

# User Guide and Reference Manual

## **Easysoft® ODBC for CODA**

**Easysoft Limited**

Published by

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

©1997 – 2006 Easysoft Limited. All rights reserved.

Information in this document is subject to change without notice. Companies, names, and data used in examples are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the written permission of Easysoft Limited.

The names of companies referred to herein, their corporate logos, the names of their hardware and software may be trade names, trademarks or registered trademarks of their respective owners.

Easysoft and the Easysoft logo are registered trademarks of Easysoft Limited.

The software described in this document is provided under a licence agreement and may be used only in accordance with the terms of the agreement.

Version 15.

This manual documents version 1.5 of Easysoft ODBC for CODA.

---

# 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 it 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: licensing@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](mailto:documentation@easysoft.com)), FAX or mail to Easysoft (attention of Easysoft Documentation).



---

# Contents

<b>Licence Agreement</b>	<b>iii</b>
<b>Contacting Easysoft</b>	<b>vii</b>
<b>Contents</b>	<b>ix</b>
<b>List of Figures</b>	<b>xii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>Welcome</b>	<b>xvii</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
Fundamentals of ODBC	1
ODBC Architecture	2
Driver Types	3
Conformance	4
The Easysoft System	5
Easysoft ODBC for CODA Architecture	5
Data Sources, Databases and Catalogs	7
Overall Process of using Easysoft Software	8
Accounting Overview	9
CODA Overview	11
Company Structure	12
Using CODA	13
External Access to CODA Data	15
<b>Chapter 2. Installation and Licensing on the Server</b>	<b>17</b>
Preparation	18
Step 1. Log on to Server	18
Step 2. Ensure 20000 Blocks Available	18
Step 3. Availability of DEC C Runtime	19
Step 4. Ensure no PC Clients are using Easysoft Software	20
Step 5. Special Instructions for Upgrade - v1.1.6 and Below	20
Step 6. Transfer Files from Floppy Disk to Server	21
Step 7. Unzip Files	22
Installation	22
Step 8. Install Easysoft Server Component	23
Step 9. Run CODA Setup Routine	27
Step 10. Set Quotas	30
Step 11. Edit the System Startup File	31
Licensing	31
Step 12. Generate Site Number	32

---

Step 13. Obtain Licence Key	32
Step 14. License Product	33
Step 15. Check Installation	33
Maintenance	34
Re-licensing	34
Changing Catalog Passwords	35
Adding CODA Companies	37
Changing CODA Usernames, Passwords and Privileges	39
Ordering BALANCE and BUDGET Data	42
<b>Chapter 3. Host Administrator Reference</b>	<b>43</b>
<b>Chapter 4. ODBC Driver Installation</b>	<b>51</b>
Starting the Driver Installation	52
Driver Installation Steps	53
<b>Chapter 5. Managing Data Sources</b>	<b>57</b>
Microsoft ODBC Administrator	57
Adding a Data Source	58
Removing or Modifying a Data Source	66
Drivers Tab	67
Tracing Tab	67
<b>Chapter 6. CODA Table Descriptions</b>	<b>69</b>
<b>Chapter 7. Using Easysoft ODBC</b>	<b>99</b>
Connecting to CODA	99
Creating the Master Files	102
Initialise the System	102
Master Files	103
Create Hierarchical Account Groups	107
Post to Books	110
<b>Chapter 8. Easysoft Excel Macro for CODA</b>	<b>113</b>
Macro Installation	114
Initialisation	116
Masters-related options	118
Download	118
Upload	120
Reports-related options	121
Budgets	122
Fixed Reports	124
Input and Intraday	129
Lookups	132
Reports	133

---

<b>Chapter 9. Troubleshooting</b>	<b>137</b>
Server-side Troubleshooting	137
Client-side Troubleshooting	141
CODA-specific Issues	143
<b>Appendix A. Names and Passwords</b>	<b>147</b>
<b>Appendix B. Files Used</b>	<b>149</b>
ODBC Files	149
EASYSOFT.INI	150
CODAXREF.COM	151
<b>Appendix C. Glossary</b>	<b>152</b>
<b>Appendix D. Data Types</b>	<b>158</b>
<b>Appendix E. Easysoft Catalog</b>	<b>159</b>
Catalog Tables	160
<b>Appendix F. Supported SQL</b>	<b>163</b>
SQL Statements	163
SQL Reserved Words	164
<b>Appendix G. Import Export Formats</b>	<b>167</b>
<b>Appendix H. Supported Scalar Functions</b>	<b>169</b>
String Functions	169
Numeric Functions	171
Time and Date Functions	173
System Functions	174
Data Type Conversion Function	175
<b>Appendix I. Error Messages</b>	<b>176</b>
Server-side Errors	176
Client-side Errors	178
CODA Errors	180
<b>Appendix J. QEP Scoring Mechanism</b>	<b>182</b>
General Rules and Comments	182
Scoring Mechanism for a Single Qualifier	185
Scoring Example	186
<b>Bibliography</b>	<b>188</b>
<b>Index</b>	<b>189</b>

# List of Figures

Figure 1. ODBC Architecture _____	2
Figure 2. Single- and multi-tier driver architecture _____	3
Figure 3. Easysoft software architecture for CODA _____	5
Figure 4. Relationships between components _____	7
Figure 5. The Three Ledgers _____	9
Figure 6. Structure of the general ledger _____	10
Figure 7. Relationship between control accounts and ledgers _____	11
Figure 8. CODA Group Architecture _____	12
Figure 9. Ledger links _____	14
Figure 9. ODBC Setup (Welcome) dialog box _____	53
Figure 10. Software License Agreement dialog box _____	53
Figure 11. User Information dialog box _____	54
Figure 12. Start Copying Files dialog box _____	55
Figure 14. Setup Complete _____	55
Figure 14. User DSN tab _____	57
Figure 15. Create New Data Source dialog box _____	59
Figure 16. Easysoft ODBC Setup dialog box _____	59
Figure 17. Easysoft ODBC Login Prompt _____	63
Figure 18. Easysoft ODBC Settings dialog box _____	63
Figure 19. Easysoft ODBC Logging dialog box _____	65
Figure 20. Easysoft ODBC Capture dialog box _____	66
Figure 21. ODBC Drivers tab _____	67
Figure 22. Tracing tab _____	67
Figure 30. Link dialog box _____	100
Figure 31. SQL Data Sources dialog box _____	100
Figure 32. Link Tables dialog box _____	101
Figure 33. Database dialog box _____	101
Figure 34. CODA data in Access _____	101
Figure 35. NEWCO Chart of Accounts _____	104
Figure 36. NEWCO account groups _____	108
Figure 37. NEWCO account group data in Access _____	109
Figure 38. Tables used for posting to the books _____	111
Figure 39. Coda macro menu options _____	113
Figure 40. Add-ins dialog box _____	114
Figure 41. Browse dialog box _____	115
Figure 42. Options dialog box _____	116

---

Figure 43. Available CODA Companies dialog box _____	117
Figure 44. Download dialog box _____	119
Figure 45. Setup Budget Format dialog box _____	122
Figure 46. Budget Download dialog box _____	123
Figure 47. Budget Upload dialog box _____	123
Figure 48. Budget Variance Selection dialog box _____	124
Figure 49. Example ABV Report _____	125
Figure 50. Age Analysis dialog box _____	125
Figure 51. Example Age Analysis Report _____	126
Figure 52. Balance Sheet Selection Criteria dialog box _____	126
Figure 53. Example Balance Sheet Report _____	127
Figure 54. Profit & Loss Selection dialog box _____	127
Figure 55. Example Profit and Loss Report _____	128
Figure 56. Trial Balance dialog box _____	128
Figure 57. Example Trial Balance _____	129
Figure 58. Details Download dialog box _____	130
Figure 59. Invoice Setup dialog box _____	130
Figure 60. Invoice Upload dialog box _____	131
Figure 61. Company Lookup dialog box _____	132
Figure 62. Setup New Report example _____	133
Figure 63. Password relationships _____	147
Figure 64. Easysoft Catalog _____	159
Figure 65. Index scoring summation _____	184

# List of Tables

Table 1. Information required for the Easysoft ODBC Setup _____	61
Table 2. Easysoft Demonstration Defaults _____	62
Table 3. Information required for Easysoft ODBC Settings _____	64
Table 4. Easysoft ODBC Logging Information _____	65
Table 5. List of CODA tables _____	69
Table 6. ACCOUNT column details _____	71
Table 7. ACCOUNT index details _____	71
Table 8. ACCOUNT_GROUP column details _____	71
Table 9. ACCOUNT_GROUP index details _____	72
Table 10. BALANCE column details _____	72
Table 11. BALANCE index details _____	72
Table 12. BUDGET column details _____	73
Table 13. BUDGET index details _____	73
Table 14. COMPANY column details _____	74
Table 15. COMPANY index details _____	75
Table 16. CURRENCY column details _____	75
Table 17. CURRENCY index details _____	76
Table 18. CURRENCY_RATE column details _____	76
Table 19. CURRENCY_RATE index details _____	76
Table 20. DESTINATION column details _____	77
Table 21. DESTINATION index details _____	78
Table 22. DETAILS column details _____	78
Table 23. DETAILS index details _____	80
Table 24. DETAILS_COMMENT column details _____	80
Table 25. DETAILS_COMMENT index details _____	80
Table 26. DOCUMENT_MASTER column details _____	80
Table 27. DOCUMENT_MASTER index details _____	82
Table 28. GROUP_HIERARCHY column details _____	82
Table 29. GROUP_HIERARCHY index details _____	82
Table 30. INTRAY column details _____	83
Table 31. INTRAY index details _____	84
Table 32. INTRAY_COMMENT column details _____	84
Table 33. INTRAY_COMMENT index details _____	85
Table 34. LEDGER column details _____	85
Table 35. LEDGER index details _____	85
Table 36. MOTHBALL column details _____	85

---

Table 37. MOTHBALL index details	87
Table 38. MOTHBALL_COMMENT column details	87
Table 39. MOTHBALL_COMMENT index details	87
Table 40. NAME column details	88
Table 41. NAME index details	89
Table 42. NAME_ADDRESS column details	90
Table 43. NAME_ADDRESS index details	90
Table 44. NAME_BANK column details	90
Table 45. NAME_BANK index details	91
Table 46. NAME_COMMENT column details	91
Table 47. NAME_COMMENT index details	91
Table 48. NAME_CONTACT column details	91
Table 49. NAME_CONTACT index details	91
Table 50. NAME_LEDGER_COMMENT column details	92
Table 51. NAME_LEDGER_COMMENT index details	92
Table 52. NAME_OWNER column details	92
Table 53. NAME_OWNER index details	92
Table 54. NOMINAL column details	93
Table 55. NOMINAL index details	93
Table 56. NOMINAL_COMMENT column details	93
Table 57. NOMINAL_COMMENT index details	94
Table 58. SUBACCOUNT column details	94
Table 59. SUBACCOUNT index details	94
Table 60. SUBACCOUNT_COMMENT column details	95
Table 61. SUBACCOUNT_COMMENT index details	95
Table 62. SUBACCOUNT_LOOKUP column details	95
Table 63. SUBACCOUNT_LOOKUP index details	95
Table 64. SUBACCOUNT_OWNER column details	95
Table 65. SUBACCOUNT_OWNER index details	96
Table 66. TAX_CODE column details	96
Table 67. TAX_CODE index details	98
Table 68. TEXT column details	98
Table 69. TEXT index details	98
Table 70. Naming restrictions in the Easysoft system	148





# Welcome

## Scope of document

The primary functions of this manual are to describe:

- how to install Easysoft ODBC for CODA on your server and PC
- how to manage data sources and drivers
- how to read and write CODA data using Easysoft ODBC for CODA
- the Easysoft Excel Macro for CODA
- the relationship between Easysoft names and CODA names so you can write specific procedures using ODBC function calls to access CODA data
- the background to ODBC and the Easysoft system

## Purpose of Software

Easysoft ODBC for CODA is used to convert queries generated by Windows applications into a form that is suitable for accessing CODA data (which is stored in RMS files on the OpenVMS operating system).

