysoft Administrator on the PC (this import will include a validation process), or

   b) Import the file in two stages. First, the CSV file should be imported into the server-based administrator, then download the catalog into the PC Administrator. The advantage of the two-stage process is that with large files, the overall import time will be quicker than by importing the CSV file directly into the PC-based Administrator

1. Set the database privileges (if the imported database is new)

2. Upload the definitions to the catalog on the server

The command syntax to run the converter is:

```
$ @EASYSOFT_SQL_SYSTEM:COBOLCNV <COBOL file specification>
[<options>]
```

Where <COBOL file specification> is the name of the COBOL file to convert. This name must include any file extension. If this parameter is omitted, a short listing of the options available is displayed, and then the routine stops.

<options> is one or more of the following, in any order.

Option	**Explanation**
/OUT=output_file	<output_file> is the name of the file which is generated by the conversion process. If <output_file> is not specified, then the output is sent to the screen.
/FILLERS	Outputs FILLER fields.
/SUBSTRUCTURES	Outputs group items (sub-structured fields).
/TURNON	Converts raw record descriptions (for example, from a text file containing these).

After the COBOL file has been parsed, you are asked to supply the name of the database and the default directory. See the next section, “Basic Conversion”, for details.

There is no limit to the size of the COBOL file which can be converted.

The converter has no effect on any other software.

If you wish to quit the routine whilst a conversion is taking place, then press **Ctrl-Y.**

If the FILE SECTION of the COBOL file contains invalid syntax, then a message is generated stating this, but processing will continue whenever possible, so that those definitions which are valid will be converted.

Basic Conversion

This example shows the basic options required to convert a COBOL file. Say there is a COBOL file called TESTDATA.TXT. To convert this and put the output in a file called TESTDATA.CSV, you would type:

```
$ @EASYSOFT_SQL_SYSTEM:COBOLCNV TESTDATA.TXT /OUT=TESTDATA.CSV

Easysoft COBOL converter Version <version>
<copyright notice>

Processing file
Parsing COBOL data

Database Name : TEST
Default Directory : TESTDIR
```

After an introductory message, you are asked for Database Name. This is the name that you want to call the database which holds the definitions which will be generated by the conversion. You must enter a value.

Default directory refers to the default location of the files stored on the server. It is optional; if a value is not entered, an empty string is generated in the output.

```
Converting parsed data to CSV data.

Conversion Statistics

Total number of FILE definitions          : <n>
<other conversion statistics

Conversion completed successfully
```

Comparison

Say the COBOL file contained these definitions:

```
FD EMPLOYEES
   <other definitions may appear here>

01 EMPLOYEE.
   03 SSN               PIC X(9).
   03 NAME              PIC X(30).
```

The CSV file that is generated would contain the following:

```
"FILE","EMPLOYEES","INDEXED","FIXED",39
"FIELD","EMPLOYEES","SSN","STRING",0,9,9,0,0,"",""
"FIELD","EMPLOYEES","NAME","STRING",9,30,30,0,0,"",""
"DB","TEST","TESTDIR","VAX-RMS",""
"TABLE","TEST","EMPLOYEE","TABLE","EMPLOYEES","EMPLOYEES",""
"COLUMN","TEST","EMPLOYEE","SSN","SSN","VARCHAR",9,1,1,""
"COLUMN","TEST","EMPLOYEE","NAME","NAME","VARCHAR",30,1,1,""
```

This is the database name that was entered at the start of the processing.

This is the default directory that was entered at the start of the processing.

The Database definition (4th line of the generated CSV) contains the name of the database and default directory (if any) that you entered.

Any hyphen characters in file or record definitions in the COBOL file (File or Field definitions in the Easysoft Administrator) are converted to underscore characters in the Table and Column definitions in the Easysoft Administrator.

Output FILLER Fields

By default, FILLER fields and fields beginning with FILLER- are not converted. If you want to see the FILLER fields, then use this option.

Because field names in the Easysoft Administrator (and corresponding SQL column names) must be unique within a record (row), a suffix is added to the FILLER fields. This suffix is _n, where n represents a character digit corresponding to the additional FILLER field.

Here is a contrived example. Say there was a record, FILLDEMO, defined as follows:

```
01 FILLDEMO.
   03 FIRSTNAME         PIC X(20).
   03 FILLER            PIC XXXX.
   03 LASTNAME          PIC X(20).
   03 FILLER            PIC X(12).
   03 FILLER            PIC X(16).
   03 FILLER-NUMERIC    PIC 99.
```

Without this option, the resulting column definitions would be:

```
"COLUMN","TEST","FILLDEMO","FIRSTNAME","FIRSTNAME","VARCHAR",20,1,1,"
"
"COLUMN","TEST","FILLDEMO","LASTNAME","LASTNAME","VARCHAR",20,1,1,""
```

To see all the FILLER fields, use the option, thus:

```
$ @EASYSOFT_SQL_SYSTEM:COBOLCNV TESTDATA.TXT /OUT=TESTDATA.CSV /FILLERS
```

Using this option, the resulting column definitions for the FILLDEMO record would be:

```
"COLUMN","TEST","FILLDEMO","FIRSTNAME","FIRSTNAME","VARCHAR",20,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER_1","FILLER_1","VARCHAR",4,1,1,""
"COLUMN","TEST","FILLDEMO","LASTNAME","LASTNAME","VARCHAR",20,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER_2","FILLER_2","VARCHAR",12,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER_3","FILLER_3","VARCHAR",16,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER_NUMERIC","FILLER-
NUMERIC","VARCHAR",
   2,1,1,"
```

Fields defined as numeric in COBOL (for example, PIC 99) are converted to character fields, because the NUMERIC datatype consists of digit characters.

Output Group Items (sub-structured fields)

If the COBOL file contains a hierarchical record structure (sub-structured fields) only the lowest level fields (elementary items) are converted by default.

For example, you may have the following COBOL substructure:

```
FD EMPLOYEES
   <other definitions may appear here>

01 EMPLOYEE.
   03 SSN               PIC X(9).
   03 NAME.
      05 FIRSTNAME     PIC X(14).
      05 LASTNAME      PIC X(16).
```

Visually, this equates to:

SSN	NAME	
	FIRST NAME	LAST NAME

Without this option only the lowest level is converted.

By default, the output from the conversion process would be:

```
"FILE","EMPLOYEES","INDEXED","FIXED",39
"FIELD","EMPLOYEES","SSN","STRING",0,9,9,0,0,"",""
"FIELD","EMPLOYEES","FIRSTNAME","STRING",9,14,14,0,0,"",""
"FIELD","EMPLOYEES","LASTNAME","STRING",23,16,16,0,0,"",""
"DB","TEST","TESTDIR","VAX-RMS",""
"TABLE","TEST","EMPLOYEE","TABLE","EMPLOYEES","EMPLOYEES",""
"COLUMN","TEST","EMPLOYEE","SSN","SSN","VARCHAR",9,1,1,""
"COLUMN","TEST","EMPLOYEE","FIRSTNAME","FIRSTNAME","VARCHAR",14,1,1,"
"
"COLUMN","TEST","EMPLOYEE","LASTNAME","LASTNAME","VARCHAR",16,1,1,""
```

The NAME field and its corresponding columns does not appear. To see the NAME field, you would use the option, thus:

```
$ @EASYSOFT_SQL_SYSTEM:COBOLCNV TESTDATA.TXT /OUT=TESTDATA.CSV /SUBSTRUCTURES
```

The output (only table and column definitions shown) would then be:

```
"TABLE","TEST","EMPLOYEE","TABLE","EMPLOYEES","EMPLOYEES",""
"COLUMN","TEST","EMPLOYEE","SSN","SSN","VARCHAR",9,1,1,""
"COLUMN","TEST","EMPLOYEE","NAME","NAME","VARCHAR",30,1,1,""
"COLUMN","TEST","EMPLOYEE","FIRSTNAME","FIRSTNAME","VARCHAR",14,1,1,
""
"COLUMN","TEST","EMPLOYEE","LASTNAME","LASTNAME","VARCHAR",16,1,1,""
```

NAME appears in output.

Convert Raw Record Descriptions

If you have a file containing valid COBOL record descriptions, and if that file is not part of a COBOL program proper, you can still convert the record descriptions. Say you had a text file which contained just this data:

```
01 EMPLOYEE.
   03 SSN              PIC X(9).
   03 NAME             PIC X(30).
```

you would use this option to convert it, thus:

```
$ @EASYSOFT_SQL_SYSTEM:COBOLCNV TESTDATA.TXT /OUT=TESTDATA.CSV /TURNON
```

The output would be:

```
"FILE","EMPLOYEES","INDEXED","FIXED",39
"FIELD","EMPLOYEES","SSN","STRING",0,9,9,0,0,"",""
"FIELD","EMPLOYEES","NAME","STRING",9,30,30,0,0,"",""
"DB","TEST","TESTDIR","<driver>",""
"TABLE","TEST","EMPLOYEE","TABLE","EMPLOYEES","EMPLOYEES",""
"COLUMN","TEST","EMPLOYEE","SSN","SSN","VARCHAR",9,1,1,""
"COLUMN","TEST","EMPLOYEE","NAME","NAME","VARCHAR",30,1,1,""
```

Notes / Information

If the COBOL file contains multiple definitions implemented using the REDEFINES or RENAMES keywords, then after import into the Easysoft Administrator the user should deselect columns from the relevant tables to prevent multiple views of the same data in an ODBC application.

Database privileges are not set, so the user is required to set them using the Easysoft Administrator before any data can be accessed.

PowerHouse PDL Converter

The Easysoft PowerHouse PDL Converter is used to convert PowerHouse data dictionary file definitions to the CSV form (Comma Separated Value) required by the PC-based Easysoft Administrator and the server-based Administrator. These file definitions are stored in PowerHouse Dictionary Language files, which are also known as PDL files.

Since QSHOW, which is part of the PowerHouse package, generates PDL files, the converter is also known as the QSHOW converter.

This text is based on the assumption that you know about the form and function of PDL files.

To use the PDL converter you will need write privileges to the server.

Using the Converter

This section describes the options available; examples for each option are presented after this.

The entire process of conversion and using the generated CSV file is:

1. Convert the PDL file

2. Either

a) Import the CSV file directly into the Easysoft Administrator on the PC (this import will include a validation process), or

b) Import the file in two stages. First, the CSV file should be imported into the server-based administrator, then download the catalog into the PC Administrator. The advantage of the two-stage process is that with large files, the overall import time will be quicker than by importing the CSV file directly into the PC-based Administrator

3. Set the database privileges (if the imported database is new)

4. Upload the definitions to the catalog on the server

The command syntax to run the converter is:

```
$ @EASYSOFT_SQL_SYSTEM:PDLCONVERT <PDL file specification> [<options>]
```

Where <PDL file specification> is the name of the PDL file to convert. This name must include the PDL file extension (if it exists). If this parameter is omitted, a short listing of the options available is displayed, and then the routine stops.

<options> is one or more of the following, in any order.

Option	Explanation
/OUT=output_file	<output_file> is the name of the file which is generated by the conversion process. If <output_file> is not specified, then the output is sent to the screen.
/FILLERS	Outputs multiple FILLER and FILLER-NUMERIC fields.
/SUBSTRUCTURES	Outputs sub-structured fields.
/DEL=delete_chars	Prevents the specified characters from being generated in the output. Used in cases where the PDL definitions contain characters which are not allowed in the Easysoft CSV definitions.
/DEF=define	Define condition names for use by the pre-processor. If condition variables are defined in the PDL file, you may want to convert the sections of the file to correspond with these condition variables.

After the PDL file has been parsed, you are asked to supply the name of the database and the default directory. See “Basic Conversion”, page 136, for details.

There is no limit to the size of the PDL file which can be converted.

The converter has no effect on any other software.

If you wish to quit the routine whilst a conversion is taking place, then press **`Ctrl-Y`**.

If the PDL file contains invalid PDL syntax, then a message is generated stating this, but processing will continue whenever possible, so that those definitions which are valid will be converted.

Basic Conversion

This example shows the basic options required to convert a PDL file. Say there is a PDL file called TESTDATA.PDL. To convert this and put the output in a file called TESTDATA.CSV, you would type:

```
$ @EASYSOFT_SQL_SYSTEM:PDLCONVERT TESTDATA.PDL /OUT=TESTDATA.CSV

Easysoft PowerHouse PDL Converter Version <version>
<copyright notice>

Pre-processing file
Processing file
Parsing PDL data

Database Name : TEST

Default Directory : TESTDIR
```

After an introductory message, you are asked for Database Name. This is the name that you want to call the database which holds the definitions which will be generated by the conversion. You must enter a value.

Default directory refers to the location of the data files stored on the server . It is optional; if a value is not entered, an empty string is generated in the output.

```
Converting parsed data to CSV data.

Conversion Statistics

Total number of FILE definitions          : <n>
<other conversion statistics

Conversion completed successfully
```

Comparison

Say the PDL file contained these definitions:

```
File EMPLOYEES   Organization INDEXED  CREATE

  Record EMPLOYEE
   Item SSN      Datatype CHARACTER   Size 9
   Item NAME     Datatype CHARACTER   Size 30
```

The CSV file that is generated would contain the following:

```
"FILE","EMPLOYEES","INDEXED","FIXED",39
"FIELD","EMPLOYEES","SSN","STRING",0,9,9,0,0,"",""
"FIELD","EMPLOYEES","NAME","STRING",9,30,30,0,0,"",""
"DB","TEST","TESTDIR","VAX-RMS",""
"TABLE","TEST","EMPLOYEE","TABLE","EMPLOYEES","EMPLOYEES",""
"COLUMN","TEST","EMPLOYEE","SSN","SSN","VARCHAR",9,1,1,""
"COLUMN","TEST","EMPLOYEE","NAME","NAME","VARCHAR",30,1,1,""
```

This is the database name that was entered at the start of the processing.

This is the default directory that was entered at the start of the processing.

The Database definition (4th line) contains the name of the database and default directory (if any) that you entered.

Any hyphen characters in Record or Item definitions in the PDL file (File or Field definitions in the Easysoft Administrator) are converted to underscore characters in the Table and Column definitions in the Easysoft Administrator.

Output FILLER Fields

If a record contains more than one FILLER field, then by default, only the first one is converted. If you want to see all the FILLER fields, then use this option.

FILLER_NUMERIC fields, and any other fields starting with the name FILLER- (hyphen) or FILLER_ (underscore) are not converted unless this option is used.

Because field names in the Easysoft Administrator (and corresponding SQL column names) must be unique within a record (row), a suffix is added to the second and subsequent FILLER fields. This suffix is _n, where n represents a character digit corresponding to the additional FILLER field.

Here is a contrived example. Say there was a record, FILLDEMO, defined as follows:

```
  Record FILLDEMO
    Item FIRSTNAME       Datatype CHARACTER   Size 20
    Item FILLER          Datatype CHARACTER   Size 4
    Item LASTNAME        Datatype CHARACTER   Size 20
    Item FILLER          Datatype CHARACTER   Size 12
    Item FILLER          Datatype CHARACTER   Size 16
    Item FILLER-NUMERIC   Datatype INTEGER Size  2
    Item FILLER-NUMERIC   Datatype INTEGER Size  2
```

Without this option, the resulting column definitions would be:

```
"COLUMN","TEST","FILLDEMO","FIRSTNAME","FIRSTNAME","VARCHAR",20,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER","FILLER","VARCHAR",4,1,1,""
"COLUMN","TEST","FILLDEMO","LASTNAME","LASTNAME","VARCHAR",20,1,1,""
```

To see all the FILLER fields, use the option, thus:

```
$ @EASYSOFT_SQL_SYSTEM:PDLCONVERT TESTDATA.PDL /OUT=TESTDATA.CSV
/FILLERS
```

Using the option, the resulting column definitions for the FILLDEMO record would be:

```
"COLUMN","TEST","FILLDEMO","FIRSTNAME","FIRSTNAME","VARCHAR",20,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER","FILLER","VARCHAR",4,1,1,""
"COLUMN","TEST","FILLDEMO","LASTNAME","LASTNAME","VARCHAR",20,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER_1","FILLER_1","VARCHAR",12,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER_2","FILLER_2","VARCHAR",16,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER_NUMERIC","FILLER-NUMERIC",
  "SMALLINT", 2,1,1,""
"COLUMN","TEST","FILLDEMO","FILLER_NUMERIC_1","FILLER-NUMERIC_1",
  "SMALLINT",2,1,1,""
```

Output Sub-structured Fields

If the PDL file contains sub-structured items implemented using the BEGIN STRUCTURE and END STRUCTURE elements these are not converted by default. Only the lowest level fields are converted.

For example, you may have the following PDL substructure:

```
File EMPLOYEES    Organization INDEXED CREATE

  Record EMPLOYEE
    Item SSN         Datatype CHARACTER   Size 9
    Item NAME        Datatype CHARACTER   Size 30
      BEGIN STRUCTURE
        Item FIRSTNAME   Datatype CHARACTER   Size 14
        Item LASTNAME    Datatype CHARACTER   Size 16
      END STRUCTURE
```

Visually, this equates to:

SSN	NAME	
	FIRST NAME	LAST NAME

Without this option only the lowest level is converted.