The software described in this manual is designed for versions of CODA that run under the OpenVMS operating system; it cannot be used to access CODA data in other environments.

## Audience

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

- install Easysoft ODBC for CODA on your server. You do not need to know details of ODBC, but you should know both the basics of how CODA operates and also you must have sufficient privileges to install the software
- install the Easysoft ODBC driver on the PC. You do not need to know about ODBC, but you should know the basics of Windows

- using Easysoft ODBC, set up one or more data sources which can be used to access CODA data
- use Windows applications to access CODA data
- write code to access CODA data using ODBC function calls

## Document Structure

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

The installation of Easysoft ODBC for CODA on the server platform is described in chapter two, and following this, the Easysoft Host Administrator functions are defined.

Chapters four and five deal respectively with the installation of the Easysoft ODBC driver on the PC platform and the setting up of a data source.

Chapter six describes the columns and indexes on the CODA tables which are supported by the Easysoft software. Chapter seven is a tutorial which explains how to use an ODBC-compliant application to access CODA data on the server. Chapter eight describes the Easysoft Excel Macro for CODA which can be used to automate many common upload and download operations.

Finally, chapter nine 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”.

The word CODA is used generically throughout this manual to include both CODA-IAS and CODA-Financials.

## 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  button. If two buttons have the same underlined character, for example, **C**opy and **C**lose, toggle between items by repeated pressing of the alt+<character> combination.



# Introduction

**CODA** is a financial accounting system which provides facilities for input and reporting of financial information. Many users of CODA wish to access the data using other software packages, and this is where Easysoft enters the picture. Easysoft ODBC for CODA is an interface between CODA data and Windows applications (which can be used to access CODA data). For example, you can use Microsoft Access to input data to a CODA file containing sales data.

This chapter gives background information to Easysoft ODBC for CODA. The first major section deals with ODBC, the second with the Easysoft system itself, the third major section presents an overview of accounting, and finally, CODA is briefly explained.

## *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.

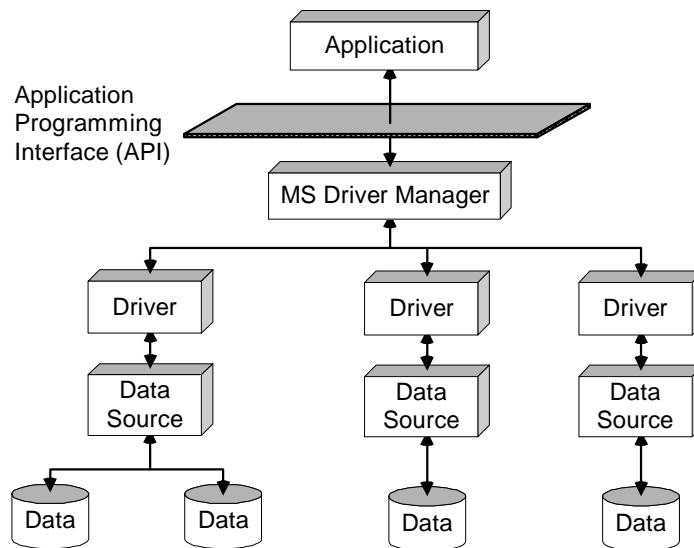
Successful operation of ODBC depends upon defining a set of standards common to all participating elements of a system; this is briefly discussed in the section entitled "Conformance" on page 4.

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

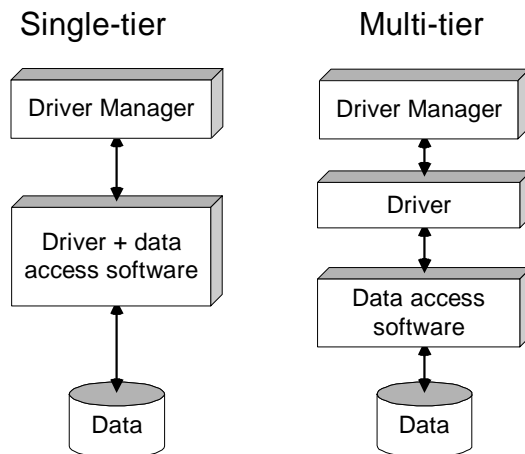
log file of these. Another function of the Driver Manager is to maintain a list of the drivers used by data sources.

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.

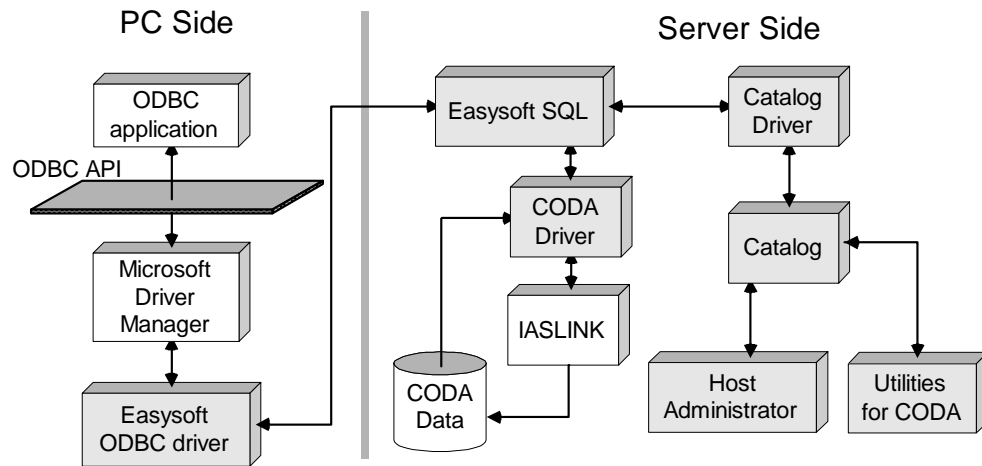


## The Easysoft System

This section first presents the global architecture of Easysoft ODBC for CODA, then we look at the relationship between data sources and catalogs and finally the overall procedure for using Easysoft software is described.

### Easysoft ODBC for CODA Architecture

Easysoft ODBC for CODA is used to connect ODBC-compliant applications (such as Lotus 1-2-3 and Microsoft Excel) to CODA data files which reside on remote file servers (on the OpenVMS platform). It has a multi-tier architecture in that the data access software resides on the server and the driver passes function calls to this. Figure 3 shows the logical architecture of the components of the system and their relationship within the overall ODBC architecture.



**Figure 3. Easysoft software architecture for CODA**

**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 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.

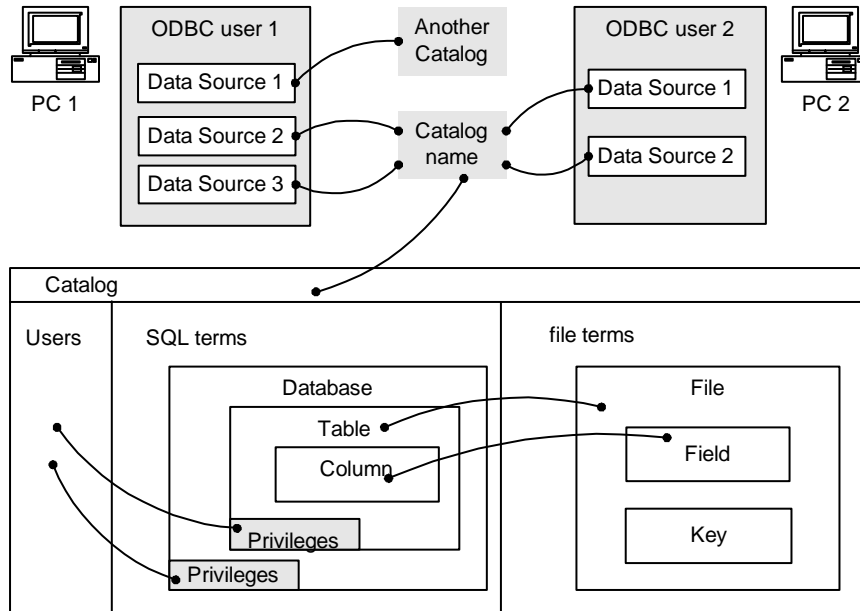
**CODA Driver** - used to access CODA data on the server.

**CODA Utilities** - used as an aid to accessing data, for example, ordering data using the Codaxref routine.

**IASLINK.** This is not part of the Easysoft software. It is part of the CODA system. It is an API, the purpose of which is to allow user programs to access CODA data without using the IAS interface. Easysoft ODBC for CODA uses IASLINK when updating CODA data, and thus there is no possibility of inadvertently corrupting the data. If IASLINK permits an operation, then that is a valid operation for the user to perform in CODA. If IASLINK does not permit an operation, then the user would not be able to perform the operation within CODA-IAS.

## 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.

All the mappings that are needed so that CODA data can be accessed in terms of SQL are carried out entirely automatically by Easysoft ODBC for CODA.

## **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. Part of this includes the CODA setup routine which inserts information into the catalog.
2. License the software.
3. Install the Easysoft Client Component on the PC and then define one or more data sources associated with a catalog.
4. Use an application to access the data.

### **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. The Easysoft Server Component installation creates a catalog for CODA and then populates it.

### **Define a Data Source**

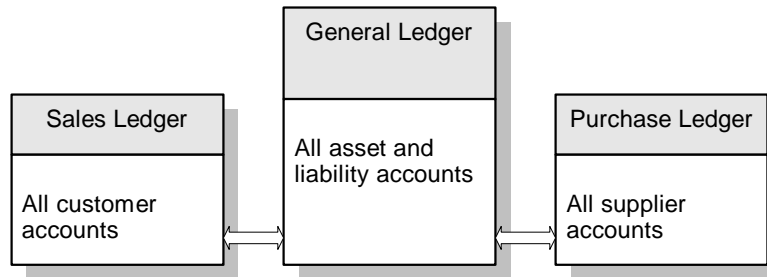
Now that a catalog exists, the Microsoft ODBC Administrator on the PC is used to configure data sources. A unique name is specified for a data source and then the location of the catalog associated with this data source is specified. (At this stage the transport protocol, server name, remote service must also be specified; passwords can either be specified at this stage, or later). Now that a data source has been defined it is possible to access the CODA data by ODBC-compliant applications.

## Accounting Overview

This section is designed for people with no accounting knowledge. Its purpose is to give them sufficient knowledge to understand the basic structure of CODA.

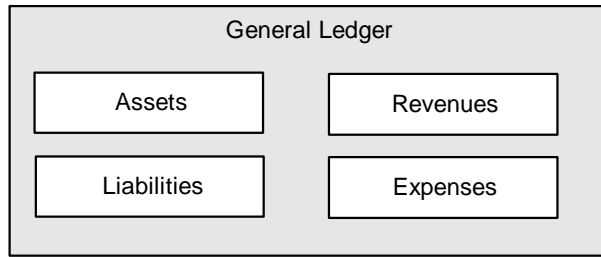
To maintain accurate financial records, we need to group related monetary transactions together. To do this, we set up a framework of individual **accounts**. Each account is given an account name to remind us what it contains, and an account number for ease of reference / inputting. In short, an account is a set of details of financial transactions on items (for example, cars, shops, capital, assets).