By default, the output from the conversion process would be:

```
"FILE","EMPLOYEES","INDEXED","FIXED",39
"FIELD","EMPLOYEES","SSN","STRING",0,9,9,0,0,"",""
"FIELD","EMPLOYEES","FIRSTNAME","STRING",9,14,14,0,0,"",""
"FIELD","EMPLOYEES","LASTNAME","STRING",23,16,16,0,0,"",""
"DB","TEST","TESTDIR","VAX-RMS",""
"TABLE","TEST","EMPLOYEE","TABLE","EMPLOYEES","EMPLOYEES",""
"COLUMN","TEST","EMPLOYEE","SSN","SSN","VARCHAR",9,1,1,""
"COLUMN","TEST","EMPLOYEE","FIRSTNAME","FIRSTNAME","VARCHAR",14,1,1,"
"
"COLUMN","TEST","EMPLOYEE","LASTNAME","LASTNAME","VARCHAR",16,1,1,""
```

The NAME field and its corresponding columns does not appear. To see the NAME field, you would use the option, thus:

```
$ @EASYSOFT_SQL_SYSTEM:PDLCONVERT TESTDATA.PDL /OUT=TESTDATA.CSV
/SUBSTRUCTURES
```

The output (only table and column definitions shown) would then be:

NAME appears in output.

```
"TABLE","TEST","EMPLOYEE","TABLE","EMPLOYEES","EMPLOYEES",""
"COLUMN","TEST","EMPLOYEE","SSN","SSN","VARCHAR",9,1,1,""
"COLUMN","TEST","EMPLOYEE","NAME","NAME","VARCHAR",30,1,1,""
"COLUMN","TEST","EMPLOYEE","FIRSTNAME","FIRSTNAME","VARCHAR",14,1,1,""
"COLUMN","TEST","EMPLOYEE","LASTNAME","LASTNAME","VARCHAR",16,1,1,""
```

Stripping Characters

Characters which are valid in the PDL definition may not be valid in the Easysoft PC Administrator, for example, the $ character. (Currently, the CSV format supported by the Administrator is a subset of the CSV which can be used in the catalog and by the IMPORT CSV and EXPORT CSV routines on the server).

To prevent invalid characters from being generated in the output file, use the option thus:

```
$ @EASYSOFT_SQL_SYSTEM:PDLCONVERT TESTDATA.PDL /OUT=TESTDATA.CSV /DEL=&
```

In the case exemplified above, the ampersand character (&), if it existed in the PDL record or item definitions, would not appear in the CSV output.

PDL definitions contain a section called System Options, and within this, there may be a section entitled Special Name Characters. Any characters that are listed in

These characters will not appear in the output file.

Special Name Characters will not appear in the output file. There is no need to use this option to remove them.

```
System Options                             &
   <definitions>                           &
   Special Name Characters "-_'%#"         &
   <definitions>                           &
```

Define Conditions for use by the Pre-processor

Say that the following record structure is defined in PDL.

```
File NAMES    Organization INDEXED CREATE
@IF GB
  Record NAME
    Item FIRSTNAME       Datatype CHARACTER   Size 18
    Item LASTNAME        Datatype CHARACTER   Size 20

@ELSEIF FR
  Record NOM
    Item PRENOM          Datatype CHARACTER   Size 18
    Item NOM             Datatype CHARACTER   Size 20
@ENDIF
```

If the converter is used without this option, then the output will not contain information for either the record called NAME, or the record called NOM (because neither of the condition variables is set to TRUE). If you ran the converter, you would see the following definitions only:

```
"FILE","NAMES","INDEXED","FIXED",1
"DB","TEST","TESTDIR","VAX-RMS",""
```

To convert, for example, the French version (condition variable FR), you would use this option to set the condition variable, thus:

```
$ @EASYSOFT_SQL_SYSTEM:PDLCONVERT TESTDATA.PDL /OUT=TESTDATA.CSV /DEF=FR
```

The output would be:

```
"FILE","NAMES","INDEXED","FIXED",38
"FIELD","NAMES","PRENOM","STRING",0,18,18,0,0,"",""
"FIELD","NAMES","NOM","STRING",18,20,20,0,0,"",""
"DB","TEST","TESTDIR","VAX-RMS",""
"TABLE","TEST","NOM","TABLE","NAMES","NAMES",""
"COLUMN","TEST","NOM","PRENOM","PRENOM","VARCHAR",18,1,1,""
"COLUMN","TEST","NOM","NOM","NOM","VARCHAR",20,1,1,""
```

You can set as many condition variables as you like; each one must be preceded by the -d option. The order in the command line is not important. The output depends upon the structure of the PDL file; output is based on the first condition that is TRUE. Say there were three conditions:

```
@IF GB OR US
  <definition 1>
@ELSEIF FR
  <definition 2>
@ELSEIF DM
  <definition 3>
@ENDIF
```

If the define options contained: **`-d FR -d US`**, then <definition 1> would be converted.

Notes / Information

If the PDL file contains multiple defined fields implemented using the REDEFINES keyword in an ITEM statement, then after importation into the Easysoft Administrator the user should deselect columns from the relevant tables to prevent multiple views of the same data appearing in an ODBC application.

Database privileges are not set, so the user is required to set them using the Easysoft Administrator before any data can be accessed.

There is no un-install option for the software. To remove the software, delete the compressed file that was transferred to the server and the files that were generated during the installation.

UserBase Converter

The Easysoft UserBase Converter is used to convert UserBase data file definitions to the CSV form (Comma Separated Value) of file definitions required by the PC-based Easysoft Administrator and the server-based Administrator.

It is assumed that you know how to use the USE DESCRIBE function within the UserBase package.

To use the UserBase Converter, you will need write privileges to the server.

Using the Converter

The entire process of conversion and using the generated CSV file is:

1. Convert the UserBase data to CSV format:
   Generate a .LIS file from within UserBase. Ensure the listing width is set to 132. This is the input to the Easysoft Converter. (The process is described in detail in the next section).

2. Either

a) Import the CSV file directly into the Easysoft Administrator on the PC (this import will include a validation process), or

b) Import the file in two stages. First, the CSV file should be imported into the server-based Administrator, then download the catalog into the PC Administrator. The advantage of the two-stage process is that with large files, the overall import time will be quicker than by importing the CSV file directly into the PC-based Administrator.

3. Define tables and columns to map onto the file definitions using the PC-based Easysoft Administrator. If you are creating a new database, then you will also need to define a database, users and privileges.

4. Upload the definitions to the catalog on the server.

The command syntax to run the converter is:

```
$ @EASYSOFT_SQL_SYSTEM:BASECNV <UserBase .LIS file specification>
      /OUT=output_file
```

Where <UserBase .LIS file specification> is the name of the .LIS file to convert. This name should include the .LIS file extension. If this parameter is omitted, a message explains the correct calling sequence and then the routine stops.

<output_file> is the name of the file which is generated by the conversion process.

If <output_file> is not specified in the command line, then when the routine runs, you are asked for the name of the output file. If you do not enter a name, but just press the **`Enter`** key, then the output is sent to the screen.

Within each .LIS file, the maximum number of defined files that can be converted is 300. The maximum number of fields which can be converted is 100,000.

The converter has no effect on any other software.

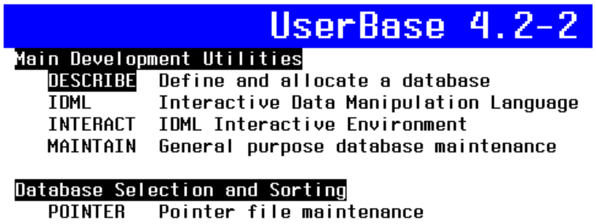
If you wish to quit the routine whilst a conversion is taking place, then press **`Ctrl-Y.`**

Conversion Example

Assume that a small file entitled UBEMPLOYEE has been defined within UserBase. This file contains three fields. The description of the file is contained in a UserBase file with an extension .DES. Files with the extension .DAT are the data files which are described by the .DES files. When the list file is created, the .DES files and the DAT files should reside in the same directory (this is the default for UserBase files, so it is unlikely that you will have to move any files).

The Easysoft UserBase Converter takes as input a UserBase List file (.LIS). Follow these steps to generate a .LIS file from the UserBase field definitions.

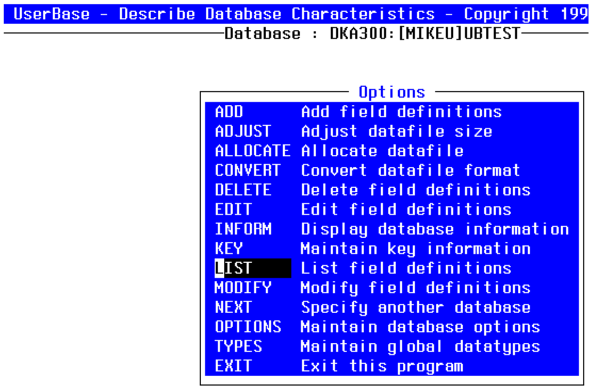
1. Select the **DESCRIBE** menu option.



```
                          UserBase 4.2-2
Main Development Utilities
   DESCRIBE  Define and allocate a database
   IDML      Interactive Data Manipulation Language
   INTERACT  IDML Interactive Environment
   MAINTAIN  General purpose database maintenance

Database Selection and Sorting
   POINTER   Pointer file maintenance
```

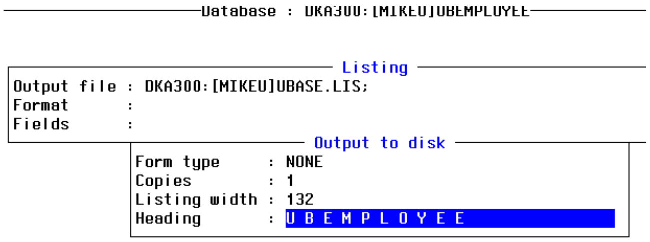
2. Select the **LIST** menu option.



```
 UserBase - Describe Database Characteristics - Copyright 199
──────────────────────Database : DKA300:[MIKEU]UBTEST──────

                     ─────── Options ───────
                     ADD       Add field definitions
                     ADJUST    Adjust datafile size
                     ALLOCATE  Allocate datafile
                     CONVERT   Convert datafile format
                     DELETE    Delete field definitions
                     EDIT      Edit field definitions
                     INFORM    Display database information
                     KEY       Maintain key information
                     LIST      List field definitions
                     MODIFY    Modify field definitions
                     NEXT      Specify another database
                     OPTIONS   Maintain database options
                     TYPES     Maintain global datatypes
                     EXIT      Exit this program
```

3. Enter the name of the output file.



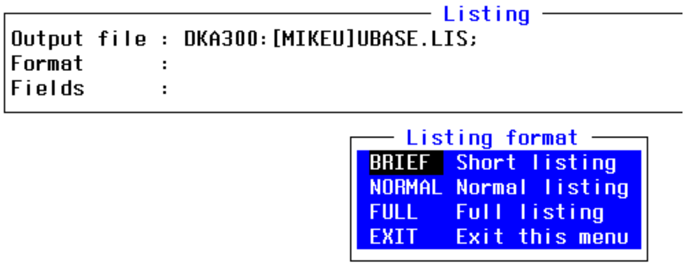
```
────────────────────Database : DKA300:[MIKEU]UBEMPLOYEE────────────

 ───────────────────────────────────── Listing ────────────
 Output file : DKA300:[MIKEU]UBASE.LIS;
 Format      :
 Fields      :
              ────────────────── Output to disk ──────────
              Form type     : NONE
              Copies        : 1
              Listing width : 132
              Heading       : U B E M P L O Y E E
```

The Output to disk submenu appears. User the default values:
Form type: NONE
Listing width: 132

4. After selecting the Heading, the Listing format submenu appears

5. Select **BRIEF**



```
 ─────────────────────────────────── Listing ─────────
 Output file : DKA300:[MIKEU]UBASE.LIS;
 Format      :
 Fields      :
 ─────────────────────────────────────────────────────
                          ──── Listing format ────
                           BRIEF   Short listing
                           NORMAL  Normal listing
                           FULL    Full listing
                           EXIT    Exit this menu
```

6. The list file is generated on disk.

7. Convert the list file (*.LIS) using the Easysoft UserBase Converter. Say the file that was generated in step 3 was called UBASE.LIS; To convert this and put the output in a file called UBASE.CSV, you would type:

```
$ @EASYSOFT_SQL_SYSTEM:BASECNV UBASE.LIS /OUT=UBASE.CSV
```

```
Easysoft USERBASE Converter Version <version>
   <copyright notice>

   Converting BASE data.
   Conversion Statistics

   Total number of FILE definitions         : 1
   Total number of FIELD definitions        : 3

   Outputting FILE definitions
   Outputting FIELD definitions

   Conversion completed successfully
```

Comparison

Assume that the UserBase list file contained these definitions:

```
Data file   : UBEMPLOYEE
Format      : RMS relative
Record size : 39
Attributes  : None
Definitions last changed on <date> <time> by MIKEU from _TNA41:

<date> <time>           U B E M P L O Y E E                 Page 1

Field Name      Description         DType   Len   Start-End  Rec Attributes
----- ---------- -------------------- ------- ----- ----------- --- ----------
1     SSN       Social Sec. Number  T       9        1-9      0  I,U
2     FIRSTNAME Employee First Name T       14      10-23     0  I,U
3     LASTNAME  Employee Last Name  T       16      24-39     0  I,U
```

The CSV file that is generated would contain the following:

```
"FILE","UBEMPLOYEE","INDEXED","FIXED",39
"FIELD","UBEMPLOYEE","SSN","STRING",0,9,9,0,0,"",""
"FIELD","UBEMPLOYEE","FIRSTNAME","STRING",1,14,14,0,0,"",""
"FIELD","UBEMPLOYEE","LASTNAME","STRING",2,16,16,0,0,"",""
```

This is the name of the data file.

Field names.

IQ Converter

This utility converts IQ2 and IQ3 data dictionary files to the CSV form (Comma Separated Value) of file definitions required by the PC-based Easysoft Administrator and the server-based Administrator.

The command to run the converter is:

```
$ @EASYSOFT_SQL_SYSTEM:IQCONVERTER <IQ input file> <output file>
```

where
<IQ input file> is the name of the IQ data dictionary file and
<output file> is the name of the file generated by the converter. This file can then be imported into the Easysoft Administrator.

Conversion Example

```
$ @EASYSOFT_SQL_SYSTEM:IQCONVERTER <ddmaster.dat> <ddmaster.csv>

Easysoft IQ converter Version <version>
<copyright notice appears here>.
All rights reserved.

Converting IQ (version x) data...

Processing DATA definitions
<other definitions appear here>

Conversion Statistics

Total number of FILE definitions        : 15
<other statistics appear here>
```

Notes / Information

Any fields containing time information are automatically converted to LONGWORD field format rather than BINTIME format as there is no way to differentiate between the data types with the information given. Fields that are known to be times can be converted to BINTIME format later.

CHAPTER 12

Troubleshooting

This chapter deals with various troubleshooting issues related to the installation and use of Easysoft ODBC for RMS. Solutions to some common problems (presented in the form of Frequently Asked Questions) are given.

Server-side Troubleshooting

- Why doesn't the Easysoft service accept requests after the server has been rebooted?

  To ensure the Easysoft service continues to operate after a machine reboot a few commands need to be added to the system startup file. This should normally be done as part of the software installation (see "Step 12. Edit the System Startup File", page 24).

  If running TCP/IP then the necessary command should be executed to ensure that TCP/IP is still accepting requests for the Easysoft service after the reboot.

  NOTE: This varies depending on the TCP/IP stack in use. For example, for DEC UCX the following line should be added:

  ```
  $ UCX ENABLE SERVICE EASYSOFT
  ```

- How do I send data to Easysoft?

  1. Preparation

  If you are requested to send your data to Easysoft, you need to make sure that you send both the Easysoft Catalog and the data. Before you back up the data make sure that no one is using the software by using the Host Administrator SHOW USERS command.

2. Sending data

- On floppy

  Back up the catalog and the data to a disk file. For example,

```
$ BACKUP/REWIND/VERIFY EASYSOFT_SQL_CATALOG:*.*,DATA:*.* -
                                EASYSOFT.BCK/SAVE/LOG
```

Create a Zip file using ZIP. For example,

```
$ ZIP:==$EASYSOFT_SQL_SYSTEM:ZIP.EXE
$ ZIP "-V" EASYSOFT.ZIP EASYSOFT.BCK
```

  Send the zipped files on a floppy disk to Easysoft.

- Electronically

  Create a .ZIP file (see above) and do either of the following:

  email the ZIP file to support@easysoft.com
  copy the ZIP file to ftp.easysoft.com/pub/upload

- How do I create a service on the server?

  To create a service on the server you must use the EASYSOFT_SQL_SYSTEM:CREATE_SERVICE.COM procedure.

```
$ SET DEF EASYSOFT_SQL_SYSTEM
$ @CREATE_SERVICE
```

  The calling sequence is as follows:

```
$ @CREATE_SERVICE p1 p2 p3
```

  Where:

  p1 = The name of the communications software (that is, network transport. See page 21 for a list of valid options.)

  p2 = The name of the network service, for example, EASYSOFT

  p3 = The port or object number (that is, 0 for DECNET, 7777 for TCP/IP)

- What do I do if I have UCX and I get the message:
  `[Easysoft][ESODBC]WS-E-CONNECT: Windows socket error ‘Connection has been refused by remote host’ (10061) occurred connecting to service ‘7777’ on ‘<server>’ (#6553762) ?`

  This error message means your Easysoft service on the server is disabled.

  To correct the state of your Easysoft service log onto your VAX and at the system prompt type:

  ```
  $ UCX SHOW SERVICES
  ```

  The output includes:

  ```
  Service   Port  Proto Process           Address         State
  EASYSOFT  7777  TCP   EASYSOFT          0.0.0.0         Disabled
  ```

  To enable the Easysoft service type:

  ```
  $ UCX ENABLE SERVICE EASYSOFT
  ```

- What do I do if I have DECNET and I get the error message:
  `DECnet not installed correctly` ?

  This error message usually occurs when you are setting up a data source. It implies that the Windows side of DECNET has not been set up. To overcome this problem it is necessary to edit the `system.ini` file in your Windows directory.