A **ledger** is a group of accounts; typically in accounting there are three ledgers, namely sales, purchase and general (also known as nominal), and accounts are associated with an appropriate ledger. The **general ledger** contains all the accounts that are needed to prepare a financial statement. The **purchase ledger** is a ledger for suppliers' accounts (it can be used to produce a list of balances owing to suppliers), and similarly, the **sales ledger** is a ledger for customers' accounts (it can be viewed as a list of balances which are owed by customers). This is visualised in Figure 5.



**Figure 5. The Three Ledgers**

Often it is helpful to subdivide the general ledger (that is, categorise and group financial transactions). Figure 6 is a simple example of this.

**Figure 6.**

**Structure of the general ledger**

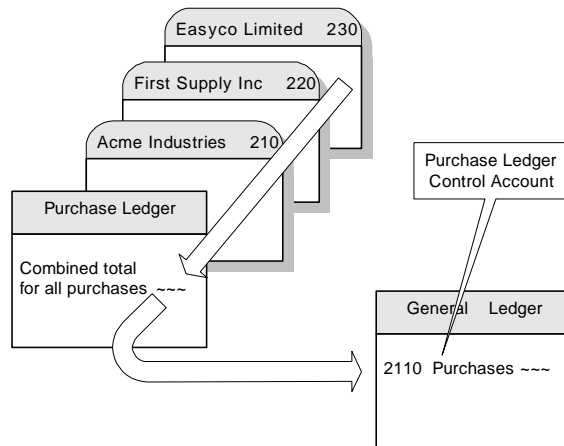
The **chart of accounts** defines the accounting structure of a business. It is a list of accounts (names and numbers) in the general ledger. It can be viewed as an index to the general ledger.

The chart of accounts shown below relates the categories of the general ledger shown in Figure 6 to the various accounts in the ledger. The numbering system associated with ledgers and accounts is designed so that all accounts of a given type have a similar initial digit. For example, in this case all the expense accounts start with the digit '4'.

#### Chart of Accounts

(1000) ASSETS	(3000) REVENUE	
<u>1100</u> <u>Current Assets</u>	3110 Cash sales	} categories
1110 Cash	3120 Royalties	
1120 Accounts receivable		
1130 Supplies	(4000) EXPENSES	} accounts
1140 Insurance paid	4110 Rent	
<u>1200</u> <u>Plant Assets</u>	4120 Advertising	
1210 Equipment	4130 Miscellaneous	
(2000) LIABILITIES		
2110 Purchases		
2120 Sales tax payable		

The Purchases account (2110) in the general ledger is a special kind of account, called a **(ledger) control account**. A control account summarises the financial content of a sales or purchase ledger (that is, it gives a balance at a point in time). Figure 7 shows the situation in pictorial form.



**Figure 7.**  
**Relationship**  
**between control**  
**accounts and**  
**ledgers**

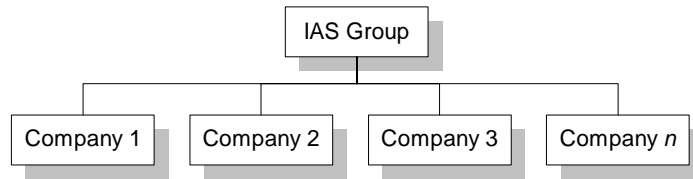
Finally, let's look at the dynamics of the system. **Books of original entry** (often called **journals**) are used to make the initial recording of account transactions in chronological order. Information in the journals is then **posted** (transferred) to the ledger accounts. Financial reports can then be produced using the information in the general ledger accounts.

This brief overview allows us to understand the principles upon which CODA operates; the next section describes CODA and relates its logical structure to the accounting concepts presented above.

## ***CODA Overview***

This section outlines the structure of data used in the CODA system. It introduces the concepts that are necessary to enable Easysoft to be used, and should not be taken as a full description.

In CODA terms a **company** is a separate accounting entity which may be a real company or a part of a company, for example, a division of a corporation. CODA also define a company as "a set of accounts that balance to zero". A company equates to the general (or nominal) ledger in accounting terms. One or more companies can exist within an **IAS group** (Figure 8). This means a company may share, if appropriate, characteristics of another company within that group. The first company that is set up after installation is the master from which shared details are taken.

**Figure 8.****CODA Group Architecture**

Other structures are possible, for example, the companies (accounting entities) could be separated into two or more IAS groups. The appropriate organisation depends on how the data is to be used; this issue is beyond the scope of this manual.

Easysoft ODBC for CODA is applicable to the Company level and below.

## Company Structure

CODA defines four types of ledger which are subdivisions of the general (nominal) ledger: sales, purchase, expense and fixed asset (not always available). Every ledger has a **Ledger name** which is a 30 character field used to describe the ledger. The **Ledger identifier** is a five character field; the first character is a letter (S, P, E or F) which identifies the ledger type, and the remaining four characters are numbers.

### Sales type ledger (S)

Groups accounts that are related to customers. There may be more than one sales ledger, but any customer is associated with only one sales ledger.

### Purchase type ledger (P)

Groups accounts that are related to suppliers. There may be more than one purchase type ledger, but any supplier is associated with only one purchase ledger.

### Expense type ledger (E)

Groups accounts that are related to income and expenditure.

### Fixed Asset type ledger (F)

If this is available, it is used to subanalyse nominal accounts (that is, the top level of the hierarchy) to subaccount level.

Note: for convenience we drop the term “type” when referring to ledger types, and simply refer to them as ledgers. The context should make clear whether we are referring to a ledger type or a specific named instance of a ledger type.

Each CODA company has a chart of accounts which defines the accounting structure for that company. This is a three level hierarchy, unlike a chart of accounts in



general accounting terms which has no theoretical limit to the subdivisions of the general ledger. The highest level of the CODA chart of accounts is called the **nominal** level, and accounts defined at this level are called **Nominals**.

It is possible, but not necessary, to have sub-accounts defined below the nominal level. An account at the second level is called a **Subaccount**, and an account at the third level is called a **Level3**. Accounts at these three levels are linked by **Ledger links**. Ledger links define which accounts each Nominal, Subaccount and Level3 can access. If a Subaccount or a Level3 exists, then it must have a Ledger Link which points back up the chart of accounts.

There are restrictions to the structure of the chart of accounts:

- Not all ledgers can be decomposed to the third level; sales and purchase ledgers can exist only at the nominal level, and subaccounts must be of a type called **names and addresses** (these hold information necessary for the operation of sales and purchase accounts).
- Every sales and purchase ledger requires a control account.

In addition to using a chart of accounts, it is possible to group unrelated entities at any given level by means of **Hierarchical Account Groups** (also known as **Account Groups**) This grouping is only used when data is to be read, not when it should be updated.

To use CODA other information not directly associated with a chart of accounts is also needed, and this is discussed next.

## Using CODA

Before CODA can be used, it must be installed and initialised. The initialisation process includes the creation of a company, and immediately following this, two users are given different access rights to the company. One is a consultant user with a CODA security level of 9 (the maximum), and the other is a normal user, with a security level between 1 and 8. The specific rights associated with the numbers 1 to 8 are defined by the organisation that uses CODA. The System Manager / Consultant user has total privileges for the system. The accounting structure is then defined, along with the static data which is needed to run the system. Finally, the system is licensed.

The data contained in CODA can be split into three areas: **Masters**, Transactions and Period Balances. Masters are the static master file records within which the system operates. They are briefly described here.

### Account Groups

These are used as an additional level of analysis in the CODA accounting structure. An account can belong to one or more account groups.

### Analysis Titles

A method of analysing Name and Address information. Used with IASLINK and Credit Management.

### Currency Masters

Contain details of foreign currency and exchange rates.

### Document masters

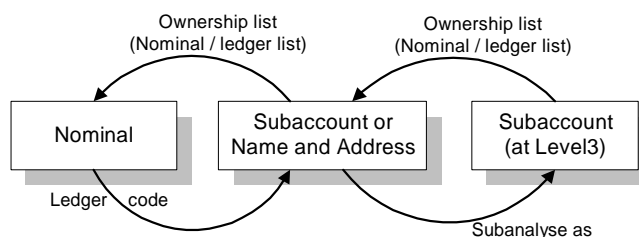
Used to set up the structure of the documents which are used to post (enter) transactions into CODA. For example, sales invoices. All CODA documents have a **document header**, known as “**line 1**”, and one or more lines (**detail lines**, also called **underneath lines**) containing other information.

### Fixed assets

Used for the accounting of Fixed Assets. This is an optional feature.

### Ledgers (links)

Ledgers are used to group accounts. Figure 9 shows this. The Nominal/ledger list is also known as an ownership list. This is used to relate an account to an account or ledger in the level above. If a ledger code is specified in a Nominal, this means that the nominal can be subanalysed. Similarly, if a Subaccount or Name and Address has a values specified for the Subanalyse field, then it can be subanalysed.



**Figure 9.**

### Ledger links

### Names and Addresses

These are a special type of subaccount for Sales and Purchase ledgers. In addition to

information needed purely for the financial aspects of accounting, these accounts also hold information needed in the use of sales and purchase ledger accounts.

**Media Headers**

For each possible method of payment (cheque, cash, BACS), a media header contains information about the method of payment.

**Nominals**

The highest account level in the chart of accounts in CODA. Linked to subaccounts by ledger links.

**Subaccounts**

There are two types of subaccount in the CODA chart of accounts, namely subaccount and level3. Subaccount is the second level in the chart of accounts, and level3 is the third (bottom) level. Very often, the terms subaccount and level3 are used synonymously in CODA documentation.

**Tax codes**

Tax codes are used to hold rates of tax which are accessed by CODA using an identification code.

**External Access to CODA Data**

IASLINK, which is an Application Programming Interface (API), is part of the CODA software. Its purpose is to allow user programs (linked with the IASLINK library) to access CODA account data without using the IAS interface. Furthermore, it is not necessary to know the underlying IAS file structure.

This is achieved by defining the format of data blocks and fields within each block in the data files. There are two defining files: LNKARGBLK.TXT can be included directly in COBOL programs, and LNKARGBLK.H can be included directly in C programs. They are both located in the directory pointed to by the IAS\$TEXT logical on the server.

Note that the files which are defined are logical - in practice a single file name may be associated with several physical files. Users of IASLINK are completely separated from the physical file structure, and hence do not need to know of any changes which CODA may make.

IASLINK names are included in the CODA tables described in chapter 6.



## CHAPTER 2

# Installation and Licensing on the Server

This first parts of this chapter explain how to install Easysoft ODBC for CODA on the server platform and how to license the software. The Maintenance section explains how to re-license the software, change catalog passwords, add CODA companies to the catalog, change CODA usernames, passwords and privileges in the catalog, and how to order BUDGET data.

<b>Steps for New Installations and Upgrades</b>	<b>New</b>	<b>Upgrade</b>
<b>Preparation</b>		
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. Special Instructions for Upgrade - v1.1.6 and Below	-	✓ (v1.1.6)
6. Transfer zipped files to server	✓	✓
7. Unzip files	✓	✓
<b>Installation</b>		
8. Install Easysoft Server Component	✓	✓
9. Run CODA setup routine to:	✓	✓
Create catalog		(special
Set up CODA logicals		cases
Create catalog contents (run Codacat)		only)
Input passwords to catalog (run Catuser and Codauser)		
Optimisation (run Codaxref)		
10. Set quotas	✓	-
11. Edit system startup file	✓	-
<b>Licensing</b>		
12. Generate site number	✓	-
13. Obtain Licence Key	✓	✓
14. License product	✓	✓
15. Check Installation	✓	✓

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.

CODA versions 6.4 to 8.0 inclusive are supported (CODA-IAS and CODA-FINANCIALS).

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 50).

## Preparation

### Step 1. Log on to Server

**NEW**

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.

**UPGRADE**

### 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 CODA 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 8) 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 8); 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 6 for an example)
2. Unzip the file (see Step 7 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. Special Instructions for Upgrade - v1.1.6 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 CODA 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 6. Transfer Files from Floppy Disk to Server

**NEW**

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.

**UPGRADE**

**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 7. Unzip Files

**NEW**

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

**UPGRADE**

```
$ 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 8. Install Easysoft Server Component

**NEW**

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

**UPGRADE**

```
$ 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 CODAVAX014 and for Alpha (AXP) it is CODAAXP014.

**<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. This applies to VAX installations only. 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 CODA 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 CODA

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.

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. It then calls the CODA setup routine (step 9).

### 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
[.CODA]	Default Easysoft Catalog for CODA	EASYSOFT_SQL_CODA_CATALOG
[.DOC]	On-line documentation	-
[.LICENCE]	Licence information	-
[.LOG]	Log files	-
[.SYSTEM]	System files	EASYSOFT_SQL_SYSTEM
[.TEMP]	Temporary files	EASYSOFT_SQL_TEMP

**Note:** If the CODA IAS\$SCRATCH logical is not defined, then when you connect to a CODA data source using Easysoft ODBC the logical is assigned to the EASYSOFT\_SQL\_TEMP logical.

## Step 9. Run CODA Setup Routine

**NEW**

The purpose of the Easysoft ODBC for CODA setup routine is to automate the processes that are required for the completion of the installation. In the general case, the CODA setup routine should be used for a new installation in all cases. Whether or not it is used for upgrades depends upon the conditions described below.

**UPGRADE**

The routine does the following:

1. Creates an empty catalog (see “CREATE CATALOG”, chapter 3).
2. Sets up CODA logicals (by running IAS.COM).
3. Runs the CODACAT routine - this populates and/or updates the Easysoft Catalog (see “Adding CODA Companies”, page 37).
4. Runs the CATUSER and CODAUSER routines. CATUSER changes the password of catalog users (see “Changing Catalog Passwords”, page 35). CODAUSER changes the password of CODA users (see “Changing CODA Usernames, Passwords and Privileges”, page 39).
5. Runs the CODAXREF routine. This is an optimisation step (see “Ordering BALANCE and BUDGET Data”, page 42).

These processes can also be completed manually, and some of them are used for maintenance purposes.

**UPGRADE  
WARNING**

The creation of an empty catalog (setup routine step 1) will overwrite any previous catalog information (assuming that the old catalog is in the default location). This means that all the catalog and CODA passwords for all CODA users will have to be re-entered. The trade-off is that if there are many new CODA companies and/or new or redundant CODA users, then the time taken to update this information in the catalog using the semi-automatic CODA setup routine can be considerably less than the manual equivalent. You have to decide what your best option is.

### Setup Routine Step 1

```
<Easysoft ODBC for CODA setup routine introductory message>
```

```
Do you wish to automatically setup Easysoft ODBC for CODA ?  
(Y/N)[Y]:
```

If you select **n**, then the routine does not run. You should run the appropriate individual routines listed in steps 1 to 5 above.

If you select **y**, Step 1 of the CODA setup routine is presented. This creates a catalog.

If you are upgrading, and assuming that the existing catalog is in the directory pointed to by the EASYSOFT\_SQL\_CODA\_CATALOG logical, you are presented with a message warning you of this. You have the option of continuing or of terminating the CODA setup routine.

```
<step 1 introductory message>
Easysoft Administrator <version and copyright information>
Creating catalog in the directory EASYSOFT_SQL_CODA_CATALOG
<list of catalog tables being created>
Catalog created successfully.
```

## Setup Routine Step 2

This checks for the existence of the IAS\_TEXT, IAS\_GROUP and IAS\_SYSTEM logicals. If they exist, you are offered the option of using them. If you choose not to, or if they do not exist, then you should enter the name of the directory which contains IAS.COM.

```
Enter the directory containing IAS.COM ? : <device><directory>
```

IAS.COM is run, with the result that various CODA logicals are set up.

## Setup Routine Step 3

This runs the Codacat routine (see “Adding CODA Companies”, page 37). If there are many CODA companies, this step may take a long time; to proceed rapidly, you can select just a single company (this is the option shown in the example below), and later, you can add more companies using the Codacat routine.

```
<step 3 introductory message>
  1) Run CODACAT for all companies
  2) Run CODACAT for a specific company
Enter required option (1,2)[1]: 2

Enter required company: <name of company to add to catalog>
<Easysoft CODA Catalog Generator copyright information>
Catalog Directory : EASYSOFT_SQL_CODA_CATALOG
Admin Password   : <catalog administrator's password>
CODA version : <version number appears here>
Updating Catalog
  Writing : <company name appears here> .....
```



## Setup Routine Step 4

This first runs the Catuser routine, and then runs the Codauser routine.

When a CODA user's details are added to the catalog (previous step), the user's password for CODA is set the same as the username. Each user has a name and a password which gives access to the Easysoft Catalog; both of these are also set to the CODA username. Step 4 changes these passwords (if a CODA user has a different password for different companies, then the passwords cannot be changed automatically, and you must run the Codauser routine manually ("Changing CODA Usernames, Passwords and Privileges", page 39) to change each of the passwords).

```
<informational message>
```

```
Enter CODA username: <CODA username>
```

```
Enter CODA password: <CODA password for CODA username above>
```

```
Easysoft Catalog Password Changer <version and copyright information>
```

```
Catalog password successfully updated for user <CODA username>
```

The output shown above comes from the Catuser routine; the output below relates to the Codauser routine.

```
Easysoft Coda Catalog Password Changer <version and copyright information>
```

```
Update successful for catalog user <Coda username> with company <company name>
```

```
Enter CODA username:
```

Entering a blank username causes the setup routine to move to the final step.

## Setup Routine Step 5

This runs the Codaxref routine, which is an optimisation process. You must run this before you can use the ACCOUNT and BUDGET tables. The companies that are available for processing with this routine are those that were added during step 3 of the CODA setup routine. Codaxref should be run whenever Nominals, Subaccounts or Level3s are added or deleted to CODA data. This step offers the option of scheduling a batch job to be run overnight which automatically runs Codaxref.

```
<step 5 introductory message>
```

- 1) Run CODAXREF for all companies now
- 2) Run CODAXREF for a specific company now
- 3) Do not run CODAXREF now

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

```

Easysoft CODA Cross Reference <version and copyright information>
Processing files in directory <device>:<directory>
<informational messages appear here>
Do you wish to schedule CODAXREF to run as a batch job overnight ?
(Y/N)[Y]:

```

If you enter **y**, a batch job is submitted which begins at midnight and which is repeated every night at midnight . If you want to change any of the batch job defaults, then you should modify the CODAXREF.COM file (see page 151).

If you enter **n**, the CODAXREF.COM file is created, but it is not submitted (and therefore, the batch job will not run).

```

<batch job information>
Press return to continue:
Easysoft ODBC for CODA server setup completed successfully
<informational message stating that software must now be licensed>
Press return to continue:

```

Finally, information about installing the Easysoft ODBC driver and creating a data source on the PC is presented (it is labelled as step 6).

```

<description of what to do on the PC (labelled step 6)>
Press return to continue:
<message stating installation is complete>
$

```

### Note: Coda Setup Routine (manual usage)

By default, the EASYSOFT\_SQL\_CODA\_CATALOG logical points to the directory [EASYSOFT.SQL.CODA.CATALOG]. If you want to use a catalog in a different location, then redefine the logical and re-run the CODA setup routine.

```

$ DEFINE EASYSOFT_SQL_CODA_CATALOG <DEVICE><DIRECTORY>
$ @EASYSOFT_SQL_SYSTEM:SETUP_CODA

```

## Step 10. 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 11. 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 SYSS\$MANAGER:SYSTARTUP\_V5.COM. For OpenVMS version 6 the file is named SYSS\$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):

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

**Note:** The directory SYSS\$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 25).

## 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 12. 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>
```

*These lines are generated  
by Easysoft software.*

```
Site number      : <unique site number appears here>
```

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

```
<Easysoft address details>
```

## Step 13. Obtain Licence Key

**NEW**

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).

**UPGRADE**

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 14. License Product

**NEW**

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

```
$ RUN EASYSOFT_SQL_ADMIN
```

**UPGRADE**

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
$
```

## Step 15. Check Installation

**NEW**

- To check that the installation is correct, use Easysoft SQL to access CODA data. For example:

**UPGRADE**

```
$ SQL:==$EASYSOFT_SQL_EASYSQL
$ SQL EASYSOFT_SQL_CODA_CATALOG <CODA user name> <CODA password>
```

```
Easysoft Interactive SQL Version <n>
<copyright message>
```

```
EASYSQL> SELECT * FROM <company>_LEDGER;
<data appears here>
```

The installation is complete, and you should now install Easysoft ODBC on the PC platform.

## ***Maintenance***

This section describes routines that are used in the maintenance of the Easysoft catalog.

### **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 turn off this message use the [SETTINGS] flag in the EASYSOFT.INI file (see Appendix B).

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 14 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 : EASYSOFT_SQL_CODA_CATALOG
Catalog User      : MIKE
Catalog Password  :
New Password      :
Verify Password   :
```

Catalog password successfully updated for user MIKE

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_CODA_CATALOG
Catalog User      : ADMIN
Catalog Password :
Administrator Options:
    1) Change Administrator Password
    2) Change User Password
    0) Exit
Enter option number [1, 2 or 0] - 2
Catalog User      : DEMO
New Password      :
Verify Password   :
Catalog password successfully updated for user DEMO
Catalog User      :

```

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

*Name and new catalog password of non-administrator 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.

## Adding CODA Companies

The Codacat routine is used to add CODA companies and associated information to the Easysoft Catalog. This command does not *create* a company for CODA; it *adds* an existing CODA company to the Easysoft Catalog. The catalog must already exist - a catalog is created using the Easysoft Host Administrator CREATE CATALOG command (see chapter 3 and "Setup Routine Step 1", page 27).

```
$ RUN EASYSOFT_SQL_SYSTEM:CODACAT
<informational messages>
Catalog Directory : EASYSOFT_SQL_CODA_CATALOG
Admin Password   :
```

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.

Only catalog administrators can use this routine; enter the Admin Password. If the specified password does not exist, an error message is generated. Admin Password equates to the Catalog Login Password (for the ADMIN user) in the Easysoft ODBC Setup dialog box (see Appendix A).

Following a message indicating the CODA version, a list of CODA companies is presented.

```
CODA version : <version number>
Companies within group IAS$GROUP :
  DEMO      *NEWDEM    OLDDEMO    TEST      * (ALL)
Select Company : NEWDEM
Updating Catalog
  Writing : NEWDEM .....
Select Company :
$
```

Quit the routine at any prompt by entering a blank line.