  In the section `[386Enh]` modify the `network=` line to include either **`decpw.386`** if you have Pathworks 5.x or **`decnet.386`** if you run Pathworks 4.1.

- What do I do if I have DECNET and I get the error message:
  `ODBC-call failed. [Easysoft] [ESODBC]DNET-E-SOCKET:DECnet error ‘No buffer space available’ occurred creating socket (#589830)` ?

  This error message appears when the PC has filled all the available sockets. A socket is used each time a connection is made to the host; if the application using the socket has an error and closes, then a socket is left open.

  To remove all open sockets it is necessary to close all applications, quit Windows and reboot the PC. If no applications closed and left sockets open, then follow the instructions for increasing the number of sockets (see next question).

- How do I increase the number of sockets?

  It is best to work at the DOS command level. From your Pathworks directory type: **`ncp show exec char`**

  This displays information about the maximum number of links; an edited example is given below.

  ```
  Executor Characteristics <date>

  Driver version number        = 5.1.195
  State                        = On
  Executor Identification      = PATHWORKS V5.0

  Maximum links                = 2

  Remote adapter names         = 3
  ```

  This line states the number of links

  It is necessary to increase the number of links to 16; to do this type the following:

  ```
  ncp define exec max links 16
  ```

  Now re-boot your machine for the changes to become permanent.

- What do I do if I have DECNET and I get the error message: `DNET-E-NODEBYNAME Unable to obtain DECnet address for the node <node name>` ?

  This error has occurred because the node has not been defined on the client. It is necessary to define the node when this happens. How exactly you do this depends upon your local conditions, but for example, under Pathworks 5.1, in the Pathworks / DECNET directory type the following using the DOS based NCP command:

  ```
  NCP DEFINE NODE <node number> NAME <machine name>
  ```

  Replace <node number> with the node number for the machine and replace <machine name> with the machine name.

- How can I test that the VAX / Alpha process is running when using DECNET?

  Enter the following (the word “type” should be included):

  ```
  type <node>"<user name> <password>"::"0=easysoft"
  ```

Replace

<node> with the VAX / Alpha node name.
<user name> with the user's VAX / Alpha account name.
<password> with the user's VAX / Alpha password.

Much of the output may appear garbled. This is not an error, but is due to the output of binary data. Included in the output should be a few lines which include the word: `ESNET`. To stop the output press **`CTRL-Y`**.

- How do I determine what Easysoft logicals are set up?

  At the server prompt type:

  ```
  $ SHOW LOG EASYSOFT*
  ```

  A list of logicals which are set up by the installation is shown.

Client-side Troubleshooting

- During the installation of the ODBC driver I see the message: `Cannot install file <directory> <file>.DLL.  It might be in use. ...`

  We recommend that you close all applications, and continue with the operation. If you still get this message, then log off, log on again, close any applications that may have opened automatically, and then re-install the Easysoft ODBC driver.

- Why is the **Add** button on the Data Sources dialog box greyed out?

  The basic reason for this is that there are no installed drivers with which to connect a data source. The two causes of this are:

  1. The Easysoft driver seemed to be installed and the Data Sources dialog box duly appeared. However, if the Easysoft ODBC driver in the list box was not highlighted, then it will not have been installed.

  2. All installed drivers have been deleted.

  Solution: Add the required data source by following the procedure in “Adding a Data Source”, page 46.

- Why can't I add data sources?

  This can happen after a General Protection Fault (GPF) has caused the Microsoft ODBC Administrator to crash on a previous ODBC setup.

  Solution: shut down and then restart Windows.

- I can't find the ODBC icon - it isn't where it should be. What should I do?

  You may have run Setup, but if the Microsoft ODBC Administrator had already been installed someone may have changed the location of the icon. Running Setup does not change the location of the icon.

  Solution 1: Find out where the icon is

  Solution 2: Start the Microsoft ODBC Administrator without using the icon.
  Windows 95: **Start**, **Run…**
  Windows NT; **File**, **Run…**
  In the Command line of the dialog box type the location of the Microsoft ODBC executable file, followed by the file name (ODBCAD32.EXE).

- When I run **Test** from the ODBC Setup dialog box, the result is a timeout; why?

  This could be due to the server being busy.

  Solution: Increase the Connect Time on the Easysoft ODBC Settings dialog box (Figure 18).

- The Add Data Source dialog box shows an installed driver, but selecting **OK** does nothing - what's the reason for this?

  This can arise after a GPF. Solution: shut down and then restart Windows.

- No Transports are available in the list in the Easysoft ODBC Setup dialog box (Figure 16); the transport shown in the list box is garbage and the pull-down is empty.

  This can happen after some error condition has arisen.

  Solution: shut down and then restart Windows.

- I used to be able to use Easysoft ODBC, but now I get a message saying I haven’t got a valid licence:
  `No client licences are available for product <product>.`

  This could be because there have been changes to the PC hardware. A new Client Licence slot is needed because the system uses PC hardware details in the allocation of Client Licence slots (and these details are stored by the licensing software). If the maximum number of client users has been reached, then no additional slots are available.

  Solution: remove all user/product combinations for the (now redundant) user by means of the Host Administrator REMOVE USER command. This frees the licence slots which had been allocated on the basis of the old PC hardware. You should now be able to connect to the server.

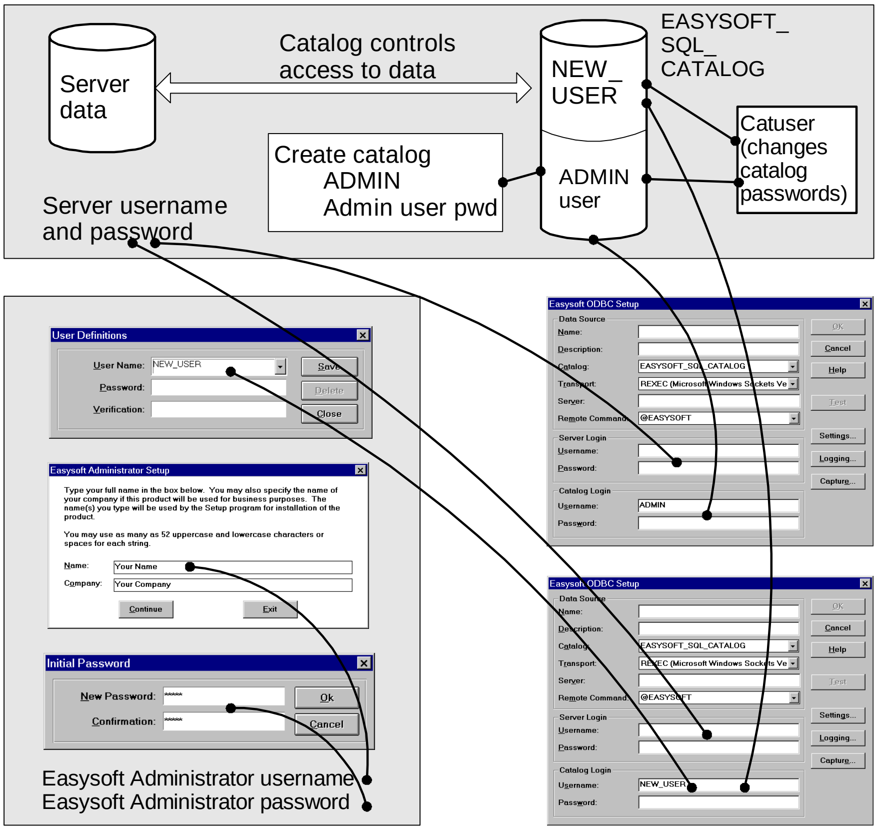
- When I use Easysoft ODBC I get the error message: `Invalid driver specified in ES_DRIVER`

  This is because you licensed the product, but did not install it on the server. Typically, this can happen if you intend to install two or more Easysoft products, since they can be licensed in one operation.

Appendices

Names and Passwords

Figure 87 shows the password relationships between components in the Easysoft system.



User Definitions

User Name: NEW_USER
Password:
Verification:

Easysoft Administrator Setup

Type your full name in the box below. You may also specify the name of your company if this product will be used for business purposes. The name(s) you type will be used by the Setup program for installation of the product.

You may use as many as 52 uppercase and lowercase characters or spaces for each string.

Name: Your Name
Company: Your Company

Initial Password

New Password: *****
Confirmation: *****

Easysoft ODBC Setup

Data Source
Name:
Description:
Catalog: EASYSOFT_SQL_CATALOG
Transport: REXEC (Microsoft Windows Sockets Ve
Server:
Remote Command: @EASYSOFT
Server Login
Username:
Password:
Catalog Login
Username: ADMIN
Password:

Easysoft ODBC Setup

Data Source
Name:
Description:
Catalog: EASYSOFT_SQL_CATALOG
Transport: REXEC (Microsoft Windows Sockets Ve
Server:
Remote Command: @EASYSOFT
Server Login
Username:
Password:
Catalog Login
Username: NEW_USER
Password:

Figure 87. Password Relationships

The ADMIN user and ADMIN user's password are defined when the Easysoft Catalog is created on the server. EASYSOFT_SQL_CATALOG is used as an example, but the principles apply to any other Catalog which is created. The only exception to this is the case of EASYSOFT_SQL_DEMO_CATALOG which is automatically created during the Easysoft Server Component installation; the ADMIN user's password is automatically set to ADMIN.

The Easysoft Administrator user creates new users and passwords (and changes existing passwords). The Catalog, which is downloaded from the server before changes can be made, must be uploaded before changes are registered in the system.

The Server Login Username and Password in the Easysoft ODBC Setup dialog box (in Easysoft ODBC) are those needed to log on to the server. The Catalog Login Username and Password are either those belonging to one of the new users created using the Easysoft Administrator or those belonging to the ADMIN user. **Note**: When the data source is set up, leaving the Server and Catalog login text boxes blank allows one data source to be used by all users (they are prompted to input their names and passwords). Additionally, this method means that passwords are not stored on the PC (Note: all passwords that are stored are always encrypted).

Table 5 lists the restrictions associated with data source names and Easysoft usernames and passwords.

Table 5. Naming restrictions in the Easysoft system

Name	**Min. chars**	**Max. chars**	**Allowed characters and/or restrictions**
Easysoft Administrator Company	1	52	no restrictions
Easysoft Administrator username	1	52	no restrictions
Easysoft Administrator password	1	48	no restrictions
Catalog directory	1	254	server operating system file specification applies
Catalog username Catalog password	1	64	First character: A to Z Other characters: no restrictions, but PC administrator only allows a to z, A to Z, 0 to 9, underscore (_). Other characters, if desired, must be added to the CSV using a text editor.
Data source name	1	32	First letter must be alphabetic not valid: [ ] { } ( ) ? * = ! @ , ;

APPENDIX B

Files Used

This appendix lists some of the files which are used by Easysoft ODBC for RMS.

Easysoft ODBC

The following Easysoft files are stored in the Windows System directory:

ESNET32.DLL
This Easysoft file is used for network connectivity.

ESODBC32.DLL
This is the 32 bit Easysoft ODBC driver.

The following files are stored in the Windows directory:

ODBCAD32.EXE
This is the Microsoft ODBC Administrator - it is supplied with Easysoft ODBC (and with other vendors' ODBC drivers).

ODBC.INI
This file, which is created and modified by the Microsoft ODBC administrator, stores information about the ODBC data sources which have been set up.

ODBCINST.INI
This Microsoft file contains a list of the ODBC drivers which are installed on the PC.

Easysoft Administrator

When the Easysoft Administrator is installed a directory is created; by default it is C:\EASYSOFT\SQL\SYSTEM. For convenience, this is referred to as the Easysoft directory. Various Easysoft files are stored in this directory - some of these are explained here.

ESADMENG.HLP
This is the on-line help file. It is accessible from within Easysoft SQL program group and from within the Administrator.

ESADMIN.DAT
This file is where all local data downloaded from the Easysoft Catalog on the server plus other information relevant to the data definition, such as imported definitions. Thus,

changes to an Easysoft Catalog do not need to be uploaded between Easysoft Administrator sessions or Windows sessions. (They must, of course, be uploaded before they can be used).

ESADMIN.EMP
This file has a similar structure to that of ESADMIN.DAT. It is an initial copy which stores all the original definitions before any changes are made.

ESADMIN.EXE
The Administrator program.

ESADMIN.LDB
This file controls locking of records within the Easysoft Administrator and is not initially present when Easysoft is first installed.

ESSYSTEM.SEC
This is a security file which is used in conjunction with ESADMIN.DAT.

README.WRI
This file contains the latest release notes. It is accessible from within Easysoft SQL program group.

EASYSOFT.INI

The Easysoft Administrator installation process places new files and modifies existing files in the Windows directory. Of relevance is the EASYSOFT.INI file, which is described here.

This file controls the Easysoft Administrator and Easysoft ODBC. An example is shown below, and following this, some important settings/options are described in detail.

```
[Options]
SystemDB=C:\EASYSOFT\SQL\SYSTEM\essystem.sec
[Administrator]
Path=C:\EASYSOFT\SQL\SYSTEM
Language=ENG
Username=Mike
Company=Easysoft Limited

[LOGGING]
ODBC=1
NET=1
```

Explanation of Sections

The order of the sections in the EASYSOFT.INI file is not important.

Comments can be included by prefixing a line with a semicolon character (;).

[LOGGING]

ODBC=1 To turn on the **Trace ODBC** option in the Easysoft ODBC Logging dialog box, set this value to 1. The log file that is generated is called \ESODBC.LOG.

NET=1 To turn on the **Trace Network** option in the Easysoft ODBC Logging dialog box, set this value to 1. The log file that is generated is called \ESNET.LOG.

[NETWORK]

IgnoreWinsock=1 If Pathworks is being used, setting this flag to 1 disables Winsock.

[SETTINGS]

messages=0 To turn off informational messages generated by the server.

[Options]

AllowReservedWords=1 This option lets you use the SQL reserved words defined in the section entitled “SQL Reserved Words”, page 179. The default is 0 (that is, the line can be omitted).

[Administrator]

This section contains information relating to the Administrator setup.

Path Location of Easysoft files.

Language Defines the language; currently only English is supported.

Username The name of the user as specified in the Easysoft Administrator Setup (User Details) dialog box.

Company The name of the company as specified in the Easysoft Administrator Setup (User Details) dialog box.

DatabaseRepair=True Repairs corrupt local database.

ThreeD=False The Easysoft Administrator uses the Microsoft file CTL3D.DLL which gives a 3-D effect to Windows dialog boxes. Sometimes, other applications can interfere with the correct functioning of this. One solution is to not use the file by: ThreeD=False.

APPENDIX C

Glossary

$ DCL prompt.

! DCL comment separator.

Alpha A type of computer produced by Digital Equipment Corporation (DEC) which can use the VMS and OpenVMS operating systems.

ANSI (American National Standards Institute) The primary formal standards-making body in the United States.

API (Application Programming Interface). A set of related functions that a computer programmer uses to obtain some kind of service from another piece of software.

application (In ODBC terms) a program that processes data and which runs under Microsoft Windows.

ASCII (American Standard Code for Information Exchange) A set of 8-bit binary numbers which represent the alphabet, punctuation, numerals and other special symbols which are used to represent text.

big-endian A computer architecture in which, within a given multi-byte numeric representation, the most significant byte has the lowest address (the word is stored "big-end-first").

bit A binary digit (0 or 1). The smallest unit of information in a binary notation system.

block (In general terms) the smallest unit of space into which a mass storage device can be divided.
(In OpenVMS terms) The smallest logically addressable unit of data that a device can transfer in an I/O (Input/Output) operation. Typically, 512 contiguous bytes.

Boolean Boolean logic is the two-valued logic of true and false.

buffer A temporary storage space for data.

byte A binary character string consisting of bits operated on as a unit.

catalog (In SQL terms) a named collection of one or more tables grouped together. The catalog contains definitions that describe SQL features of application databases, such as which columns belong to which tables, user privileges, etc.

column the vertical dimension of a table (compare row). The field at the intersection of a row and a column holds the value in accordance with the data type specified.

Comma Separated Text see CSV.

conformance level (API) Refers to the set of ODBC functions that an ODBC driver supports.

conformance level (SQL) Refers to the set of SQL functions that an ODBC driver supports.

CST (Comma Separated Text). see CSV.

CSV (Comma Separated Values). CSV is text that defines data values. The values are separated by commas (same as CST).

data source A set of database files plus (if appropriate) the associated operating system, DBMS and network.

data type A data type is the specification of permitted values. A data type limits the values which are allowed to be used.

database A collection of data files.

DBMS (Database Management System). Software that handles access to a database.

DCL Digital Command Language. The standard common interface to Digital's major operating systems.

DCL prompt By default this is the dollar sign ($). It indicates that DCL is ready to accept a command.

DECNET A proprietary communication protocol produced by Digital Equipment Corporation (DEC).

default A value automatically used in the case where a user has not specified a value.

DLL (Dynamic Link Library). A shared library of code that is loaded in and out of memory as and when it is needed.

download(ing) In non-technical terms downloading is taking data from the server and storing it on the client PC (compare uploading).

driver See ODBC driver.

Driver Manager A dynamic link library (DLL) provided by Microsoft, the main function of which is to load ODBC drivers.

field A segment of a data record.

File Definition A file definition in the Easysoft Administrator contains information relating to the organisation and length of a file stored on the server platform.

file organisation (in VAX/OpenVMS terms) The file structure that is used as the physical arrangement of the data on the storage medium. RMS file organisations are sequential, indexed and relative.

file specification A unique identification for a file which gives its physical location, the file type and the version number.

floating (1) A number that may be positive or negative but that has a whole (integer) portion and a fractional (decimal) portion. (2) An arithmetic operation in which the decimal point is not fixed, but placed automatically in a correct position.

FTP (File Transfer Protocol). A standard method (rfc 959) of transferring files between different machines.

function (SQL) A function in SQL takes a *scalar value* or a set of scalar values and returns a scalar value.

hexadecimal Pertaining to a number system using the base 16.

image (in VAX/OpenVMS terms) Procedures and data bound together by the linker. There are three types of image: executable, shareable and system

indexed file organisation (in VAX/OpenVMS terms) A file organisation in which a file contains records and a primary key index (and possibly other alternate key indexes) which is used to process the records sequentially by index or randomly by index.

interoperability An application that is interoperable is one that can access many different databases.

key (in DEC terms) A string or numeric data that specifies a particular record that is accessed randomly. In indexed files the user defines the length and location within the records. RMS uses the key to build an index. In relative files, key refers to the relative record number of each data record in the data file. RMS uses the relative record number to identify and access data records.

Licence Key A number (in hexadecimal format) which is provided by Easysoft and which is used in the licensing process.

literal A literal is a way of representing a value. Each value has a *data type*, and for each data type there is a corresponding literal specification.

little-endian A computer architecture in which, within a given word, bytes at lower addresses have lower significance (the word is stored "little-end-first").

local data type A local data type represents the storage format of the data within the data files on the server. This local data type is mapped onto an SQL data type within the Easysoft SQL Engine.

logical see logical name.

logical name In the OpenVMS and VMS operating systems, an alias for a file specification (can include directories, subdirectories, input/output devices). It is a named variable which is replaced by a value when it is used.

multi-tier driver (also see single-tier driver) In a multi-tier configuration the ODBC driver sends SQL requests to a server that processes those requests. Typically, multiple-tier systems are divided across platforms.

nibble Four contiguous bits of memory; one half of a byte.

ODBC (Open DataBase Connectivity). An industry standard defined by Microsoft which is by software that allows communication between different database systems.

ODBC driver The software that implements ODBC function calls. Each driver is specific to an application.

OpenVMS An operating system that runs on Alpha and VAX machines (see VMS).

operating system (1) An integrated collection of programs that controls the execution of computer programs and performs system functions. (2) Software that organises a central processor and peripheral devices into an active unit for the development and execution of programs.

packed decimal (in VAX/OpenVMS terms) A method of representing a decimal number by storing a pair of decimal digits in 1 byte. This can be done because there are 8 bits in 1 byte and only 4 bits are needed to represent any one of the numbers 0 through to 9.

packet In communications technology, a packet is the smallest unit of information that can be transmitted over a network.

parameter A value that is passed to a command.

parse To break a string of characters into primary components for the purpose of interpreting the string.

precision The precision of a numeric field is the maximum number of digits used by the data type. The precision of a non-numeric field is either the maximum length or the defined length of the field.

privilege A privilege is the right to perform a particular action (for example, INSERT) on an object (for example, a table).

record A collection of related data items treated as a unit. A record contains one or more fields.

relative file organisation (in VAX/OpenVMS terms) The arrangement of records in a file in which each record occupies a cell of equal length.

RMS (Record Management Services) A DIGITAL-specific term. A set of routines that is used to manipulate data files. The OpenVMS high-level file system.

routine A set of computer instructions that perform an operation.

row A row is the horizontal dimension of a table (compare column). A row in its most basic roll equates to a record within a file.

scalar value A scalar value is in principle any value that can be assigned to a column.

search condition A search condition is a statement or group of statements joined together by Boolean operators which results in a true, false or unknown condition.

sequential file organisation (in VAX/OpenVMS terms) A file organisation in which records appear in the order in which they were originally written. The records can be fixed length or variable length.

server A server can be considered as an engine providing some service. A client calls the server, resulting in the service being performed.

set function (SQL) An operation on a set of values in a column of a table or all values from a column in a group of rows in a table.

single-tier driver (also see multi-tier driver) An ODBC implementation in which the data is processed directly by the ODBC driver.

SQL (Structured Query Language). A standard language for interacting with relational database systems.

SQL statement SQL statements can be categorised as data manipulation, data definition or SQL control statements.

syntax The structure of a language, command or statement.

system data source A data source which can be accessed by any user on a given computer compare user data source.

table A table consists of column definitions and is the view a user sees of data definitions they have defined. You can consider a table as a rectangular sheet containing *columns* and *rows*. Each intersection (cell) can contain a value.

TCP/IP (Transmission Control Protocol/Internet Protocol). A standard method (rfc 793) of accessing data on different machines.

UIC (in VAX/OpenVMS terms) The abbreviation for user identification code. A 32 bit value assigned to users and files that specifies the allowed operations. A unique identifier for a user in the system.

upload(ing) In non-technical terms uploading is taking data from the client PC and storing it on the server (compare downloading).

user data source A data source which can be accessed by a specific user only (compare system data source).

User Identification Code see UIC.

VAX (Virtual Address eXtension). A type of computer produced by Digital Equipment Corporation (DEC) which can use the VMS and OpenVMS operating systems.

VMS (Virtual Management System). An operating system that runs on VAX and Alpha machines. Superseded by OpenVMS.

APPENDIX D

Data Types

This appendix consists of 2 sections:

- SQL data types that are supported by Easysoft ODBC
- Local data types supported by the Easysoft SQL Engine on the server

SQL Data Types

This section lists the SQL data types that are supported by Easysoft ODBC.

Character String Data Types

CHAR Character string of fixed string length *n*, where *n* is less than or equal to 254.

VARCHAR Variable length character string with a maximum string length 254.

Exact Numeric Data Types

TINYINT Signed, exact, numeric value with precision 3 and scale 0.

SMALLINT Signed, exact, numeric value with a precision 5 and a scale 0.

INTEGER Signed, exact, numeric value with a precision 10 and a scale 0.

Approximate Numeric Data Types

DOUBLE Signed, approximate, numeric value with a mantissa precision 15 (zero or absolute value 10^{-38} to 10^{38}).

Datetime Data Types

DATE Date data.

TIME Time data.

TIMESTAMP Date/time data.

Local Data Types

The supported data types of the Easysoft SQL Engine on the server are shown in the table below.

Notes

1. **Storage** column heading: this refers to the storage type. There are three basic storage types, namely
   ASCII, denoted by the character A
   Packed, denoted by the character P
   Binary, denoted by the character B.

Binary integer data types are a special case, because there are big-endian and little-endian storage formats to consider. There are four variants, namely

Native, denoted by the character N. This may be big-endian or little-endian.
Reversed native, denoted by the character R
Big-endian, denoted by the character B
Little-endian, denoted by the character L

Native means that the data is stored according to the default defined by the architecture of your machine.

Reversed native means that the bytes are stored in reversed order to the native storage format of your machine.

Big-endian means that the data is always stored in big-endian format, regardless of the native storage format of your machine.

Little-endian means that the data is always stored in little-endian format, regardless of the native storage format of your machine.

For binary integers, the storage sub-type is indicated after the B character. For example, a reversed native binary integer is shown as BR.

Do not confuse the storage type with the SQL data type. For example, the BASIC-DATE data type is stored as a binary integer (BN), but the SQL data type is DATE.

2. **Format** column heading: for dates and times the following symbols are used: Y = Year, M = Month, D = day, h = hour, m = minute, s = second, c = 0.01 second. This column may also contain more general comments on the format.
3. **Length** column heading: if the length is *not* fixed, then the default length is shown first, and the maximum length is shown in brackets. If the length is fixed, then this is indicated by a single value in the column.
4. **Prec** column heading: this is the precision. For a numeric field this refers to the maximum number of digits used by the data type. The precision of a non-numeric field is either the maximum length or defined length of the field.

5. **Scale** column heading: for numeric data types this refers to the maximum number of digits to the right of the decimal point. Scale is not applicable to non-numeric data types.
6. **Start date/time (End date/time)** column heading: for dates and times this refers to the starting date or time. If there is an end date or time, this is shown in brackets following the start date/time.
7. **Sign** column heading: S indicates that the data type is signed, U indicates that it is not signed.
8. (d) after an entry indicates that it is a default value and can be changed.
9. For historical reasons, data types with different names may have the same structure.
10. Additional notes (indicated by a number after the data type name) on a few data types are contained in Table 7.

Table 6. Local Data Types

Name	Storage	SQL data type (default)	Length	Prec	Scale	Format	Start date/time (End date/time)	Interval	Sign
ABC-PACKED [Note 1]	P	DOUBLE	8 (16)	16 (d)	0 (d)	Trailing sign	-	-	S
ABC-STRING	A	VARCHAR	50 (254)	50 (d)	0	-	-	-	U
AMOS-F6	BN	DOUBLE	6	10	0 (d)	-	-	-	U
BASIC-DATE	BN	DATE	2	5	-	DDDYY	01-JAN-1970	1 day	S
BASIC-DATE4	BN	DATE	4	10	-	DDDYY	01-JAN-1970	1 day	S
BD-DEC	P	DOUBLE	8 (16)	8 (d)	0 (d)	-	-	-	S
BDC-OVERPUNCHED-NUMERIC	A	VARCHAR	16 (32)	16 (d)	0 (d)	-	-	-	S
BINDATE	BN	DATE	4	10	-	YYYYMMDD (d)	-	-	S
BINDATE-REVERSED	BR	DATE	4	10	-	YYYYMMDD (d)	-	-	S
BINTIME	BN	TIME	4	10	-	hhmmsscc (d)	-	-	S
BINTIME-2-U	BN	TIME	2	5	-	hhmm (d)	-	-	U
BINTIME-U	BN	TIME	4	10	-	hhmmsscc (d)	-	-	U
BIT-1	BN	TINYINT	1	1	0 (d)	-	-	-	U
BIT-2	BN	TINYINT	1	1	0 (d)	-	-	-	U
BIT-3	BN	TINYINT	1	1	0 (d)	-	-	-	U
BIT-4	BN	TINYINT	1	1	0 (d)	-	-	-	U
BIT-5	BN	TINYINT	1	1	0 (d)	-	-	-	U
BIT-6	BN	TINYINT	1	1	0 (d)	-	-	-	U
BIT-7	BN	TINYINT	1	1	0 (d)	-	-	-	U
BIT-8	BN	TINYINT	1	1	0 (d)	-	-	-	U
BPDATE	BN	DATE	4 (20)	15 (d)	-	YYYY-	31-12-1799- 00:00:00.00	1 day	U
BPDOUBLE	A	DOUBLE	8 (32)	15 (d)	0 (d)	-	-	-	S

Table continued on next page

Local data types (continued)

BYTE	BN	TINYINT	1	3	0	-	-	-	S
C-TIME	BN	TIMESTAMP	4	10	-	Count of seconds since Start	01-JAN-1970 00:00:00.00	1 second	S
C-TIME-REV	BR	TIMESTAMP	4	10	-	Count of seconds since Start	01-JAN-1970 00:00:00.00	1 second	S
C-TIME2	BN	TIMESTAMP	4	10	-	Count of seconds since Start	01-JAN-1900 00:00:00.00	1 second	S
CDS-DATE	BN	DATE	2	5	-	366 days in a year	01-JAN-1900	1 day	S
CDS-DATE2	BN	DATE	2	5	-	366 days in a year	24-OCT-1924	1 day	S
CDS-TIME3	B	TIME	3	9	-	Hhmmss	-	-	S
CODA-DATE2	BN	TIMESTAMP	4	10	-	Count of minutes	18-NOV-1858 00:00:00.00	1 minute	S
D-FLOATING	B	DOUBLE	8	15	0 (d)	-	-	-	S
DATE	A	DATE	11 (254)	11 (d)	-	DD-MMM-YYYY Upper case characters only		-	-
DATE-LOWER-CASE	A	DATE	11 (254)	11 (d)	-	DD-MMM-YYYY Lower-case characters only	-	-	-
DATE-MIXED-CASE	A	DATE	11 (254)	11 (d)	-	DD-MMM-YYYY Mixed-case characters, first character must be upper-case	-	-	-
DATE-PAD-ZERO [Note 2]	A	DATE	8 (254)	8 (d)	-	DDMMYYYY	-	-	-
DIAL-199-DATE	B	DATE	3	9	-	YYMMDD Special format with Year-2000 conformance	00-00-1900 (31-12-2155)	-	-
DIAL-99-DATE	B	DATE	3	9	-	YYMMDD Special format with Year-2000 conformance	00-00-1900 (31-12-2155)	-	-
DIAL-DATE3	B	DATE	3	9	-	YYMMDD Special format with Year-2000 conformance	00-00-1900 31-12-2155	-	-
DIAL-DATE6	A	DATE	6	6	-	XYMMDD Special format with Year-2000 conformance. X = 1 to 9, A to Z.	- (31-12-2250)	-	-
DIBOL-DATE-5	A	DATE	5	5	-	-	31-12-1900	1 day	-
DIBOL-OVERPUNCHED-NUMERIC	A	DOUBLE	16 (19)	16 (d)	0 (d)	Overpunched trailing sign	-	-	S
DIBOL-TIME-5	A	TIME	5	5	0		00:00:00.00	1 second	U
DOUBLE	B	DOUBLE	8	15	0 (d)	-	-	-	S
DOUBLE-REVERSED	BR	DOUBLE	8	15	0 (d)	-	-	-	S
DOUBLE_SPACES_NULL	B	DOUBLE	8	15	0 (d)	-	-	-	S
F-FLOATING	B	DOUBLE	4	7	0 (d)	-	-	-	S
FLOAT	B	DOUBLE	4	7	0 (d)	-	-	-	S
FLOAT-REVERSED	BR	DOUBLE	4	7	0 (d)	-	-	-	S
FLOATING –DATE-4	B	DATE	4	15	0	-	-	-	S
FLOATING –DATE-8	B	DATE	8	15	0 (d)	-	-	-	S
FLOATING-JULIAN	A	DATE	8	15	0 (d)	-	31-12-1799	1 day	S
FMBINDATE	B	DATE	4	10	0	-YYYYMMDD	-	-	S
FREEFORM	A	VARCHAR	16 (32)	16 (d)	-	-	-	-	-
FUG-DATE	B	DATE	3	10	0	YYMMDD	00-00-1800 00:00:00.00 (31-12-2055)	-	-

Table continued on next page

Local data types (continued)

G-FLOATING	B	DOUBLE	8	15	0 (d)	-	-	-	S
G-FLOATING-REV	BR	DOUBLE	8	15	0 (d)	-	-	-	S
H-FLOATING	B	DOUBLE	16	33	0 (d)	-	-	-	S
HEXADECIMAL	B	VARCHAR	10 (127)	20 (d)	0	-	-	-	-
HEXADECIMAL-REVERSED	BR	VARCHAR	10 (127)	20 (d)	0	-	-	-	-
HEXDATE	B	DATE	4	10	0	YYYYMMDD	-	-	S
HEXDATE-REVERSED	BR	DATE	4	10	0	YYYYMMDD	-	-	S
HEXDATE3	B	DATE	3	10	0	YYMMDD	00-00-1900	-	S
HEXDATE3-REVERSED	BR	DATE	3	10	0	YYMMDD	00-00-1900	-	S
IBASEDATE [Note 3]	BN	TIMESTAMP	8	19	-	-	17-NOV-1858 00:00:00.00	1×10^{-7} sec.	S
ICC-INT1	B	SMALLINT	1	3	0 (d)	-	-	-	S
ICC-INT2	B	SMALLINT	2	5	0 (d)	-	-	-	S
ICC-INT3	B	INTEGER	3	8	0 (d)	-	-	-	S
ICC-INT4	B	INTEGER	4	1	0 (d)	-	-	-	S
INFO-DATE	A	DATE	254	-	-	-	-	-	-
INFO-L-TYPE	A	DOUBLE	16 (32)	16 (d)	0 (d)	Leading sign	-	-	S
INFO-N-TYPE	A	DOUBLE	16 (32)	16 (d)	0 (d)	Explicit decimal point	-	-	S
INFO-T-TYPE	A	DOUBLE	16 (32)	16 (d)	0 (d)	Trailing sign	-	-	S
INFO-V-TYPE	A	DOUBLE	16 (32)	16 (d)	0 (d)	Leading sign, implicit decimal point	-	-	S
INFORMIX-DATE-4	BN	DATE	4	10	-	-	31-DEC1899	1 day	-
INFORMIX-DATE-4-REV	BR	DATE	4	10	-	-	31-DEC1899	1 day	-
INFORMIX- INTEGER	B	INTEGER	4	10	0 (d)	-	-	-	S
INFORMIX- INTEGER-REVERSED	BR	INTEGER	4	10	0 (d)	-	-	-	S
INFORMIX-SMALLINT	B	SMALLINT	2	5	0 (d)	-	-	-	S
INFORMIX-SMALLINT-REVERSED	BR	SMALLINT	2	5	0 (d)	-	-	-	S
INFORMIX-STRING	A	VARCHAR	50 (254)	50	-	-	-	-	-
INTEGER-S	BN	INTEGER	4	10	0 (d)	-	-	-	S
INTEGER1	BL	SMALLINT	1	3	0 (d)	-	-	-	S
INTEGER2	BL	SMALLINT	2	5	0 (d)	-	-	-	S
INTEGER3	BL	INTEGER	3	7	0 (d)	-	-	-	S
INTEGER4	BL	INTEGER	4	10	0 (d)	-	-	-	S
INTEGER5	BL	DOUBLE	5	12	0 (d)	-	-	-	S
INTEGER6	BL	DOUBLE	6	15	0 (d)	-	-	-	S
INTEGER7	BL	DOUBLE	7	17	0 (d)	-	-	-	S
INTEGER8	BL	DOUBLE	8	19	0 (d)	-	-	-	S
JDATE	BN	DATE	2	5	-	-	01-JAN-1900	-	-
KD-DATE [Note 4]	A	DATE	7	7	-	YYYYDDD	-	-	-
LDA-LEADING-SIGN-NUMERIC	A	DOUBLE	16 (32)	16 (d)	0 (d)	-	-	-	S
LDA-NUMERIC	A	DOUBLE	16 (32)	16 (d)	0 (d)	-	-	-	S
LEADING-OVERPUNCHED-NUMERIC	A	VARCHAR	16 (32)	16 (d)	-	Overpunched leading sign	-	-	S
LEADING-SIGN-NUMERIC	A	VARCHAR	16 (32)	16 (d)	-	Leading sign	-	-	S