Enter one or more companies at the Select Company: prompt. If you enter a list of companies, they must be separated either by a comma and/or a space. To select all the companies use the asterisk character (\*).

If there is a CODA company called ES, then the Codacat routine recognises this as a special case (since ES is reserved for Easysoft data) and asks you for an alternative name. This alternative must not be ES, or any company name that already exists.

The asterisk character (\*) can be used as a wildcard, representing one or more characters; it can be placed either at the start or the end of a word, and there can only be one wildcard in a company name. For example, entering \***DEMO** at the `Select Company` prompt would result in `DEMO` and `OLDDEMO` being added to the catalog.

The asterisk character (\*) preceding a company (for example, \***NEWDEM**) indicates that this company already exists in the catalog. If you try to add this to the company, you are presented with the following prompt:

```
Select Company : NEWDEM  
NEWDEM exists in catalog. Overwrite, Delete or Refresh users? (O/D/R)  
:
```

The Overwrite option clears all existing definitions for the company, and enters new definitions into the catalog.

The Delete option removes all information about the specified company from the catalog.

The Refresh users option results in the addition of new CODA users to the catalog (that is, if a CODA user is not already in the catalog, then the user is added) and the deletion from the catalog of users who are no longer valid CODA users. If there are any existing users already in the catalog, their names and passwords are not added to the catalog (see next section for information about names and passwords in the catalog).

If a new company is added to the catalog, and if a user of the company already exists within the catalog the catalog password will not be changed.

## Users and Passwords

The Codacat routine does not store CODA passwords in the catalog. Instead, for each CODA user, the password is set to the username. To make the password which is stored in the catalog the same as the CODA password, use the Codauser routine described in the next section.

Additionally, each user is given a Catalog Login Username and Password which is used to control access to the catalog. These are both set to the CODA username. To change the password of the catalog user, use the Catuser routine (see “Changing Catalog Passwords”, page 35).

For example, say there is a CODA user called MIKE with a password MIKEPWD. The Codacat routine stores the following information in the catalog:

CODA user:	MIKE	—	<i>Change using the Codauser routine.</i>
CODA password:	MIKE	—	
Catalog (Login) user:	MIKE	—	<i>Change using the Catuser routine.</i>
Catalog (Login) password:	MIKE	—	

To access CODA data, either the CODA password stored in the catalog must be changed to MIKEPWD using the Codauser routine (see next section) or the CODA password itself must be changed (to MIKE) using CODA.

Once a Catalog user has been added to the catalog, the username cannot be changed.

## Changing CODA Usernames, Passwords and Privileges

The Codauser routine is used to change the names and/or passwords of existing CODA users and to change access privileges to CODA companies. These changes are within the Easysoft Catalog only; no changes are made to CODA data.. This latter function is only available to the catalog administrator and is explained in the next section. The Codauser routine resides in the directory pointed to by the EASYSOFT\_SQL\_SYSTEM logical.

```
$ RUN EASYSOFT_SQL_SYSTEM:CODAUSER
<informational message>
Catalog Directory      : EASYSOFT_SQL_CODA_CATALOG
Catalog User          : STEVE
Catalog Password      :
```

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 CODA name or password you wish to change. If the specified user does not exist in the catalog, an error message is generated: Catalog user <name> does not exist. Catalog User equates to Catalog Login Username in the Easysoft ODBC Setup dialog box (see Appendix A).

Type the Catalog Password for the user. If the password is incorrect, an error message is generated: `Catalog Password for catalog user <name> is invalid . Catalog Password equates to Catalog Login Password in the Easysoft ODBC Setup dialog box (see Appendix A).`

Since each CODA user may have access to many different companies, a list of available companies is presented. You can change the password either for all the companies at once, in which case all the passwords will be identical, or you can change the password for one or more companies separately, in which case the password for the user can be different for different companies.

```
Companies available to catalog user STEVE :
NEWDEM * (ALL)

Company                : NEWDEM
Coda User              : <enter CODA user here>
New Password          :
Verify Password       :

Update successful for catalog user STEVE with company NEWDEM
Company                :
```

Enter the CODA User. If the user does not already exist, then a user with the new name is created and linked to the catalog user.

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 re-appears.

## Changing CODA Privileges

The catalog administrator can change CODA access privileges for all users. This is an additional restriction on CODA security, it does not over-ride the security set in CODA-IAS. (Additionally, if you change a user's security level from within CODA-IAS, say from 4 to 6), this change is effective immediately with respect to Easysoft ODBC for CODA).

```
Catalog Directory      : EASYSOFT_SQL_CODA_CATALOG
Catalog User          : ADMIN
Catalog Password      : <administrator password>
Administrator Options:
```

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

```
1) Change Coda Password
2) Change Coda company privileges
0) Exit

Enter option number [1, 2 or 0] - 2
Catalog User (*=ALL) : STEVE
Companies available to catalog user STEVE :
    DEMO          NEWDEM          OLDDEMO          * (ALL)
Company          : NEWDEM
Privilege Options:
1) Read Only
2) Full Access
0) Exit

Enter option number [1, 2 or 0] - 1
Privileges changed for catalog user STEVE within company NEWDEM

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 access privileges to CODA companies. The options are:

```
Read Only 1
Full Access 2
Exit      0
```

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

## Ordering BALANCE and BUDGET Data

The BUDGET and BALANCE tables return data in Nominal/Subaccount/Level3 order by using a special table called ACCOUNT which is generated by running the Codaxref routine. This can take from a few minutes to a few hours. The time taken depends not only on the processor speed of the server, and the priority allocated to the process, but also on the size of the data files. Although this is purely an optimisation process for reading data in N/S/L3 order, and does not in any way affect the CODA data, you must run the routine before you will be able to see any BALANCE and BUDGET data (a default batch job to run Codaxref each night is set up during the installation of Easysoft ODBC for CODA on the server platform. For information about the file (CODAXREF.COM) that controls the batch job refer to "CODAXREF.COM", page 151).

Take the following steps to run Codaxref manually. This must be done for each company in the CODA group for which you wish to use the Easysoft software.

```
$ RUN EASYSOFT_SQL_SYSTEM:CODAXREF
<copyright message>
Catalog Directory : EASYSOFT_SQL_CODA_CATALOG
Catalog User      : STEVE
Catalog Password  :
```

Enter the location of the catalog for which you want to generate the cross-reference information. Catalog user should be one that is also a CODA user who has privileges on all the companies for which you are generating the cross-reference information.

```
Companies available to user STEVE :
EASYSOFT * (ALL)
Company      : *

Processing files in directory <device><directory>
<informational messages appear here>
Company      :
```

Whenever Nominals, Subaccounts or Level3s are added or deleted, Codaxref should be run.

If a new N/S/L3 combination is inserted through the BALANCE or BUDGET table, that combination is automatically written to the ACCOUNT table. If *only* Easysoft ODBC for CODA is used to insert Budget data using these two tables, the Codaxref routine need not be run, apart from the initial installation. However, if you use any other tables to change Budget data, or if you use CODA itself to make the changes, then Codaxref should be run as usual.

## 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.

Some of the Host Administrator commands make reference to “company”. In all cases, this refers to the organisation which has purchased the Easysoft software, and has no reference to CODA companies.

## 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. In the general case it is not used with Easysoft ODBC for CODA. 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
```

Username 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 10).



Figure 10.

**ODBC Setup  
(Welcome) dialog  
box**

Click Next to continue. The Software License Agreement dialog box is displayed (Figure 11).

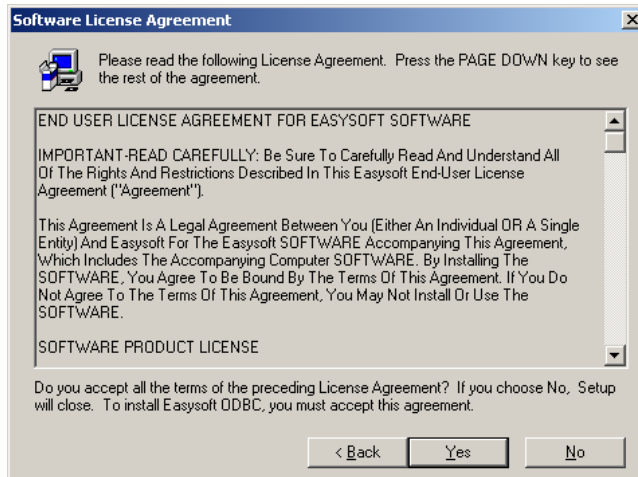
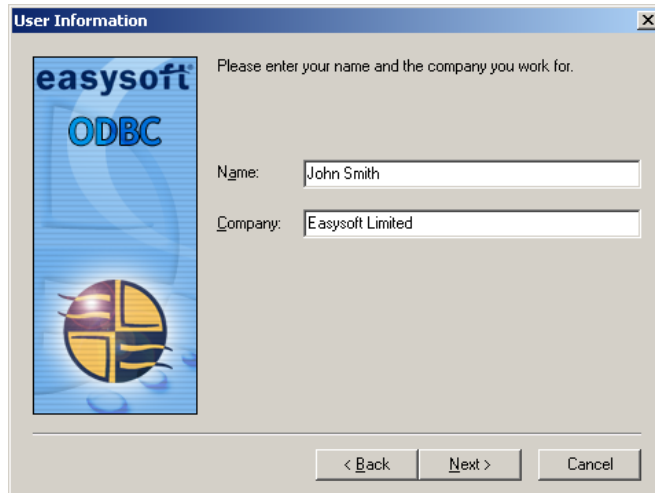


Figure 11.

**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 12**) is displayed.

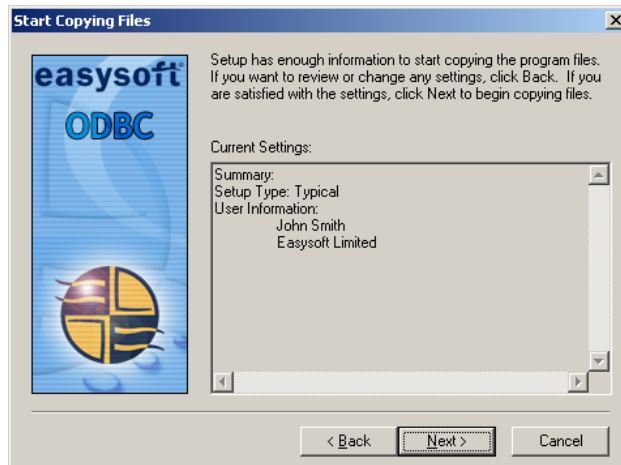


**Figure 12.**

**User Information  
dialog box**

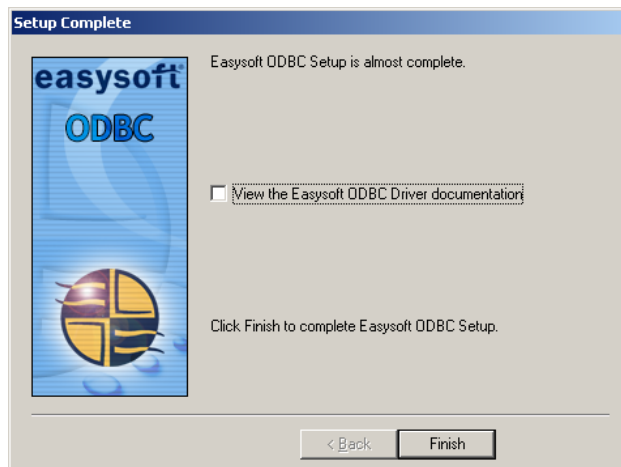
Type your name and company in the spaces provided. Click Next.