Table continued on next page

Local data types (continued)

LONG	BN	INTEGER	4	10	0 (d)	-	-	-	S
LONGWORD	BN	INTEGER	4	10	0 (d)	-	-	-	S
LONGWORD-REVERSED	BR	INTEGER	4	10	0 (d)	-	-	-	S
MAGIC-DATE	P	DATE	4 (16)	8 (d)	0		00:00:00.00	-	-
MAGIC-INTEGER-DATE	B	DATE	4	10	-	Count of days since Start	01-01-0000	1 day	-
MAGIC-INTEGER-DATE-REVERSED	BR	DATE	4	10	-	Count of days since Start	01-01-0000	1 day	-
MAGIC-INTEGER-TIME	B	TIME	4	10	-	-	01-01-1970 00:00:00.00	1 second	-
MAGIC-INTEGER-TIME-REVERSED	BR	TIME	4	10	-	-	01-01-1970 00:00:00.00	1 second	-
MAGIC-NUMBER	P	DOUBLE	8 (16)	8 (d)	0 (d)	-	-	-	S
MAGIC-STRING-MEMO	A	VARCHAR	50 (254)	48 (d)	-	First two bytes contain length	-	-	-
MAGIC-TIME	P	TIME	4 (16)	8 (d)	0	-	00:00:00.00		-
MICROFOCUS-COMP-5	B	DOUBLE	8	19	0 (d)	-			S
MOTOROLA-LONGWORD	B	INTEGER	4	10	0 (d)	-	-	-	S
MOTOROLA-LONGWORD –REVERSED	B	INTEGER	4	10	0 (d)	First two bytes contain length	-	-	S
NULL-TERMINATED-STRING	A	VARCHAR	59 (254)	50 (d)	-	-	-	-	-
NUMERIC	A	VARCHAR	16 (32)	16 (d)	-	-	-	-	-
OPENVMS-GREEK-STRING	A	VARCHAR	50 (254)	50 (d)	0		-	-	-
OPENVMS-SLOVENIAN-STRING	A	VARCHAR	50 (254)	50 (d)	0		-	-	-
OVERPUNCHED-DATE-1	A	DATE	6	6	0	Trailing sign	31-12-1799 00:00:00.00	1 day	S
PACKED	P	DOUBLE	8 (16)	16 (d)	0 (d)	Trailing sign	-	-	S
PACKED-S	P	DOUBLE	8 (16)	16 (d)	0 (d)	Trailing sign	-	-	S
PACKED-U	P	DOUBLE	8 (16)	16 (d)	0 (d)	-	-	-	U
PHDATE [Note 5]	BN	DATE	2	5	-	-	-	-	-
PHDATE4 [Note 6]	BN	DATE	4	10	-	-	-	-	-
PHDATETIME	B	TIMESTAMP	8	19	-	YYYYMMDD hhmmsscc	-	-	U
PSI-DATE [Note 7]	P	DATE	4	7	-	YYYYDDD Trailing sign	-	-	S
QUADWORD	BN	DOUBLE	8	19	0 (d)	-	-	-	S
QUADWORD-REVERSED	BR	DOUBLE	8	19	0 (d)	-	-	-	S
RFA	B	STRING	6	15	-	-	-	-	U
RMC-ASCDATE-DDMMYY	A	DATE	6	-	-	DDMMYY	-	-	-
RMC-ASCDATE-YYMMDD	A	DATE	6	-	-	YYMMDD	-	-	-
RMC-BINDATE-DDMMYY	B	DATE	6	-	-	DDMMYY	-	-	-
RMC-BINDATE-YYMMDD	B	DATE	6	-	-	YYMMDD	-	-	-
SAIC-TIMESTAMP	B	TIMESTAMP	4	10	0	-	01-01-1980 00:00:00.00	-	S
SEB-DATE	B	DATE	6	15	-	YYYYMMDD	-	-	-
SINT3	BN	INTEGER	3	7	0 (d)	-	-	-	S

Table continued on next page

Local data types (continued)

SINT5	BN	DOUBLE	5	12	0 (d)	-	-	-	S
SINT6	BN	DOUBLE	6	15	0 (d)	-	-	-	S
SINT7	BN	DOUBLE	7	17	0 (d)	-	-	-	S
SIRSI-MONEY	B	DOUBLE	8	14	0	-	-	-	S
SMITHSONIAN-DATE	BN	DATE	4	10	-	Count of days since Start	17-NOV-1825	1 day	-
SOCIALE-DATE	B	DATE	4	6	-	-	-	-	-
SOCIALE-DATE-REV	B	DATE	4	6	-	-	-	-	-
SOCIALE–INT1	B	INTEGER	1	2	0 (d)	-	-	-	S
SOCIALE–INT2	B	INTEGER	2	4	0 (d)	-	-	-	S
SOCIALE–INT3	B	INTEGER	3	6	0 (d)	-	-	-	S
SOCIALE–INT4	B	INTEGER	4	8	0 (d)	-	-	-	S
SOCIALE–INT5	B	INTEGER	5	10	0 (d)	-	-	-	S
SOCIALE–INT6	B	DOUBLE	6	16	0 (d)	-	-	-	S
SOCIALE–INT7	B	DOUBLE	3	16	0 (d)	-	-	-	S
SP3-DATE	P	DATE	3	5	-	YYDDD Trailing sign	-	-	S
SPD-DATE	P	DATE	16	15	-	YYYYMMDD (d) Trailing sign	-	-	S
SPD-DATE-5	P	DATE	5	8	-	YYYYMMDD (d) Trailing sign	-	-	S
SPECIAL-1P	A	VARCHAR	4 (4)	4	0	-	-	-	-
SPECIAL-2P	A	VARCHAR	4 (4)	4	0	-	-	-	-
SPECIAL-3P	A	VARCHAR	6 (6)	6	0	-	-	-	-
SPECIAL-4P	A	VARCHAR	4 (4)	4	0	-	-	-	-
SPECIAL-5P	A	VARCHAR	4 (4)	4	0	-	-	-	-
SPECIAL-6P	A	VARCHAR	6 (6)	6	0	-	-	-	-
SPECIAL-DATE-1	BN	DATE	2	5	-	-	05-AUG-1948	1 day	S
SPECIAL-DATE-10	A	DATE	5 (254)	0 (d)	0	-	31-12-1799 00:00:00.00	1 day	-
SPECIAL-DATE-11	BR	DATE	2	5	0	-	01-01-1970	1 year	U
SPECIAL-DATE-12	P	DATE	5	8	0	DDMMYYYY Trailing sign	-	-	U
SPECIAL-DATE-13	B	DATE	6	8	0	YYYYDDMM	0-0-0000 00:00:00.00	-	-
SPECIAL-DATE-14	BR	DATE	4	10	0	-	00:00:00.00	1 day	-
SPECIAL-DATE-15	B	TIME	4	7	0	-	-	-	S
SPECIAL-DATE-16	B	DATE	3	5	0	-	31-12-1599 00:00:00.00	1 day	-
SPECIAL-DATE-17	A	DATE	6 (6)	6	0	PPMMDD	00:00:00.00	-	-
SPECIAL-DATE-18	B	DATE	2	5	0	-	31-12-1969	1 day	S
SPECIAL-DATE-19	P	DATE	4	6	0	YYMMDD Trailing sign	-	-	S
SPECIAL-DATE-2	BL	DATE	2	5	-	-	31-DEC-1919	1 day	-
SPECIAL-DATE-20	P	DATE	3	5	0	HHMM Trailing sign	-	-	U
SPECIAL-DATE-21	B	DATE	2	5	0	-	01-01-1970	1 year	U
SPECIAL-DATE-3	BN	DATE	5	5	0	-	30-DEC-1899	1 day	-
SPECIAL-DATE-4	A	DATE	5 (5)	5	-	YYDDD	00:00:00.00	-	-
SPECIAL-DATE-5	B	TIMESTAMP	4	10	0	-	01-01-1984 00:00:00.00	1 second	-

Table continued on next page

Local data types (continued)

SPECIAL-DATE-6	A	DATE	5	5	0	-	31-12-1900 00:00:00.00	1 day	-
SPECIAL-DATE-7	B	DATE	2	5	0	-	01-01-1970	1 year	U
SPECIAL-DATE-8	B	DATE	2	5	0	-	01-01-1970	1 day	S
SPECIAL-DATE-9	A	DATE	3 (3)	3	0	DDD	00:00:00.00	-	-
STAPLES-DATE	A	DATE	6	6	0	-	31-12-1899 00:00:00.00	1 day	-
STARDATE	BN	TIMESTAMP	8	20	-	-	17-NOV-1858 (09-DEC-5941)	1×10^{-7} sec.	-
STRING	A	VARCHAR	50 (254)	50 (d)	-	-	-	-	-
TETRA-DATE	A	DATE	-	-	-	-	-	-	-
TETRA-JULIAN-DATE	A	DATE	6	6	0	-	30-12-1899 00:00:00.00	1 day	-
TIME	A	TIME	11 (254)	11 (d)	-	hh:mm:ss.cc (d)	-	-	-
TIMESTAMP	BN	TIMESTAMP	8	20	-	-	17-NOV-1858 (31-DEC-9999)	1×10^{-7} sec.	-
TRAILING-NEGSIGN-DECIMAL	A	DOUBLE	16 (23)	16 (d)	0 (d)	Trailing sign	-	-	S
TRAILING-OVERPUNCHED-NUMERIC	A	VARCHAR	16 (32)	16 (d)	-	Overpunched trailing sign	-	-	S
TRAILING-SIGN-NUMERIC	A	VARCHAR	16 (32)	16 (d)	-	Trailing sign	-	-	S
TRESTLE-DATE	B	DATE	2	5	0	DDMMYY	00-00-1980 00:00:00.00	-	-
TRESTLE-TIME	B	TIME	3	9	0	hhmmss	00:00:00.00	-	-
UINT3	BN	INTEGER	3	8	0 (d)	-	-	-	U
UINT5	BN	DOUBLE	5	13	0 (d)	-	-	-	U
UINT6	BN	DOUBLE	6	15	0 (d)	-	-	-	U
UINT7	BN	DOUBLE	7	17	0 (d)	-	-	-	U
UINTEGER1	BL	SMALLINT	1	3	0 (d)	-	-	-	U
UINTEGER2	BL	INTEGER	2	5	0 (d)	-	-	-	U
UINTEGER3	BL	INTEGER	3	8	0 (d)	-	-	-	U
ULONG	BN	DOUBLE	4	10	0 (d)	-	-	-	U
ULONGWORD-REVERSED	BR	DOUBLE	4	10	0 (d)	-	-	-	U
UP3-DATE	P	DATE	3	5	-	YYDDD (D)	-	-	U
UQUADWORD	BN	DOUBLE	8	20	0 (d)	-	-	-	U
UQUADWORD-REVERSED	BR	DOUBLE	8	20	0 (d)	-	-	-	U
UWORD	BN	INTEGER	2	5	0 (d)	-	-	-	U
UWORD-REVERSED	BR	INTEGER	2	5	0 (d)	-	-	-	U
VMSDATE4	B	DATE	4	10	-	-	17-NOV-1858 (31-DEC-9999)	1×10^{-7} sec.	-
VMSDATE4-REVERSED	B	DATE	4	10	-	-	17-NOV-1858 (31-DEC-9999)	1×10^{-7} sec.	-
VMSDATE8	B	DATE	8	20	-	-	17-NOV-1858 (31-DEC-9999)	1×10^{-7} sec.	-
VMSDATE8-REVERSED	B	DATE	8	20	-	-	17-NOV-1858 (31-DEC-9999)	1×10^{-7} sec.	-
WORD	BN	SMALLINT	2	5	0 (d)	-	-	-	S
WORD-REVERSED	BR	SMALLINT	2	5	0 (d)	-	-	-	S
ZONED	A	VARCHAR	16 (32)	16 (d)	-	-	-	-	U
ZONED-NUM	A	VARCHAR	16 (32)	16 (d)	-	Trailing sign	-	-	S
ZONED-S	A	VARCHAR	16 (32)	16 (d)	-	Trailing sign	-	-	S
ZONED-U	A	VARCHAR	16 (32)	16 (d)	-	-	-	-	U

Table 7. Notes on Local Data Types

Note number	Data type	Comment
1	ABC-PACKED	Sign is stored in last nibble (4 bits) of packed string. Positive sign represented by 3, negative sign represented by 5
2	DATE-PAD-ZERO	ASCII date, padded with zero values (hexadecimal 0x30)
3	IBASEDATE	Begin date: 01-JAN-0100 00:00:00.00, End date: 09-DEC-5941 00:00:00.00
4	KD-DATE	DDD is number of days since start of year. Example: 20th February 1996 is stored as 1996051 (051= 31 days Jan, + 20 days Feb).
5	PHDATE	2 byte storage, first 7 bits for year, next four for month, last five for day.
6	PHDATE4	4 byte storage, first 23 bits for year, next four for month, last five for day.
7	PSI-DATE	DDD is number of days since start of year. Example: 20th February 1996 is stored as 1996051 (051= 31 days Jan, + 20 days Feb).

APPENDIX E

Easysoft Catalog

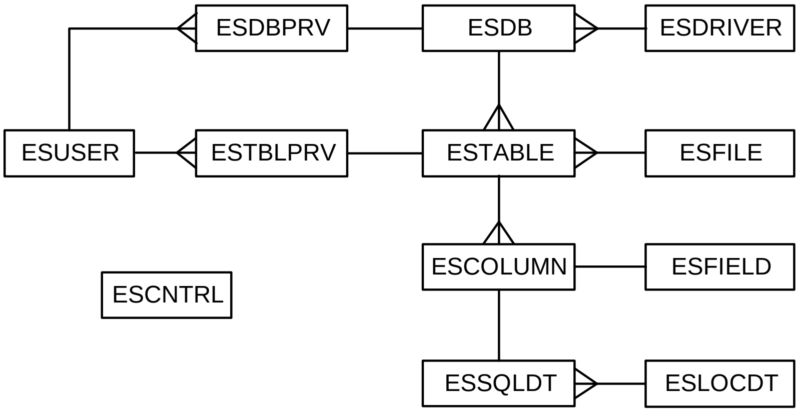


Figure 88. **Easysoft Catalog**

Figure 88 shows the entity relationships of the Easysoft Catalog. A detailed breakdown of each table follows.

For each of the tables in the Easysoft Catalog the corresponding file name, record type and record size is indicated. Following this, the columns in the table are listed along with the data type and length of that column.

Catalog Tables

ESCNTRL

Filename	ESCNTRL.DAT
Record Type	FIXED
Record Size	346

Column	Data Type	Length
PARAM_NAME	STRING	16
PARAM_VALUE	STRING	254
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESCOLUMN

Filename	ESCOLUMN.DAT
Record Type	FIXED
Record Size	920

Column	Data Type	Length
DATABASE_NAME	STRING	10
TABLE_NAME	STRING	64
COLUMN_NUMBER	WORD	2
COLUMN_NAME	STRING	64
FIELD_NAME	STRING	64
SQL_TYPE_NAME	STRING	64
LENGTH	LONG	4
PRECISION	LONG	4
SCALE	WORD	2
NULLABLE	WORD	2
UNSIGNED_ATTRIBUTE	WORD	2
UPDATABLE	WORD	2
VISIBLE	WORD	2
MONEY	WORD	2
AUTO_INCREMENT	WORD	2
CASE_SENSITIVE	WORD	2
SEARCHABLE	WORD	2
RADIX	WORD	2
DEFAULT_VALUE	STRING	32
REMARKS	STRING	254
COLUMN_TYPE	WORD	2
DISPLAY_NAME	STRING	64
MODIFIED BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESDB

Filename	ESDB.DAT
Record Type	FIXED
Record Size	914

Column	Data Type	Length
DATABASE_NAME	STRING	10
DEFAULT_DIRECTORY	STRING	128
DRIVER_NAME	STRING	64
SETUP_PROCEDURE	STRING	128
CONNECT_STRING	STRING	254
REMARKS	STRING	254
DISPLAY_NAME	STRING	10
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESDBPRV

Filename	ESDBPRV.DAT
Record Type	FIXED
Record Size	492

Column	Data Type	Length
USER_NAME	STRING	64
DATABASE_NAME	STRING	10
DATABASE_USERNAME	STRING	32
DATABASE_PASSWORD	STRING	32
ALLOW_SELECT	WORD	2
ALLOW_INSERT	WORD	2
ALLOW_UPDATE	WORD	2
ALLOW_DELETE	WORD	2
SECURITY_LEVEL	STRING	16
REMARKS	STRING	254
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESDRIVER

Filename	ESDRIVER.DAT
Record Type	FIXED
Record Size	522

Column	Data Type	Length
DRIVER_NAME	STRING	64
FILE_SPECIFICATION	STRING	128
REMARKS	STRING	254
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESFIELD

Filename	ESFIELD.DAT
Record Type	FIXED
Record Size	826

Column	Data Type	Length
FILE_NAME	STRING	64
FIELD_NUMBER	WORD	2
FIELD_NAME	STRING	64
LOCAL_TYPE_NAME	STRING	64
OFFSET	LONG	4
LENGTH	LONG	4
PRECISION	LONG	4
SCALE	WORD	2
ENCRYPTED	WORD	2
FORMAT	STRING	254
DEFAULT_VALUE	STRING	32
REMARKS	STRING	254
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESFILE

Filename	ESFILE.DAT
Record Type	FIXED
Record Size	684

Column	Data Type	Length
FILE_NAME	STRING	64
OPEN_STRING	STRING	254
RECORD_SIZE	LONG	4
FILE_ORGANISATION	STRING	16
RECORD_TYPE	STRING	16
REMARKS	STRING	254
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESLOCDT

Filename	ESLOCDT.DAT
Record Type	FIXED
Record Size	712

Column	Data Type	Length
LOCAL_TYPE_NAME	STRING	64
SQL_TYPE_NAME	STRING	64
ATTRIBUTES	STRING	254
REMARKS	STRING	254
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESSQLDT

Filename	ESSQLDT.DAT
Record Type	FIXED
Record Size	530

Column	Data Type	Length
SQL_TYPE_NAME	STRING	64
SQL_DATA_TYPE	WORD	2
LENGTH	LONG	4
PRECISION	LONG	4
LITERAL_PREFIX	STRING	16
LITERAL_SUFFIX	STRING	16
CREATE_PARAMS	STRING	16
NULLABLE	WORD	2
CASE_SENSITIVE	WORD	2
SEARCHABLE	WORD	2
UNSIGNED_ATTRIBUTE	WORD	2
MONEY	WORD	2
AUTO_INCREMENT	WORD	2
RADIX	WORD	2
LOCAL_TYPE_NAME	STRING	64
REMARKS	STRING	254
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESTABLE

File name	ESTABLE.DAT
Record Type	FIXED
Record Size	976

Column	Data Type	Length
DATABASE_NAME	STRING	10
TABLE_NAME	STRING	64
TABLE_TYPE	STRING	64
FILE_NAME	STRING	64
FILE_SPECIFICATION	STRING	128
NUMBER_COLUMNS	WORD	2
CRITERIA	STRING	254
REMARKS	STRING	254
DISPLAY_NAME	STRING	64
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESTBLPRV

Filename	ESTBLPRV.DAT
Record Type	FIXED
Record Size	476

Column	Data Type	Length
DATABASE_NAME	STRING	64
TABLE_NAME	STRING	64
USER_NAME	STRING	64
ALLOW_SELECT	WORD	2
ALLOW_INSERT	WORD	2
ALLOW_UPDATE	WORD	2
ALLOW_DELETE	WORD	2
REMARKS	STRING	254
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

ESUSER

Filename	ESUSER.DAT
Record Type	FIXED
Record Size	458

Column	Data Type	Length
USER_NAME	STRING	64
USER_PASSWORD	STRING	64
REMARKS	STRING	254
MODIFIED_BY	STRING	64
MODIFIED_ON	TIMESTAMP	8

Note: by default, the MODIFIED_BY and MODIFIED_ON columns are not visible.

APPENDIX F

Supported SQL

SQL Statements

The following SQL statements are supported by the Easysoft software. Words enclosed in angle brackets (<>) should be replaced by any valid SQL syntax. For a detailed description of valid SQL syntax, refer to, for example, the Microsoft ODBC Programmer's Reference (see Bibliography).

INSERT

```
INSERT INTO <table> [( <column identifiers>)]  VALUES (<values>)
```

SELECT

```
SELECT [ALL | DISTINCT] <select-list>
FROM <table>
[WHERE <search condition>]
[GROUP BY <column names>]
[HAVING <search condition>]
[ORDER BY <sort specification>]
```

DELETE SEARCHED

```
DELETE FROM <table> [WHERE <search condition>]
```

UPDATE SEARCHED

```
UPDATE <table>
SET <column identifiers> = <expression>
[WHERE <search condition>]
```

Supported Set Functions

COUNT(*), COUNT, MAX, MIN, SUM, AVG

SQL Reserved Words

Reserved words are words that have a special significance for SQL. A user may not use reserved words in writing SQL commands except within character literals. This section provides a list of the current list of reserved words in the Easysoft SQL Engine.

ABSOLUTE
ACTION
ADA
ADD
AFTER
ALIAS
ALL
ALLOCATE
ALTER
AND
ANY
ARE
AS
ASC
ASSERTION
ASYNC
AT
AUTHORIZATION
AVG
BEFORE
BEGIN
BETWEEN
BIT
BIT_LENGTH
BOOLEAN
BOTH
BREADTH
BY
C
CALL
CASCADE
CASCADED
CASE
CAST
CATALOG
CATALOG_NAME
CHAR
CHAR_LENGTH
CHARACTER
CHARACTER_LENGTH
CHARACTER_SET_NAME
CHARACTER_SET_SCHEMA
CHECK
CLASS_ORIGIN
CLOSE
COALESCE
COBOL
COLLATE
COLLATION
COLLATION_CATALOG
COLLATION_NAME
COLLATION_SCHEMA
COLUMN
COLUMN_NAME
COMMAND_FUNCTION
COMMIT
COMMITTED
COMPLETION
CONDITION_NUMBER
CONNECT
CONNECTION
CONNECTION_NAME
CONSTRAINT
CONSTRAINT_CATALOG
CONSTRAINT_NAME
CONSTRAINT_SCHEMA
CONSTRAINTS
CONTINUE
CONVERT
CORRESPONDING
COUNT
CREATE
CROSS
CURRENT
CURRENT_DATE
CURRENT_TIME
CURRENT_TIMESTAMP
CURRENT_USER
CURSOR
CURSOR_NAME
CYCLE
DATA
DATE
DATETIME_INTERVAL_CODE
DATETIME_INTERVAL_PRECISION
DAY
DEALLOCATE
DEC
DECIMAL
DECLARE
DEFAULT
DEFERRABLE
DEFERRED
DELETE
DEPTH
DESC
DESCRIBE
DESCRIPTOR
DIAGNOSTICS
DICTIONARY
DISCONNECT
DISTINCT
DOMAIN
DOUBLE
DROP
DYNAMIC_FUNCTION
EACH
ELSE
ELSEIF
END
END-EXEC
EQUALS
ESCAPE
EXCEPT
EXCEPTION
EXEC
EXECUTE
EXISTS
EXTERNAL
EXTRACT
FALSE
FETCH
FIRST
FLOAT
FOR
FOREIGN
FORTRAN
FOUND
FROM
FULL
GENERAL
GET
GLOBAL
GO
GOTO
GRANT
GROUP
HAVING

HOUR
IDENTITY
IF
IGNORE
IMMEDIATE
IN
INCLUDE
INDEX
INDICATOR
INITIALLY
INNER
INPUT
INSENSITIVE
INSERT
INT
INTEGER
INTERSECT
INTERVAL
INTO
IS
ISOLATION
JOIN
KEY
LANGUAGE
LAST
LEADING
LEAVE
LEFT
LENGTH
LESS
LEVEL
LIKE
LIMIT
LOCAL
LOOP
LOWER
MATCH
MAX
MESSAGE_LENGTH
MESSAGE_OCTET_LENGTH
MESSAGE_TEXT
MIN
MINUTE
MODIFY
MODULE
MONTH
MORE
MUMPS
NAME
NAMES
NATIONAL
NATURAL
NCHAR
NEW
NEXT
NO
NONE
NOT
NOTRACEPOINT
NULL
NULLABLE
NULLIF
NUMBER
NUMERIC
OBJECT
OCTET_LENGTH
OF
OFF
OID
OLD
ON
ONLY
OPEN
OPERATION
OPERATORS
OPTION
OR
ORDER
OTHERS
OUTER
OUTPUT
OVERLAPS
PAD
PARAMETERS
PARTIAL
PASCAL
PENDANT
PLI
POSITION
PRECISION
PREORDER
PREPARE
PRESERVE
PRIMARY
PRIOR
PRIVATE
PRIVILEGES
PROCEDURE
PROTECTED
PUBLIC
READ
REAL
RECURSIVE
REF
REFERENCES
REFERENCING
RELATIVE
REPEATABLE
REPLACE
RESIGNAL
RESTRICT
RETURN
RETURNED_LENGTH
RETURNED_OCTET_LENGTH
RETURNED_SQLSTATE
RETURNS
REVOKE
RIGHT
ROLE
ROLLBACK
ROUTINE
ROW
ROW_COUNT
ROWS
SAVEPOINT
SCALE
SCHEMA
SCHEMA_NAME
SCROLL
SEARCH
SECOND
SECTION
SELECT
SENSITIVE
SEQUENCE
SERIALIZABLE
SERVER_NAME
SESSION
SESSION_USER
SET
SIGNAL
SIMILAR
SIZE
SMALLINT
SOME
SPACE
SQL
SQLCA
SQLCODE
SQLERROR
SQLEXCEPTION
SQLSTATE
SQLWARNING
STRUCTURE
SUBCLASS_ORIGIN
SUBSTRING
SUM
SYSTEM
SYSTEM_USER
TABLE
TABLE_NAME
TEMPORARY
TEST
THEN
THERE

TIME
TIMESTAMP
TIMEZONE_HOUR
TIMEZONE_MINUTE
TO
TRACEPOINT
TRAILING
TRANSACTION
TRANSLATE
TRANSLATION
TRIGGER
TRIM
TRUE
TYPE
UNCOMMITTED
UNDER
UNION
UNIQUE
UNKNOWN
UNNAMED
UPDATE
UPPER
USAGE
USER
USING
VALUE
VALUES
VARCHAR
VARIABLE
VARYING
VIEW
VIRTUAL
VISIBLE
WAIT
WHEN
WHENEVER
WHERE
WHILE
WITH
WITHOUT
WORK
WRITE
YEAR
ZONE

APPENDIX G

Import Export Formats

The following import/export formats are supported by RMS.

- Comma Separated Values (CSV, also known as CST)
- Fixed Width Text

Comma Separated Values

Angle brackets (<>) indicate that some value is substituted for the text shown inside the braces; the brackets are not part of the file definition.

File Definition

"FILE","<File Name>" , "<File Organisation>" , "<Record Type>" , <Record Size>

Field Definition

"FIELD" , "<File Name>" , "<Field Name>" , "<Data Type>" , <Offset> , <Length> , <Precision> , <Scale> , <Encrypted> , *"<Date Format>" , "<Default>"

or

"FIELD" , "<File Name>" , "<Field Name>" , "<Data Type>" , <Offset> , <Length>

(Other details are defaulted by the import routine).

* indicates only used for ASCII Dates/Times.

Database Definition

"DB" , "<Database Name>" , "<Default Directory>" , "<Driver Name>", "<Connect String>"

Table Definition

"TABLE" , "<Database Name>" , "<Table Name>" , "<Table Type>" , "<File Name>" , "<File Specification>" , "<Criteria>"

Column Definition

"COLUMN" , "<Database Name>" , "<Table Name>" , "<Column Name>" , "<Field Name"> , "<SQL Data Type>" , <Length> , <Updatable> , <Visible> , "<Default Value>"

User Definition

"USER" , "<User Name>" , "<Password>"

Note: The password field is not written on exports

Database Privileges Definition

"DBPRV" , "<User Name>" , "<Database Name>" , "<Database Username>" , "<Database Password>" , <Allow Select> , <Allow Insert>, <Allow Update> , <Allow Delete> , "<Security Level>"

Table Privileges Definition

"TBLPRV" , "<Database Username>" , "<Database Name>" , "<Database Password>" , "<Table Name>" , <Allow Select> , <Allow Insert> , <Allow Update> , <Allow Delete>

CSV Example

This example shows the CSV for the Easysoft demonstration data.

```
"FILE","LASER","INDEXED","FIXED",74
"FILE","PRODUCT","INDEXED","FIXED",50
"FILE","SUPPLIER","INDEXED","FIXED",77
"FILE","SUPPLIER_PRODUCT","INDEXED","FIXED",4
"FIELD","LASER","CATALOG_NUMBER","LONG",0,4,10,0,0,"",""
"FIELD","LASER","DISC_NAME","STRING",4,55,55,0,0,"",""
"FIELD","LASER","RETAIL_PRICE","DOUBLE",59,8,15,0,0,"",""
"FIELD","LASER","RATING","STRING",67,4,4,0,0,"",""
"FIELD","LASER","STUDIO","STRING",71,3,3,0,0,"",""
"FIELD","PRODUCT","PRODUCT","WORD",0,2,5,0,0,"",""
"FIELD","PRODUCT","PRODUCT_NAME","STRING",2,40,40,0,0,"",""
"FIELD","PRODUCT","PRICE","DOUBLE",42,8,15,0,0,"",""
"FIELD","SUPPLIER","SUPPLIER","WORD",0,2,5,0,0,"",""
"FIELD","SUPPLIER","SUPPLIER_NAME","STRING",2,40,40,0,0,"",""
"FIELD","SUPPLIER","TOWN","STRING",42,20,20,0,0,"",""
"FIELD","SUPPLIER","TELEPHONE","STRING",62,15,15,0,0,"",""
"FIELD","SUPPLIER_PRODUCT","SUPPLIER","WORD-REVERSED",0,2,5,0,0,"",""
"FIELD","SUPPLIER_PRODUCT","PRODUCT","WORD-REVERSED",2,2,5,0,0,"",""
"DB","DEMO","EASYSOFT_SQL_DEMO_DATA","VAX-RMS",""
"TABLE","DEMO","LASER","TABLE","LASER","LASER.DAT",""
"TABLE","DEMO","LASER_PG","TABLE","LASER","LASER.DAT","DEMO_LASER_PG.RATIN
   G='PG'"
"TABLE","DEMO","PRODUCT","TABLE","PRODUCT","PRODUCT.DAT",""
"TABLE","DEMO","SUPPLIER","TABLE","SUPPLIER","SUPPLIER.DAT",""
"TABLE","DEMO","SUPPLIER_PRODUCT","TABLE","SUPPLIER_PRODUCT","SUPPLIER_PRO
   DUCT.DAT",""
"COLUMN","DEMO","LASER","CATALOG_NUMBER","CATALOG_NUMBER","INTEGER",4,1,1,
   ""
"COLUMN","DEMO","LASER","DISC_NAME","DISC_NAME","VARCHAR",55,1,1,""
"COLUMN","DEMO","LASER","RETAIL_PRICE","RETAIL_PRICE","DOUBLE",8,1,1,""
"COLUMN","DEMO","LASER","RATING","RATING","VARCHAR",4,1,1,""
"COLUMN","DEMO","LASER","STUDIO","STUDIO","VARCHAR",3,1,1,""
"COLUMN","DEMO","LASER_PG","CATALOG_NUMBER","CATALOG_NUMBER","INTEGER",4,1
   ,1,""
"COLUMN","DEMO","LASER_PG","DISC_NAME","DISC_NAME","VARCHAR",55,1,1,""
"COLUMN","DEMO","LASER_PG","RETAIL_PRICE","RETAIL_PRICE","DOUBLE",8,1,1,""
"COLUMN","DEMO","LASER_PG","RATING","RATING","VARCHAR",4,1,1,""
"COLUMN","DEMO","LASER_PG","STUDIO","STUDIO","VARCHAR",3,1,1,""
"COLUMN","DEMO","PRODUCT","PRODUCT","PRODUCT","SMALLINT",2,1,1,""
"COLUMN","DEMO","PRODUCT","PRODUCT_NAME","PRODUCT_NAME","VARCHAR",40,1,1,"
   "
"COLUMN","DEMO","PRODUCT","PRICE","PRICE","DOUBLE",8,1,1,""
"COLUMN","DEMO","SUPPLIER","SUPPLIER","SUPPLIER","SMALLINT",2,1,1,""
"COLUMN","DEMO","SUPPLIER","SUPPLIER_NAME","SUPPLIER_NAME","VARCHAR",40,1,
   1,""
"COLUMN","DEMO","SUPPLIER","TOWN","TOWN","VARCHAR",20,1,1,""
"COLUMN","DEMO","SUPPLIER","TELEPHONE","TELEPHONE","VARCHAR",15,1,1,""
"COLUMN","DEMO","SUPPLIER_PRODUCT","SUPPLIER","SUPPLIER","SMALLINT",2,1,1,
   ""
"COLUMN","DEMO","SUPPLIER_PRODUCT","PRODUCT","PRODUCT","SMALLINT",2,1,1,""
"DBPRV","ADMIN","DEMO","","",1,1,1,1,"DATABASE"
```

Fixed Width Text Format

File Definition Record length = 113

Record Type	7	"FILE"
File Name	64	
Organisation	16	
Record Type	16	
Record Size	10	

Field Definition Record length = 336

Record Type	7	"FIELD"
File Name	64	
Field Name	64	
Data Type	64	
Offset	10	
Length	10	
Precision	10	
Scale	10	
Encrypted	1	
Format	64	(ASCII Dates/Times Only)
Default Value	32	

Database Definition Record Length = 209

Record Type	7	"DB"
Database Name	10	
Default Directory	128	
Driver Name	64	

Table Definition Record length = 591

Record Type	7	"TABLE"
Database Name	10	
Table Name	64	
Table Type	64	
File Name	64	
File Specification	128	
Criteria	254	

Column Definition Record length = 317

Record Type	7	"COLUMN"	
Database Name	10		
Table Name	64		
Column Name	64		
Field Name	64		
Column Data Type	64		
Length	10		
Updatable	1	(1) Updatable,	(0) Not Updatable
Visible	1	(1) Visible,	(0) Not Visible
Default Value	32		

User Definition Record length = 135

Record Type	7	"USER"
User Name	64	
Password	64	(Not Exported)

Database Privileges Definition Record length = 165

Record Type	7	"DBPRV"
User Name	64	
Database Name	10	
Database Username	32	
Database Password	32	
Allow Select	1	(1) Yes, (0) No
Allow Insert	1	(1) Yes, (0) No
Allow Update	1	(1) Yes, (0) No
Allow Delete	1	(1) Yes, (0) No
Security Level	16	

Table Privileges Definition Record length = 149

Record Type	7	"TBLPRV"
User Name	64	
Database Name	10	
Table Name	64	
Allow Select	1	(1) Yes, (0) No
Allow Insert	1	(1) Yes, (0) No
Allow Update	1	(1) Yes, (0) No
Allow Delete	1	(1) Yes, (0) No

APPENDIX H

Supported Scalar Functions

This appendix lists the scalar functions which are supported by Easysoft ODBC. Refer to Appendix D for a list of supported data types on which the scalar functions can operate.