In the Start Copying Files dialog box (Figure 13), click Next.



**Figure 13.**  
**Start Copying Files**  
**dialog box**

In the Setup Complete dialog box (**Figure 14**), click Finish to return to Windows. To display the Easysoft ODBC Driver online Help, click the check box.



**Figure 14.**  
**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

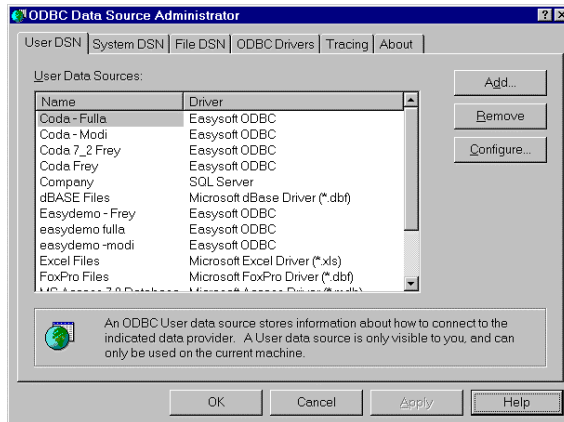
This chapter explains how to configure the data sources you intend to use. A detailed walk-through of most options is shown, the data you need to input to the system is explained, and some of the more common problems encountered are explained. The working example shows how to create a data source called CODA which can be used to access your CODA data. You can have as many different data sources as you want.

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.



**Figure 15.**

**User DSN tab**

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

<b>Tab</b>	<b>Function</b>
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 22). 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 15) on the Microsoft ODBC Data Source Administrator and then click the **Add...** button. The Create New Data Source dialog box (Figure 16) is displayed.

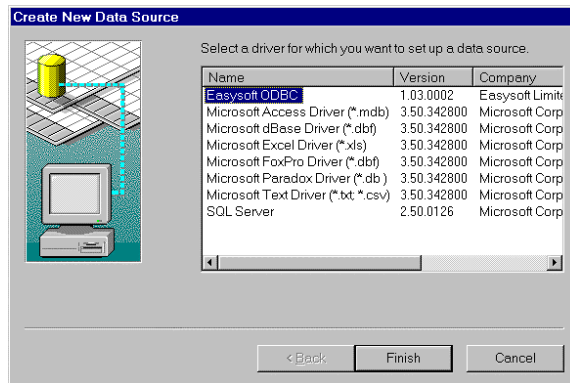


Figure 16.

**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 17) appears.

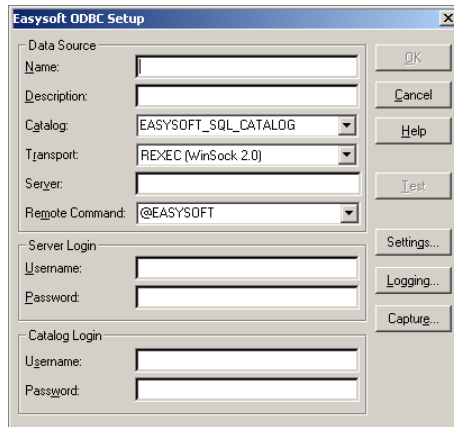


Figure 17.

**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 61 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 62 so you can have a guaranteed working example to test.



**Table 1. Information required for the Easysoft ODBC Setup**

<b>Name</b>	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 70. Naming restrictions in the Easysoft system on page 148 lists invalid characters).								
<b>Description</b>	A more descriptive name for the data source (optional).								
<b>Catalog</b>	<p>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).</p> <p>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.</p> <p>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</p>								
<b>Transport</b>	<p>The network transport to be used. You are presented with the options available to your PC.</p> <table border="0" style="margin-left: 2em;"> <thead> <tr> <th style="text-align: left;"><u>connection method</u></th> <th style="text-align: left;"><u>protocol</u></th> </tr> </thead> <tbody> <tr> <td>REXEC</td> <td>Windows sockets 2.0</td> </tr> <tr> <td>TCP/IP</td> <td>Windows sockets 2.0</td> </tr> <tr> <td>DECNET</td> <td>Pathworks 4.1 upwards</td> </tr> </tbody> </table>	<u>connection method</u>	<u>protocol</u>	REXEC	Windows sockets 2.0	TCP/IP	Windows sockets 2.0	DECNET	Pathworks 4.1 upwards
<u>connection method</u>	<u>protocol</u>								
REXEC	Windows sockets 2.0								
TCP/IP	Windows sockets 2.0								
DECNET	Pathworks 4.1 upwards								
<b>Server</b>	The name of the server on which the source data resides.								
<b>Remote Service / Command / Object</b>	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.								
<b>Server Login Username</b>	The user name used to connect to the server. Case is ignored.								
<b>Server Login Password</b>	The password associated with the server login username. Case is ignored.								
<b>Catalog Login Username</b>	The user name that provides access to the catalog. Case is ignored.								
<b>Catalog Login Password</b>	The password associated with the Catalog login user name. Case is ignored.								

**Table 2. Easysoft Demonstration Defaults**

<b>Name</b>	EASYDEMO
<b>Description</b>	Easysoft demonstration (You can type anything you like)
<b>Catalog</b>	EASYSOFT_SQL_DEMO_CATALOG (Select this from the available options)
<b>Transport</b>	REXEC (Recommended option - select from available options)
<b>Server</b>	The name of the server
<b>Remote Service /Command/Object</b>	Select one of the available options
<b>Server Login Username</b>	Your user name for the server
<b>Server Login Password</b>	Your password for the server
<b>Catalog Login Username</b>	ADMIN This is provided automatically after you select the demonstration catalog
<b>Catalog Login Password</b>	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 18) 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.

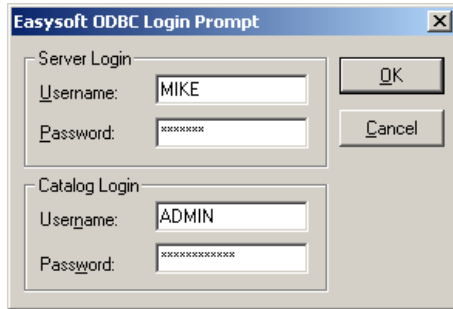


Figure 18.

### 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 19) is accessed by selecting **Settings...** on the Easysoft ODBC Setup dialog box (Figure 17). It is used to set driver, network and SQL options.

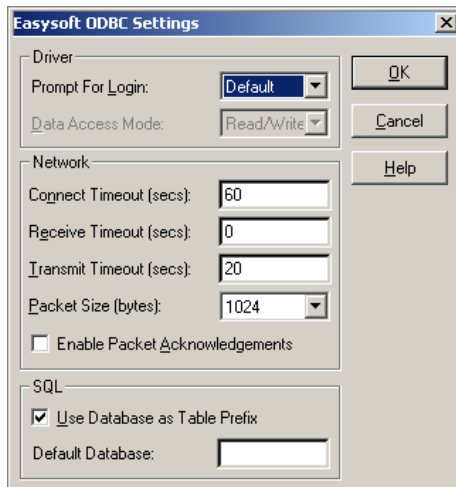


Figure 19.

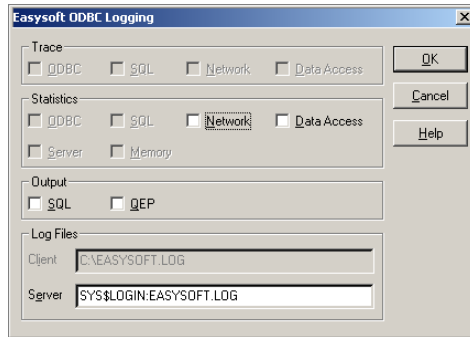
### Easysoft ODBC Settings dialog box

**Table 3. Information required for Easysoft ODBC Settings**

<b>Prompt For Login</b>	Controls the display of the Easysoft ODBC Login Prompt when a connection is made to a data source. <b>Default:</b> display is defined by the application. <b>Never:</b> dialog box is never displayed. For successful operation, ensure that login information is supplied in the Easysoft ODBC Setup dialog box. <b>Always:</b> dialog box is always displayed even if login information is contained in Easysoft ODBC Setup dialog box.
<b>Data Access Mode</b> (not yet supported)	Allows data to be either read/write or read only.
<b>Connect Timeout</b>	The attempt to connect is cancelled if it is not successful within the specified time.
<b>Receive Timeout</b>	The connection is cancelled if a reply is not received within the specified time. <b>Note:</b> the value zero means wait forever.
<b>Transmit Timeout</b>	The connection is cancelled if the network layer cannot send the packet in the specified time.
<b>Note:</b> the timeouts for each option are identical for both client and server.	
<b>Packet Size</b>	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.
<b>Enable Packet Acknowledgements</b>	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.
<b>Use Database as Table Prefix</b>	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.
<b>Default Database</b>	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 20) is accessed by selecting **Logging...** on the Easysoft ODBC Setup dialog box (Figure 17). It is used to control the logging options that are described in Table 4.



**Figure 20.**

**Easysoft ODBC  
Logging dialog box**

**Table 4. Easysoft ODBC Logging Information**

(Only available options are described)

<b>Trace ODBC, Trace Network Network</b>	These can be enabled using the EASYSOFT.INI file. Refer to the [LOGGING] option in “Explanation of Sections”, page 150
<b>Data Access</b>	Statistics about data transmitted and received.
<b>SQL</b>	Lists the files accessed for a given query and the number of I/Os to files for that query.
<b>QEP</b>	Lists the SQL query which is passed to the server.
<b>Log file (Server)</b>	Lists the Query Execution Plan.
	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

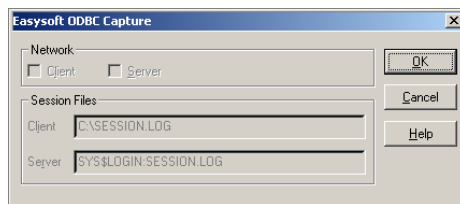
with a long log containing entries for many queries. To aid clarity, delete the Easysoft log file (by default called EASYSOFT.LOG) before sending the query you wish to inspect. After the query has been sent, the log file relates to just that single query.

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 182.

## Capture Information

The Easysoft ODBC Capture dialog box (Figure 21) is accessed by selecting **Capture...** on the Easysoft ODBC Setup dialog box (Figure 17). 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.



**Figure 21.**

**Easysoft ODBC  
Capture dialog box**

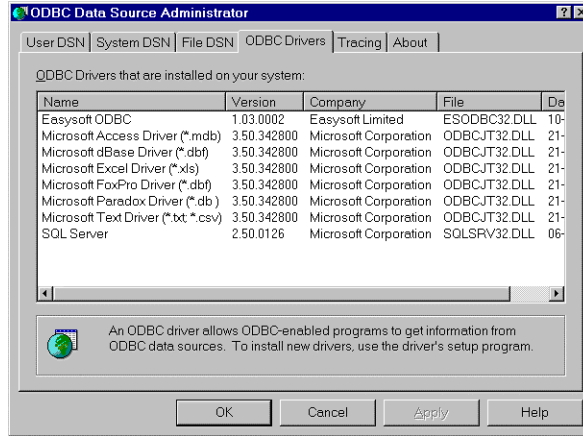
## Removing or Modifying a Data Source

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 22) gives information about drivers.