Scalar functions are functions supported by an ODBC driver, which extend the functionality of that driver. They provide the means for an application to issue SQL statements to the ODBC driver/server which then call functions based on the input.

The syntax for use of a function is defined as:

{fn <*function name*> (<*function parameters*>)}

where *function name* is the name of the function that is to be called, and *function parameters* are the parameters which are to be passed with each function (these vary from function to function, please see the lists of scalar functions that follow).

For example, to perform an SQL query which returns a single column which contains the numeric length of a string field <*stringfieldname*> in characters from table <*tablename*> use the following SQL statement:

```
select {fn length(<stringfieldname>)}as stringcheck from
<tablename>;
```

String Functions

The following table lists string manipulation functions.

Character string literals used as arguments to scalar functions must be bounded by single quotes.

Arguments denoted as *str_exp* can be the name of a column, a string literal, or the result of another scalar function, where the underlying data type can be represented as one of the supported Character String data types.

Arguments denoted as *start, length* or *count* can be a numeric literal or the result of another scalar function, where the underlying data type can be represented as one of the supported Exact Numeric data types.

The string functions listed here are 1-based, that is, the first character in the string is character 1.

Function	**Description**
ASCII(*str_exp*)	Returns the ASCII code value of the leftmost character of *str_exp* as an integer.
CHAR(*code*)	Returns the character that has the ASCII code value specified by *code*. The value of *code* should be between 0 and 255; otherwise, the return value is data source dependent.
CONCAT(*str_exp1*, *str_exp2*)	Returns a character string that is the result of concatenating *str_exp2* to *str_exp1*. The resulting string is DBMS dependent.
DIFFERENCE(*str_exp1*, *str_exp2*)	Returns an integer value that indicates the difference between the values returned by the SOUNDEX function for *str_exp1* and *str_exp2*.
INSERT(*str_exp1*, *start*, *length*, *str_exp2*)	Returns a character string where *length* characters have been deleted from *str_exp1* beginning at *start* and where *str_exp2* has been inserted into *str_exp1*, beginning at *start*.
LCASE(*str_exp*)	Converts all upper case characters in *str_exp* to lower case.
LEFT(*str_exp*, *count*)	Returns the leftmost *count* of characters of *str_exp*.
LENGTH(*str_exp*)	Returns the number of characters in *str_exp*, excluding trailing blanks and the string termination character.
LOCATE(*str_exp1*, *str_exp2* [, *start*])	Returns the starting position of the first occurrence of *str_exp1* within *str_exp2*. The search for the first occurrence of *str_exp1* begins with the first character position in *str_exp2* unless the optional argument, *start*, is specified. If *start* is specified, the search begins with the character position indicated by the value of *start*. The first character position in *str_exp2* is indicated by the value 1. If *str_exp1* is not found within *str_exp2*, the value 0 is returned.
LTRIM(*str_exp*)	Returns the characters of *str_exp*, with leading blanks removed.
REPEAT(*str_exp*, *count*)	Returns a character string composed of *str_exp* repeated *count* times.
REPLACE(*str_exp1*, *str_exp2*, *str_exp3*)	Replaces all occurrences of *str_exp2* in *str_exp1* with *str_exp3*.
RIGHT(*str_exp*, *count*)	Returns the rightmost *count* of characters of *str_exp*.
RTRIM(*str_exp*)	Returns the characters of *str_exp* with trailing blanks removed.
SOUNDEX(*str_exp*)	Returns a data source dependent character string representing the sound of the words in *str_exp*.
SPACE(*count*)	Returns a character string consisting of *count* spaces.

SUBSTRING(*str_exp*, *start*, *length*)	Returns a character string that is derived from *str_exp* beginning at the character position specified by *start* for *length* characters.
UCASE(*str_exp*)	Converts all lower case characters in *str_exp* to upper case.

Numeric Functions

The following table describes numeric functions that are included in the ODBC scalar function set.

Arguments denoted as *numeric_exp* can be the name of a column, the result of another scalar function, or a numeric literal, where the underlying data type could be represented as one of the supported Exact Numeric or Approximate Numeric data types.

Arguments denoted as *float_exp* can be the name of a column, the result of another scalar function, or a numeric literal, where the underlying data type can be represented as DOUBLE.

Arguments denoted as *int_exp* (integer expression) can be the name of a column, the result of another scalar function, or a numeric literal, where the underlying data type can be represented as one of the supported Exact Numeric data types.

Function	**Description**
ABS(*numeric_exp*)	Returns the absolute value of *numeric_exp*.
ACOS(*float_exp*)	Returns the arccosine of *float_exp* as an angle, expressed in radians.
ASIN(*float_exp*)	Returns the arcsine of *float_exp* as an angle, expressed in radians.
ATAN(*float_exp*)	Returns the arctangent of *float_exp* as an angle, expressed in radians.
ATAN2(*float_exp1*, *float_exp2*)	Returns the arctangent of the x and y co-ordinates, specified by *float_exp1* and *float_exp2*, respectively, as an angle, expressed in radians.
CEILING(*numeric_exp*)	Returns the smallest integer greater than or equal to *numeric_exp*.
COS(*float_exp*)	Returns the cosine of *float_exp*, where *float_exp* is an angle expressed in radians.
COT(*float_exp*)	Returns the cotangent of *float_exp*, where *float_exp* is an angle expressed in radians.

DEGREES(*numeric_exp*) Returns the number of degrees converted from *numeric_exp* radians.

EXP(*float_exp*) Returns the exponential value of *float_exp*.

FLOOR(*numeric_exp*) Returns the largest integer less than or equal to *numeric_exp*.

LOG(*float_exp*) Returns the natural logarithm of *float_exp*.

LOG10(*float_exp*) Returns the base 10 logarithm of *float_exp*.

MOD(*int_exp1*, *int_exp2*) Returns the remainder (modulus) of *int_exp1* divided by *int_exp2*.

PI( ) Returns the constant value of pi as a floating point value.

POWER(*numeric_exp*, *int_exp*) Returns the value of *numeric_exp* to the power of *int_exp*.

RADIANS(*numeric_exp*) Returns the number of radians converted from *numeric_exp* degrees.

RAND([*int_exp*]) Returns a random floating point value using *int_exp* as the optional seed value.

ROUND(*numeric_exp*, *int_exp*) Returns *numeric_exp* rounded to *int_exp* places right of the decimal point. If *int_exp* is negative, *numeric_exp* is rounded to |*int_exp*| places to the left of the decimal point.

SIGN(*numeric_exp*) Returns an indicator or the sign of *numeric_exp*. If *numeric_exp* is less than zero, -1 is returned. If *numeric_exp* equals zero, 0 is returned. If *numeric_exp* is greater than zero, 1 is returned.

SIN(*float_exp*) Returns the sine of *float_exp*, where *float_exp* is an angle expressed in radians.

SQRT(*float_exp*) Returns the square root of *float_exp*.

TAN(*float_exp*) Returns the tangent of *float_exp* , where *float_exp* is an angle expressed in radians.

TRUNCATE(*numeric_exp*, *int_exp*) Returns *numeric_exp* truncated to *int_exp* places right of the decimal point. If *int_exp* is negative, *numeric_exp* is truncated to *int_exp1* places to the left of the decimal point.

Time and Date Functions

The following table lists time and date functions that are included in the ODBC scalar function set.

Arguments denoted as *timestamp_exp* can be the name of a column, the result of another scalar function, or a time, date, or timestamp literal, where the underlying data type could be represented as CHAR, VARCHAR, TIME, DATE or TIMESTAMP.

Arguments denoted as *date_exp* can be the name of a column, the result of another scalar function, or a date or timestamp literal, where the underlying data type could be represented as CHAR, VARCHAR, DATE or TIMESTAMP.

Arguments denoted as *time_exp* can be the name of a column, the result of another scalar function, or a time or timestamp literal, where the underlying data type could be represented as CHAR, VARCHAR, TIME or TIMESTAMP.

Values returned are represented as ODBC data types.

Function	**Description**
CURDATE( )	Returns the current date as a date value.
CURTIME( )	Returns the current local time as a time value.
DAYNAME(*date_exp*)	Returns a character string containing the data source-specific name of the day for the day portion of *date_exp*. Examples: Sunday for a data source that uses English, or Sonntag for a data source that uses German.
DAYOFMONTH(*date_exp*)	Returns the day of the month in *date_exp* as an integer value in the range 1 to 31.
DAYOFWEEK(*date_exp*)	Returns the day to the week in *date_exp* as an integer value in the range 1 to 7, where 1 represents Sunday.
DAYOFYEAR(*date_exp*)	Returns the day of the year in *date_exp* as an integer value in the range 1 to 366.
HOUR(*time_exp*)	Returns the hour in *time_exp* as an integer value in the range 0 to 23.
MINUTE(*time_exp*)	Returns the minute in *time_exp* as an integer value in the range 0 to 59.
MONTH(*date_exp*)	Returns the month in *date_exp* as an integer value in the range 1 to 12.

MONTHNAME(*date_exp*)	Returns a character string containing the data source-specific name of the month (for example, January for a data source that uses English, or Januar for a data source that uses German) for the month portion of *date_exp*.
NOW( )	Returns the current date and time as a timestamp value.
QUARTER(*date_exp*)	Returns the quarter in *date_exp* as an integer value in the range 1 to 4, where 1 represents January 1 to March 31.
SECOND(*time_exp*)	Returns the second in *time_exp* as an integer value in the range 0 to 59.
WEEK(*date_exp*)	Returns the week of the year in *date_exp* as an integer value in the range 1 to 53.
YEAR(*date_exp*)	Returns the year in *date_exp* as an integer value. The range depends on the data source.

System Functions

The following table lists system functions that are included in the ODBC scalar function set.

Arguments denoted as *exp* can be the name of a column, the result of another scalar function, or a literal, where the underlying data type could be represented as TINYINT, SMALLINT, INTEGER, DOUBLE, DATE, TIME or TIMESTAMP.

Arguments denoted as *value* can be a literal constant, where the underlying data type can be represented as TINYINT, SMALLINT, INTEGER, DOUBLE, DATE, TIME or TIMESTAMP.

Values returned are represented as ODBC data types.

Function	**Description**
DATABASE( )	Returns the name of the database corresponding to the connection handle (*hdbc*). (The name of the database is also available by calling **SQLGetConnectOption** with the CURRENT_QUALIFIER connection option.)
IFNULL(*exp*, *value*)	If *exp* is null, *value* is returned. If *exp* is not null, *exp* is returned. The possible data type(s) of *value* must be compatible with the data type of *exp*.
USER( )	Returns the user's authorisation name. (The user's authorisation name is also available via **SQLGetInfo** by specifying the information type: USER_NAME.)

Data Type Conversion Function

The following table shows the data type conversion function that is included in the ODBC scalar function set.

Function	Description
CONVERT(*value_exp*, *data_type*)	Returns the value specified by *value_exp*; the value is converted to the *data_type* specified.

APPENDIX I

Error Messages

This appendix lists the error messages which may be generated by components of the Easysoft system. A word in angle brackets (<>) indicates that an appropriate value is substituted in the error message when that message is displayed.

Server-side Errors

This section lists the error messages which may be generated as a result of error conditions associated with Easysoft SQL.

<string>
<string> is/are unsupported in this version
A field could be found at offset <numeric value> in file <string>
Ambiguous source for column <string>
Column <string> does not exist in table
Column name expected
Column number <numeric value> could not be found
Column, literal or parameter expected, found <string>
Database qualifier <string> is invalid or insufficient privileges exist
Default Database <string> is invalid
Error occurred creating temporary file <string>
Error occurred opening temporary file
Error opening log file '<string>' - Check syntax/permissions
Expected boolean operator
Expected expression in AND clause
Expected expression in OR clause
Expected expression in SQL predicate
Expected ODBC extension terminator
Expected ODBC extension terminator '}' or '*)--'
Expected outer join extension
Fatal Internal System Error in module <string> at line <numeric value>
File organisation '<string>' specified for the file '<string>' is invalid
File Organisation catalog definition incorrect for file <string>. It should be <string>
FROM expected
Full outer join not supported
Incompatible join specification for <string>
Invalid arguments from insert function
Invalid attribute string for data type <string>
Invalid catalog location <string>
Invalid data type conversion from <string> to <string>
Invalid database version <string>, expected <string>
Invalid Date '<string>'
Invalid Date Literal. Syntax {d 'YYYY-MM-DD'}
Invalid driver specified in ES_DRIVER
Invalid format for data type <string>

Invalid key identifier
Invalid local ISAM data type <string>
Invalid number <numeric value> was supplied for FetchRecordCount
Invalid number of active statements <numeric value>
Invalid number of arguments specified for function
Invalid ODBC extension identifier '<string>'
Invalid outer join expression
Invalid reference '<string>.<string>' in outer join expression
Invalid SQL data type <string>
Invalid statement <string>
Invalid Time '<string>'.
Invalid Time Literal. Syntax {t 'HH:MM:SS'}
Invalid Timestamp '<string>'
Invalid Timestamp Literal. Syntax {ts 'YYYY-MM-DD HH:MM:SS'}
Invalid use of parameter markers
Invalid user and password combination supplied for catalog
Licence for <string> cannot be found
Licence for <string> has expired
Maximum number of files open
Memory Allocation Failure
Missing bracket after <string>
No client licences are available for <string>
No column exists for the field <string>
No column information could be found for table
No database privileges exist for this user
No DB_VERSION record in ES_CNTRL
No Default Database Specified use SET DATABASE 'database_name'
No delete privileges exist for the database
No delete privileges exist for the table
No field information could be found for table
No insert privileges exist for the database
No insert privileges exist for the table
No keys are defined for file
No more streams available for file
No select privileges exist for the database
No select privileges exist for the table
No server licences are available for <string>
No update privileges exist for the database
No update privileges exist for the table
Only columns are allowed in outer join expression
Parameters cannot be specified in aggregate functions
Precision exceeded for type <string>
Query timed out
Record Size catalog definition incorrect for file <string>. It should be <numeric value>
Record Type catalog definition incorrect for file <string>. It should be <string>
RMS Error '<string>'
Sort failed with error '<string>'
Specified column <string> could not be found
Specified field <string> could not be found
Specified file <string> could not be found
Syntax error
Syntax error after '<string>' in '<string>'
Syntax error with '<string>' in '<string>'
Table <string> does not exist

Table name expected
Table or correlation name '<string>' is not specified in the FROM clause
Tablename specification is invalid
The connection string specified is invalid
Too many result columns in select
Too many tables in select
Type Mismatch
Unable to initialize ISAM driver
Unable to lookup table <string>
Unable to obtain error information from data driver while <string> <string>
Unknown error number
Unsupported or unknown function <string>()
Value is greater than permitted for SQL type <string> (maximum = <string>)
Value is greater than permitted size for data type <string>
Value is less than permitted for SQL type <string> (minimum = <string>)
Value is out of range for data type <string>
Value list must have as many entries as column list

Client-side Errors

Generic Network Errors

<string> not supported.
Communication link rating has been reduced to <numeric value> bytes per packet.
Connection closed due to unreliability.
Data lost or truncated.
Duplicate packet type <numeric value> for message <numeric value> received has been ignored.
Incompatible protocol version.
Invalid connection mode.
Invalid escaped value of <numeric value>.
Invalid field type <numeric value> expected in packet.
Invalid network connection parameter <numeric value>.
Invalid network transport parameter <numeric value> supplied.
Invalid or missing connection value specified for '<string>'.
Invalid packet data.
Invalid packet header.
Invalid packet type 0x<numeric value>.
Invalid process started, this may be caused by a server from a previous version running on the remote machine.
Invalid transport <numeric value> specified.
Length of '<string>' is too large, maximum length is <numeric value>.
Message was truncated.
Network transport does not support this function.
Network transport not initialised.
Next packet sync received before end current packet.
No connection option specified.
Operation timed out after <numeric value> seconds.
Packet NAK received for offset <numeric value>.
Remote host closed connection.

Remote host requested connection reset.
Remote service <string> failed to start on <string>.
Unable to establish connection rating.
Unable to match packet <string> for offset <numeric value>.
Unexpected packet type 0x<numeric value>.