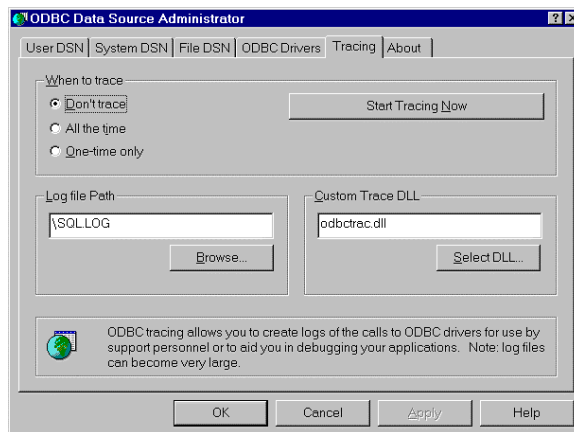
**Figure 22.**

**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 23) is used to specify how the ODBC Driver Manager traces calls to ODBC functions.



**Figure 23.**

**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

# CODA Table Descriptions

The tables that users see when accessing CODA data using Easysoft ODBC are called *CODA tables*. This chapter describes these.

Table 5 lists the CODA tables, their function, their equivalent menu access in CODA, and their equivalent in IASLINK.

<b>Table 5. List of CODA tables</b>			
<b>CODA table name</b>	<b>Contains</b>	<b>CODA menu</b>	<b>IASLINK name</b>
ACCOUNT	Easysoft optimisation	-	-
ACCOUNT_GROUP	Account Group information	MM AG	GROUP
BALANCE	Budgets	BU	BALANCE
BUDGET	Budgets	BU	BUDGET
COMPANY	CODA Company information	MA GM CO	COMPANY
CURRENCY	Currencies	MM CM	CURNCYD
CURRENCY_RATE	Currency rates	MM CM	CURNCYR
DESTINATION	Intercompany Destination	IC MD	DESTNTN
DETAILS [note 1]	Input to books	IN IN	DOCUMENT
DETAILS_COMMENT [note 1]	Details comments	IN IN	COMMENT
DOCUMENT_MASTER	Document Masters	MM DO	DOCMST
GROUP_HIERARCHY	Account Group hierarchy	MM AG	GRPREF
INTRAY [note 1]	Input to Inray	IN AD	INTRAY
INTRAY_COMMENT [note 1]	Inray comments	IN AD	COMMENT
LEDGER	Ledgers	MM LE	LEDGER
MOTHBALL	Mothballed documents	Browsers, LRS	DETAILS
MOTHBALL_COMMENT	Mothballed documents comments	Browsers, LRS	COMMENT
NAME	Name and Address	MM NA	NAME
NAME_ADDRESS	Name and Address	MM NA	NAME
NAME_BANK	Name and Address	MM NA	NAME
NAME_COMMENT	Name and Address account comments	MM NA	NAMCOMM
NAME_CONTACT	Name and Address contact list	MM NA	NAMCONT
NAME_LEDGER_COMMENT	Name and Address ledger comments	MM NA	NAMCOMML
NAME_OWNER	Name and Address financial details	MM NA	NAMOWNER
NOMINAL	Nominal accounts	MM NO	NOMINAL
NOMINAL_COMMENT	Nominal accounts comments list	MM NO	NOMCOMM
SUBACCOUNT	Subaccounts	MM SA	SUBACC
SUBACCOUNT_COMMENT	Subaccount comments	MM SA	SUBCOMM
SUBACCOUNT_LOOKUP	Subaccounts	MM SA	SUBACC
SUBACCOUNT_OWNER	Subaccount Nominal/ledger list	MM SA	SUBOWNER
TAX_CODE	Tax Codes	MM TX	TAXCODE
TEXT	Field labels, messages in CODA	MA GM TE	TEXT

Note 1: CODA security on these tables is based on CODA "Browse Accounts".

The structures of the CODA tables are themselves described in tabular form in this chapter. *Column details* tables are those tables in this document which describe the CODA tables. *Index details* tables are those tables in this document which describe indexes that CODA tables have. The information contained in the index tables is read-only, and when you connect to the CODA data, these indexes are not visible as CODA tables since they are part of the internal structure of the CODA tables.

## Column Details Tables

*Column name* is the name of the column as it appears to users of ODBC applications. The columns that you see in an application depend upon the CODA version that you are using; the Column Details tables list what you will see for the various versions of CODA from 6.4 upwards. **PRIMARY KEYS** are indicated by **BOLD CAPITAL** text. The values NOMINAL\_ID, SUBACCOUNT\_ID and LEVEL3\_ID represent columns which reference internal identification numbers and which are not seen on CODA screens.

*Data type* is the SQL data type of the column and *size* is the number of bytes needed to store the data. If the data type is *varchar*, then *size* is the maximum number of bytes needed to store the data. The format for columns of datatype DATE depends upon the Windows settings.

If a default value exists for a particular CODA column, this is shown in the column labelled *Def*. If there is no default, the column is blank.

*CODA name* is the corresponding name of the column as it appears on CODA menus. The value “[field not named]” indicates that the corresponding CODA field has no name on screen, for example, with the COMMENT\_TEXT column. The value “-” indicates that there is no equivalent field in CODA, for example, with COMMENT\_NUMBER.

There are four columns which are a special case, namely CREATED\_USER, MODIFIED\_USER, CREATED\_DATE and MODIFIED\_DATE. The information in these is taken from CODA internal information, and they do not appear on the CODA interface.

*IASLINK* name is the corresponding name of the column in IASLINK. Replication of the IASLINK name is indicated by the suffix (*n*), for example, LB-BUDGET-VALUE(*n*).

## Index details tables

*No* refers to the index number. Each index is given a number for identification.

*Dup* refers to duplicates. Options are Yes or No.

Yes means that duplicate values are allowed in the index.

No means that duplicate values are not allowed in the index.

*Coll* refers to collation, which may be Ascending (A) or Descending (D) for each of the columns in the index. However, in all cases here, since collation is either ascending or descending for all columns, “A” or “D” refers to the collation of each of the columns in the index.

## List of tables

Column name	Data type	Size
NOMINAL	VARCHAR	12
SUBACCOUNT	VARCHAR	12
LEVEL3	VARCHAR	12
NOMINAL_ID	INTEGER	4
SUBACCOUNT_ID	INTEGER	4
LEVEL3_ID	INTEGER	4

ACCOUNT is an Easysoft file which is used to reference internal CODA identification numbers. It is generated when the Codaxref routine is run (see “Setup Routine Step 5”, page 29 and “Ordering BALANCE and BUDGET Data”, page 42). It is shown here, as it appears alongside the CODA tables when an application connects to a CODA data source. It adds keys which are not available in CODA.

Index name	No	Dup	Coll	Fields in index
ACCOUNT_001	1	No	A	NOMINAL, SUBACCOUNT, LEVEL3
ACCOUNT_002	2	Yes	A	NOMINAL
ACCOUNT_003	3	Yes	A	SUBACCOUNT
ACCOUNT_004	4	Yes	A	LEVEL3

Column name	Data type	Size	Def	CODA name	IASLINK name
ACCOUNT_GROUP	VARCHAR	12		Account group	LB-GROUP-CODE
DESCRIPTION	VARCHAR	30		Group title	LB-GROUP-TITLE
GROUP_TYPE	VARCHAR	1	N	Group type	LB-GROUP-TYPE

The ACCOUNT\_GROUP table holds information on hierarchical account groups. The details are kept in the GROUP\_HIERARCHY table.

<b>Table 9. ACCOUNT_GROUP index details</b>				
Index name	No	Dup	Coll	Fields in index
ACCOUNT_GROUP_001	1	No	A	ACCOUNT_GROUP

<b>Table 10. BALANCE column details</b>					
Column name	Data type	Size	Def	CODA name	IASLINK name
NOMINAL	VARCHAR	12		Nominal	LB-BALANCE-NOM
SUBACCOUNT	VARCHAR	12		Subaccount	LB-BALANCE-SUB
LEVEL3	VARCHAR	12		Level3	LB-BALANCE-LV3
BUDGET_YEAR	SMALLINT	2		Year	LB-BALANCE-YEAR
BUDGET_CODE	VARCHAR	1	B	Budget code	LB-BALANCE-BUD
PERIOD..[note 1]	SMALLINT	2		Period	LB-BALANCE-PERIOD
BALANCE	DOUBLE	8		Value	LB-BALANCE-VALUE
ACCOUNT_BALANCE.. [note 2]	VARCHAR	1		-	-

Note 1: Valid range for the PERIOD field is between 1 and the number of periods in the accounting year. Additionally, there are three other allowed values:

-2, indicates an adjustment period

-1, indicates a final period

0, indicates an opening period

Note 2: ACCOUNT\_BALANCE is an internal flag (allowed values: Y, N) which is not seen on the ODA screen. Y indicates that the values shown in the record have been entered as data to the system. This is the normal balance of the account. N indicates that the values shown exist only as calculated data, that is, the total of N/S/L3 or a combination of N/S/L3.

<b>Table 11. BALANCE index details</b>				
Index name	No	Dup	Coll	Fields in index
BALANCE_001	1	No	A	NOMINAL, SUBACCOUNT, LEVEL3, BUDGET_YEAR, BUDGET_CODE, PERIOD
BALANCE_002	2	Yes	A	NOMINAL
BALANCE_003	3	Yes	A	SUBACCOUNT
BALANCE_004	4	Yes	A	LEVEL3

Column name	Data type	Size	Def	CODA name	IASLINK name
NOMINAL	VARCHAR	12		Nominal	LB-BUDGET-NOM
SUBACCOUNT	VARCHAR	12		Sub-account	LB-BUDGET-SUB
LEVEL3	VARCHAR	12		Level-3	LB-BUDGET-LV3
BUDGET_YEAR	SMALLINT	2		Year & period range	LB-BUDGET-YEAR
BUDGET_CODE	VARCHAR	1	B	Budget	LB-BUDGET-BUD
ACCOUNT_BALANCE [note 1]	VARCHAR	1		-	-
ADJUSTMENT_VALUE	DOUBLE	8		Value	LB-BUDGET-ADJUSTMENTS
FINAL_VALUE	DOUBLE	8		Value	LB-BUDGET-FINAL
OPENING_VALUE	DOUBLE	8		Value	LB-BUDGET-OPENING
TOTAL_VALUE [note 2]	DOUBLE	8		Total value	[note 2]
VALUE_n [note 3]	DOUBLE	8		[number]	LB-BUDGET-VALUE(n)

Note 1: ACCOUNT\_BALANCE is an internal flag (allowed values: Y, N) which is not seen on the CODA screen. Y indicates that the values shown in the record have been entered as data to the system. This is the normal balance of the account. N indicates that the values shown exist only as calculated data, that is, the total of N/S/L3 or a combination of N/S/L3.

Note 2: TOTAL\_VALUE contains a computed value which is the sum (for any given record) of the data in all the value columns except FINAL\_VALUE.

Note 3: The BUDGET table contains a variable number of VALUE columns depending upon the number of budget periods. The column VALUE\_n shown here represents one or more columns in the CODA BUDGET table which have sequential numbers starting from 0001 and finishing at 9999. Thus a maximum of 9999 budget periods are supported (in addition to the opening, final and adjustment periods).