DECNET Errors

DECnet error '<string>' occurred creating socket
DECnet error '<string>' occurred during connect to object <string> on <string>
DECnet error '<string>' occurred during sioctl
DECnet error '<string>' occurred obtaining node address for <string>
DECnet error '<string>' occurred setting socket option
DECnet error '<string>' occurred trying to send data
DECnet error '<string>' occurred when disconnecting
DECnet error '<string>' occurred while checking for incoming data
DECnet error '<string>' occurred while receiving data
DECnet is not installed correctly
Failed to convert address <string>
Failed to load DNETDLL.DLL (<numeric value>)
Incomplete packet was sent
Unable to get entry point for <string>
Unable to obtain DECnet address for the node <string>
Unable to obtain local host name (<numeric value>)

Windows Socket Errors

Failed to load WINSOCK.DLL (<numeric value>)
Installed WINSOCK.DLL does not provide the required version
Installed WINSOCK.DLL is not compatible
Link disconnect by remote server
Only <numeric value> out <numeric value> bytes were sent successfully
TCP/IP network subsystem is not ready
Unable to get entry point for '<string>'
Unexpected windows sockets error '<string>' (<numeric value>) occurred
Windows socket error '<string>' (<numeric value>) occurred closing socket
Windows socket error '<string>' (<numeric value>) occurred connecting to service '<string>' on '<string>'
Windows socket error '<string>' (<numeric value>) occurred during ioctl
Windows socket error '<string>' (<numeric value>) occurred during select
Windows socket error '<string>' (<numeric value>) occurred obtaining node address for '<string>'
Windows socket error '<string>' (<numeric value>) occurred obtaining node name
Windows socket error '<string>' (<numeric value>) occurred obtaining port for service '<string>'
Windows socket error '<string>' (<numeric value>) occurred receiving data
Windows socket error '<string>' (<numeric value>) occurred sending data
Windows socket error '<string>' (<numeric value>) occurred setting socket option '<string>'
Windows socket error '<string>' (<numeric value>) occurred when creating socket
Windows socket error '<string>' (<numeric value>) occurred when setting blocking hook

APPENDIX J

QEP Scoring Mechanism

The Query Execution Plan (QEP) index scoring mechanism consists of a set of rules which attempt to determine the best index (if any) to use to return data from a table. There is no guarantee that the best index will always be chosen, because to a certain extent, the best index to use depends upon the data contained in the file.

The rules for the scoring mechanism are shown below.

General Rules and Comments

1. For each table, the score for each of the indexes available on that table is evaluated. The index with the highest score is used in the query. If two or more indexes both have the same highest score, then the index with the lowest index number is used. For each index, the score from the highest scoring qualifier (defined in 2) is taken as the index score. There can be many qualifiers for each index.
2. A qualifier is a set of conditions that will return a result set (which may contain zero records). (A qualifier is equivalent to a *boolean term* in SQL syntax). Examples:

   a. WHERE CATALOG_NUMBER = 42

   b.
   ```
   WHERE (CATALOG_NUMBER > 5) OR (CATALOG_NUMBER < 20)
          _______ qualifier 1_______   ________ qualifier 2 _______
   ```

   c.
   ```
   WHERE (CATALOG_NUMBER > 5 AND STUDIO LIKE ‘M%’) OR (CATALOG_NUMBER < 20)
          ______________________ qualifier 1 ______________   ________ qualifier 2 ______
   ```

3. For each qualifier evaluated, the total score is the sum of the scores obtained for each of the fields in the index (see “Scoring Mechanism”, page 201).
4. The score for an index field with duplicates is the same as the score for a field without duplicates.
5. The length of a field is factored as part of the score.
6. Each field is scored based on its position in the key. The first field in the key is given the highest factor, and the last field in the key is given the lowest factor.
7. The columns which are selected in the query and the order of the qualifiers in the query have no effect on the index scoring.

8. *Fast Mode* is a term that appears in the output from the QEP logging. It refers to special logic that substantially improves performance by making better use of supplied criteria information. This information would previously only have been used purely as an expression evaluator. *Fast Mode* is automatically applied in situations where
    - The key contains more than one field
    - A criteria exists on a field such that all previous fields do not have a criteria using an operator that receives a field score of 4, namely equal (=), IN or IS NULL (see the next section for information on field scores).

    A few examples are given based on a key containing the fields N, S, L

    Fast Mode is off
    N > ‘A’
    N = ‘A’ AND S > ‘B’
    N = ‘A’ AND S = ‘B’ AND S > ‘C’

    Fast Mode is on
    S > ‘B’
    L = ‘C’
    N > ‘A’ AND S = ‘B’ AND L = ‘C’

9. The scoring for multi-table queries is different from the scoring for single-table queries. If the multi-table query contains an equi-join, that is, a join based on equality, then the score for the join column(s) is evaluated in the second (and subsequent) tables in addition to the normal scoring for the table. For example, say there are two tables, T1(A, B, C) and T2(C, D, E), against which the following query is posed:

    ```
    SELECT     <columns>
    FROM       T1, T2
    WHERE      <qualifiers> AND T1.C = T2.C
    ```

    In addition to the scoring for T1 and T2 which is based on the qualifiers which appear in <qualifiers>, an additional score will be calculated for column T2.C using the field score for the equal operator (4).

10. In multi-table queries, the order in which tables are accessed can be explicitly specified (this is particularly useful if you are using Microsoft Query). This is two stage process.

    First, the TABLE_ORDER Selector option in the Special Columns section of the Column Definitions dialog box in the Easysoft PC Administrator (not available with all Easysoft products) must be selected, and the catalog definitions must be uploaded to the server.

Second, the required order of the table access can be specified in the WHERE clause of the SQL query. For example:

SELECT <columns> FROM <table1>, <table2>
WHERE <table1>.TABLE_ORDER = 2 AND <table2>.TABLE_ORDER = 1

In this case <table2> will be accessed by the query first, and then <table1> will be accessed.

9. The QEP index selection can be overridden. This is two stage process.

   First, the INDEX Number option in the Special Columns section of the Column Definitions dialog box in the Easysoft PC Administrator (not available with all Easysoft products) must be selected, and the catalog definitions must be uploaded to the server.

   Second, the required index order can be stated in the WHERE clause of the SQL query. For example:

   SELECT <columns> FROM <table>
   WHERE <table>.INDEX_NUMBER = <n>

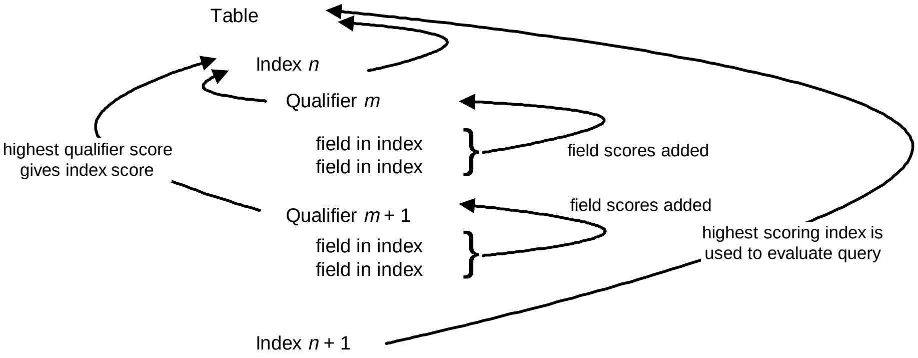


Figure 89. Index scoring summation

Scoring Mechanism for a Single Qualifier

The index score for a single qualifier is the sum of the scores obtained for each of the fields in the index that are related to the qualifier:

= □Field score × Field length(k) × Field factor(k) × exp ((key offset) / (index factor))
where *n* is the number of fields in the index

Field score is dependent on the operator in the qualifier

Operator	**Field Score**	**Example**
=, IN, IS NULL	4	NAME = 'SMITH'
LIKE, BETWEEN	2	NAME LIKE 'SM%'
<, <=, >, >=	1	NAME < 'S'
<>, NOT, IS NOT NULL	0	NAME NOT BETWEEN 'S' AND 'W'

Note 1: With a LIKE qualifier, only initial known characters are scored. Examples

NAME LIKE '%SON'	score = zero
NAME LIKE 'SM%SON'	first two characters are evaluated in score

Note 2: The operators <>, NOT and IS NOT NULL have a field score of zero which results in the index score for the field on which they operate also being zero. This is because they cannot be used as a starting value by the index.

Note 3: If > (or >=) and < (or <=) are combined using an AND operator on the same field in the same qualifier then the field scores for each operation are added (that is, the field score for the column = 2). Example:

```
WHERE (CATALOG_NUMBER > 5) AND (CATALOG_NUMBER <= 20)
```

This combination can be expressed using BETWEEN, which scores 2.

Note 4: The index scoring mechanism is a heuristic tool which cannot guarantee optimum results. For example, it is possible to express the condition in Note 2 using the IN operator (which scores 4, rather than 2). However, it is unlikely that the IN operator would be used with such a wide range of values. IN scores higher, because we expect that only a few values would be specified.

Field length is the length of the index field.

Field factor
for all operators except LIKE = 1.0
for LIKE operator = (number of supplied characters) / (field length)

Key offset = (maximum key length) **-** (index field offset)

Maximum key length = 255

Index factor = (maximum key length / 10)

Known Limitations

In a multi-table query, the order of the tables is not optimised. If the tables are supplied in, say, alphabetical order and if the first table doesn't contain the best index, then the query may take considerably longer than if the first table had the best index.

Scoring Examples

To demonstrate how the indexes are used, we show output from the Easysoft log file on the server.

The columns that are returned in the query are irrelevant to the choosing of the index.

For exemplary purposes we will use the LASER table (which contains information about laser discs), which is one of the tables contained in the Easysoft demonstration data.

Table 8. LASER table

Column name	Data type	Size
CATALOG_NUMBER	INTEGER	4
DISC_NAME	VARCHAR	55
RETAIL_PRICE	DOUBLE	8
RATING	VARCHAR	4
STUDIO	VARCHAR	3

Table 9. LASER indexes

Index name	Dup	Collation	Fields in index
LASER_001	No	Ascending	CATALOG_NUMBER
LASER_002	Yes	Ascending	DISC_NAME
LASER_003	Yes	Ascending	RATING
LASER_004	Yes	Descending	STUDIO

Note: index names are 1-based. However, the QEP log is 0-based. Thus, for example, the index labelled LASER_002 will be shown as Index 1 in the QEP. Similarly, the first qualifier in the QEP is shown as qualifier 0.

Query 1

SQL condition: WHERE RATING = ‘15’

Scoring for Index 2 (LASER_003), qualifier 0 (RATING=15)

Field score	4	due to equal (=) operator
Index field length	4	RATING field is 4 bytes
Field factor	1	for all operators (except LIKE) = 1
Index field offset	0	offset of RATING field in index LASER_003
Key offset	255	maximum key length (255) - index field offset (0)
Index factor	25.5	maximum key length / 10

score = □ Field score × Field length(k) × Field factor(k) × exp ((key offset) / (index factor))

= 4 × 4 × 1 × exp ((255-0) / 25.5) = 352,423
There is only one field to evaluate in the index.

There is only one qualifier in this query. It scores zero on all the other indexes, thus index LASER_003 (index number 2) will be used to evaluate the query.

SQL and QEP log for query 1

```
===================================================================
==
EXECUTE STATEMENT

  SELECT DEMO_LASER.CATALOG_NUMBER, DEMO_LASER.DISC_NAME,
DEMO_LASER.
  RETAIL_PRICE, DEMO_LASER.RATING, DEMO_LASER.STUDIO,
DEMO_LASER.CATA
  LOG_NUMBER
FROM DEMO_LASER DEMO_LASER
WHERE (DEMO_LASER.RATING='15')

===================================================================
==
QUERY EXECUTION PLAN

Index Scores

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 0   Index :
0
  Score : 0
          Column  1 - Score          0 - CATALOG_NUMBER
          Key Usage: 0%

          Fast mode available: No

  Table : DEMO_LASER      (DEMO_LASER      ) Qualifier : 0   Index : 1
  Score : 0
          Column  1 - Score          0 - DISC_NAME
          Key Usage: 0%
          Fast mode available: No

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 0   Index :
2
  Score : 352423
          Column  1 - Score     352423 - RATING
          Key Usage: 100%

          Fast mode available: No

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 0   Index :
3
  Score : 0
          Column  1 - Score          0 - STUDIO
          Key Usage: 0%

          Fast mode available: No

Qualifier 0

  Table      : DEMO_LASER (DEMO_LASER)
  Index      : 2        Start Mode : 7 (Greater than or equal)
  Start with :
    COLUMN(DEMO_LASER.RATING) = VALUE(STRING("15"))
  Finish with :
    COLUMN(DEMO_LASER.RATING) = VALUE(STRING("15"))
  Qualifiers:
    COLUMN(DEMO_LASER.RATING) = VALUE(STRING("15"))

Summary:  Tables [1]   Index Sort [No]   Quals [1]
$
```

Index number 2 is used for this query. It has the highest score.

Query 2

SQL condition:

```
  WHERE (CATALOG_NUMBER>5 AND CATALOG_NUMBER<20) OR
          (STUDIO LIKE 'ME%')
```

Scoring for index 0 (LASER_001), qualifier 1 (CATALOG_NUMBER>5 AND CATALOG_NUMBER<20)

Field score	2	less than and greater than operators score 1 each
Index field length	4	CATALOG_NUMBER field is 4 bytes
Field factor	1	for all operators (except LIKE) = 1
Index field offset	0	offset of CATALOG_NUMBER field in index LASER_001
Key offset	255	maximum key length (255) - index field offset (0)
Index factor	25.5	maximum key length / 10

score = □ Field score × Field length(k) × Field factor(k) × exp ((key offset) / (index factor))

= 2 × 4 × 1 × exp ((255-0) / 25.5) = 176,211

There is only one field to evaluate in the index.

Scoring for index 3 (LASER_004), qualifier 0 (STUDIO Like 'ME%')

Field score	2	from LIKE operator
Index field length	3	STUDIO field is 3 bytes
Field factor	0.67	supplied characters / total characters = 2/3 = 0.67
Index field offset	0	offset of STUDIO field in index LASER_004
Key offset	255	maximum key length (255) - index field offset (0)
Index factor	25.5	maximum key length / 10

score = □ Field score × Field length(k) × Field factor(k) × exp ((key offset) / (index factor))

= 2 × 3 × 0.67 × exp ((255-0) / 25.5) = 88,105

There is only one field to evaluate in the index.

All other qualifiers on all other indexes score zero, thus index LASER_001 will be used to evaluate the query.

SQL and QEP log for query 2

```
=====================================================================
EXECUTE STATEMENT

SELECT DEMO_LASER.CATALOG_NUMBER, DEMO_LASER.DISC_NAME,
DEMO_LASER.RETAIL_PRICE, DEMO_LASER.RATING, DEMO_LASER.STUDIO,
DEMO_LASER.CATALOG_NUMBER
FROM DEMO_LASER DEMO_LASER
WHERE (DEMO_LASER.CATALOG_NUMBER>5 AND DEMO_LASER.CATALOG_NUMBER<20)
OR (DEMO_LASER.STUDIO Like 'ME%')

====================================================================
QUERY EXECUTION PLAN

Index Scores

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 0   Index :
0
  Score : 0
          Column  1 - Score          0 - CATALOG_NUMBER
          Key Usage: 0%

          Fast mode available: No

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 1   Index :
0
  Score : 176211
          Column  1 - Score     176211 - CATALOG_NUMBER
          Key Usage: 50%

          Fast mode available: No

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 0   Index :
1
  Score : 0
          Column  1 - Score          0 - DISC_NAME
          Key Usage: 0%

          Fast mode available: No

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 1   Index :
1
  Score : 0
          Column  1 - Score          0 - DISC_NAME
          Key Usage: 0%

          Fast mode available: No

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 0   Index :
2
  Score : 0
          Column  1 - Score          0 - RATING
          Key Usage: 0%

          Fast mode available: No

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 1   Index :
2
  Score : 0
          Column  1 - Score          0 - RATING
          Key Usage: 0%
          Fast mode available: No

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 0   Index :
3
  Score : 88105
          Column  1 - Score      88105 - STUDIO
          Key Usage: 66%

          Fast mode available: No

  Table : DEMO_LASER       (DEMO_LASER      ) Qualifier : 1   Index :
3
  Score : 0
          Column  1 - Score          0 - STUDIO
          Key Usage: 0%

          Fast mode available: No


Qualifier 0

  Table      : DEMO_LASER (DEMO_LASER)
  Index      : 4        Start Mode : 7 (Greater than or equal)
  Start with :
    COLUMN(DEMO_LASER.STUDIO) >= VALUE(STRING("ME"))
  Finish with :
    COLUMN(DEMO_LASER.STUDIO) <= VALUE(STRING("ME~"))
  Qualifiers:
    COLUMN(DEMO_LASER.STUDIO) LIKE VALUE(STRING("ME%"))

Qualifier 1

  Table      : DEMO_LASER (DEMO_LASER)
  Index      : 0        Start Mode : 6 (Greater than)
  Start with :
    COLUMN(DEMO_LASER.CATALOG_NUMBER) > VALUE(INTEGER(5))
  Finish with :
    COLUMN(DEMO_LASER.CATALOG_NUMBER) < VALUE(INTEGER(20))
  Qualifiers:
    COLUMN(DEMO_LASER.CATALOG_NUMBER) > VALUE(INTEGER(5))
    COLUMN(DEMO_LASER.CATALOG_NUMBER) < VALUE(INTEGER(20))

Summary:  Tables [1]   Index Sort [No]   Quals [2]
$
```

Bibliography

This bibliography lists a few sources of information pertaining to ODBC and related topics.

Inside ODBC, K. Geiger. Published by: Microsoft Press, 1995. ISBN 1-55615-815-7.

Microsoft ODBC 2.0 Programmer's Reference and SDK Guide, Microsoft. Published by Microsoft Press, 1994. ISBN 1-55615-658-8.

Microsoft TechNet CD. This CD contains technical information relevant to the Microsoft Corporation. The CD is updated frequently.

SQL - *The Standard Handbook,* S. J. Cannan & G. A. M. Otten. Published by: McGraw Hill, 1993. ISBN 0-07-707664-8.

Using ODBC 2, R. Gryphon, L. Charpentier, J. Oelschlaeger, A. Shoemaker, J. Cross, A. W. Lilley. QUE Corporation, 1995. ISBN 0-7897-0015-8.

The ODBC Solution - *Open Database Connectivity in Distributed Environments*, R. Signore, J. Creamer, M. O. Stegman. Published by McGraw-Hill,1995. ISBN 0-07-911880-1.

Index

Page numbers for items which appear in alphabetic order in the Glossary, Data Types, Supported SQL (for SQL reserved words), Supported Scalar Functions and Error Messages appendices are not included in this index.