Index name	No	Dup	Coll	Fields in index
BUDGET_001	1	No	A	NOMINAL, SUBACCOUNT, LEVEL3, BUDGET_YEAR, BUDGET_CODE
BUDGET_002	2	Yes	A	NOMINAL
BUDGET_003	3	Yes	A	SUBACCOUNT
BUDGET_004	4	Yes	A	LEVEL3

<b>Table 14. COMPANY column details</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
COMPANY_NAME	VARCHAR	40		Name	LB-COMPANY-NAME
TELEPHONE	VARCHAR	20		Telephone	LB-COMPANY-PHONE
FAX	VARCHAR	20		Fax	LB-COMPANY-TELEX
COMPANY_YEAR	SMALLINT	2		Year	LB-COMPANY-YEAR
CURRENT_PERIOD	SMALLINT	2		Period	LB-COMPANY-PERIOD
POST_QUANTITIES_TO_ALL_LEVELS	VARCHAR	1		Balance qtys at NS3	LB-COMPANY-QTYNS3
AUTOMATIC_BATCH_NUMBERS	VARCHAR	1		Automatic batch nos	LB-COMPANY-AUTO-BATCH
LOGS_CHANGES_TO_MASTER_FILES	VARCHAR	1		Master file logging	LB-COMPANY-MFLOG
ALLOWS_DEFERRED_POSTING	VARCHAR	1		Deferred postings	LB-COMPANY-DEFINP
DECIMAL_POINT_CHARACTER	VARCHAR	1		Decimal point	LB-COMPANY-DPT-CHAR
PERIODS_IN_CURRENT_YEAR	SMALLINT	2		Prds/Year	LB-COMPANY-PRDS-THIS-YEAR
ACCUMULATES_FOREIGN_BALANCES	VARCHAR	1		Foreign balances	LB-COMPANY-FOREIGN-BAL
COMPANY_LANGUAGE	VARCHAR	2		Language	LB-COMPANY-LANG
DATE_FORMAT	VARCHAR	1		Date Format	LB-COMPANY-DATE-TYPE
DATE_SEPARATOR_CHARACTER	VARCHAR	1		Separator	LB-COMPANY-DATE-SEP
COMMITMENT_ACCOUNTING	VARCHAR	1		Commitment accounting	LB-COMPANY-COMACC
CURRENCY_LOCATION	VARCHAR	1		Company or group	LB-COMPANY-CURRLOCN
CURRENCY_RULE	VARCHAR	1		rate rule	LB-COMPANY-CURRRULE
CURRENCY	VARCHAR	4		Code	LB-COMPANY-CURRENCY
BANK_NAME	VARCHAR	12		Default bank account	LB-COMPANY-BANK-NAME
EARLIEST_YEAR_WITH_DATA	SMALLINT	2		Earliest year with data	LB-COMPANY-BASE-YEAR
EARLIEST_YEAR_TO_POST_TO	SMALLINT	2		Earliest year to post to	LB-COMPANY-MIN-YEAR
FIXED_ASSETS	VARCHAR	1		[CODA internal]	LB_COMPANY_INC_FA
POST_DEBITS	VARCHAR	1		Are your debits +ve	LB-COMPANY-POSDEBITS
SPECIAL_TAX_CODE	VARCHAR	2		Special tax code	LB-COMPANY-SPEC-TAX-CODE
ADDRESS_FORMAT	VARCHAR	2		Address format	LB-COMPANY-AFMT
BANK_FORMAT	VARCHAR	2		Bank Format	LB-COMPANY-BFMT
SORT_CODE	VARCHAR	12		Sort Code	LB-COMPANY-SORTCD
ACCOUNT_NUMBER	VARCHAR	20		Account Number	LB-COMPANY-ACTNUM
ACCOUNT_NAME	VARCHAR	40		Account Name	LB-COMPANY-ACTNAM
ACCOUNT_REFERENCE	VARCHAR	20		Account Reference	LB-COMPANY-ACTREF
TAX_FORMAT	VARCHAR	2		Tax format	LB-COMPANY-TFMT
TAX_ID_NUMBER	VARCHAR	25		Tax id Number	LB-COMPANY-TAXID
PREVIOUS_TAX_ID_NUMBER	VARCHAR	25		Previous tax id no	LB-COMPANY-PREVID
MEMBER_STATE_ID	VARCHAR	2		Member state id	LB-COMPANY-MEMID
ADDRESS_LINE_1	VARCHAR	35		Address	LB-COMPANY-ADDR1
ADDRESS_LINE_2	VARCHAR	35		Address	LB-COMPANY-ADDR2
ADDRESS_LINE_3	VARCHAR	35		Address	LB-COMPANY-ADDR3
ADDRESS_LINE_4	VARCHAR	35		Address	LB-COMPANY-ADDR4
TOWN	VARCHAR	35		Post town	LB-COMPANY-TOWN
COUNTY	VARCHAR	35		County	LB-COMPANY-COUNTY
POST_CODE	VARCHAR	10		Post code	LB-COMPANY-POSTCODE
COUNTRY	VARCHAR	35		Country	LB-COMPANY-COUNTRY
<b>Version 6.6 and above</b>					
COMPANY_NUMBER	VARCHAR	25		Company number	LB-COMPANY-NUM
THOUSANDS_SEPARATOR	VARCHAR	1		Thousands Separator	LB-COMPANY-THOUSANDS
<b>Version 6.7 and above</b>					
MULTI_LEDGER	VARCHAR	1		Multi ledgers	LB-COMPANY-MULTI-LEDGER
RESERVE_STATUS	VARCHAR	1		Reserved status	LB-COMPANY-RESERVE-STATUS

Table continued on next page

<b>COMPANY column details (continued)</b>					
<b>Version 7.0 and above</b>					
REG_DATE	DATE	6		Closed reg. date	LB-COMPANY-REG-DATE
<b>Version 7.2</b>					
CREDIT_LEVEL3	VARCHAR	1		Credit to level3	LB-COMPANY-CREDIT-LV3
TURNOVER_LEVEL3	VARCHAR	1		Turnover to level3	LB-COMPANY-TRNOVR-LV3
ALLOCATION_DATE	VARCHAR	3		Alloc. date method	LB-COMPANY-ALDATE
<b>Version 7.3</b>					
PARALLEL_COMPANY	VARCHAR	9		Parallel Company	LB-COMPANY-PARALLEL-COMPANY

**Table 15. COMPANY index details**

Index name	No	Dup	Coll	Fields in index
COMPANY_001	1	No	A	COMPANY_NAME

**Table 16. CURRENCY column details**

Column name	Data type	Size	Def	CODA name	IASLINK name
CURRENCY_CODE	VARCHAR	4		Currency code	LB-CURNCY-CODE
TITLE	VARCHAR	30		[not shown on screen]	LB-CURDET-TITLE
UNIT_SINGULAR	VARCHAR	12		Names of units/ Singular	LB-CURDET-US
UNIT_PLURAL	VARCHAR	12		Names of units/ Plural	LB-CURDET-UP
UNIT_ABBREVIATION	VARCHAR	4		Names of units/ Abbrev	LB-CURDET-UA
DECIMAL_SINGULAR	VARCHAR	12		Names of decimals/ Singular	LB-CURDET-DS
DECIMAL_PLURAL	VARCHAR	12		Names of decimals/ Plural	LB-CURDET-DP
DECIMAL_ABBREVIATION	VARCHAR	4		Names of decimals/ Abbrev	LB-CURDET-DA
ALLOW_DECIMALS	VARCHAR	1	Y	Allow decimals	LB-CURDET-DPS
DECIMAL_POINT	VARCHAR	1	full stop	Decimal point symbol	LB-CURDET-DECPT
<b>Version 6.5 and above</b>					
UNIT_INVERT	VARCHAR	1	H	Home/foreign rates	LB-CURDET-INVERT
UNIT_SCALE	SMALLINT	2	1	The scaling factor	LB-CURDET-SCALE
GENDER_OF_UNITS	VARCHAR	1	N	Names of units/ Gender	LB-CURDET-GENDER-U
GENDER_OF_DECIMALS	VARCHAR	1	N	Names of decimals/ Gender	LB-CURDET-GENDER-D
<b>Version 7.2</b>					
CURRENCY_END_DATE	DATE	6	31-DEC-3000	Currency end date	LB-CURNCY-ENDDATE
<b>Version 7.3</b>					
PARENT_CODE	VARCHAR	4		Parent currency	LB-CURNCY-PARENT-CODE
RATE_TO_PARENT	DOUBLE	8		Rate	LB-CURNCY-RATE-TO-PARENT
PARENT_EFFECTIVE_DATE	DATE	6		Effective Date	LB-CURNCY-PARENT-EFF-DATE

<b>Table 17. CURRENCY index details</b>				
Index name	No	Dup	Coll	Fields in index
CURRENCY_001	1	No	A	CURRENCY_CODE

<b>Table 18. CURRENCY_RATE column details</b>					
Column name	Data type	Size	Def	CODA name	IASLINK name
CURRENCY_CODE	VARCHAR	4		Currency code	LB-CURNCY-CODE
CURRENCY_DATE	DATE	6		Effective date	LB-CURNCY-DATE
CURRENCY_RATE	DOUBLE	8		Exchange rate	LB-CURNCY-RATE
<b>Version 7.3</b>					
PARENT_CODE	VARCHAR	4		Parent currency	LB-CURNCY-PARENT-CODE
RATE_TO_PARENT	DOUBLE	8		Rate	LB-CURNCY-RATE-TO-PARENT
PARENT_EFFECTIVE_DATE	DATE	6		Effective Date	LB-CURNCY-PARENT-EFF-DATE

<b>Table 19. CURRENCY_RATE index details</b>				
Index name	No	Dup	Coll	Fields in index
CURRENCY_RATE_001	1	No	A	CURRENCY_CODE, CURRENCY_DATE



<b>Table 20. DESTINATION column details</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
<b>Version 6.5 and above</b>					
<b>DESTINATION_NAME</b>	VARCHAR	12		Destination name	LB-DESTINATION-NAME
TITLE	VARCHAR	40		Destination title	LB-DESTINATION-TITLE
DESCRIPTION	VARCHAR	18		Description	LB-DESTINATION-DESCR
DESTINATION_TYPE	VARCHAR	3		Destination type	LB-DESTINATION-TYPE
COMPANY	VARCHAR	9		Company name	LB-DESTINATION-COMPANY
ACCESS_MEDIUM	VARCHAR	3		Transfer medium	LB-DESTINATION-MEDIUM
ACCESS_METHOD	VARCHAR	3		Transfer method	LB-DESTINATION-METHOD
DECNET_NODE	VARCHAR	6		DECnet node name	LB-DESTINATION-NODE
DECNET_OBJECT	VARCHAR	12		IAS object name	LB-DESTINATION-OBJECT
OUR_NAME	VARCHAR	12		Our destination name	LB-DESTINATION-OUR-NAME
FORWARD	VARCHAR	12		Forward destination	LB-DESTINATION-FORWARD
HIS_NOMINAL	VARCHAR	12		Destination accounts: Nominal	LB-DESTINATION-HIS-NOM
HIS_SUBACCOUNT	VARCHAR	12		Destination accounts: Sub-account	LB-DESTINATION-HIS-SUB
HIS_LEVEL3	VARCHAR	12		Destination accounts: Level3	LB-DESTINATION-HIS-LV3
MY_NOMINAL	VARCHAR	12		Own accounts: Nominal	LB-DESTINATION-MY-NOM
MY_SUBACCOUNT	VARCHAR	12		Own accounts: Sub- account	LB-DESTINATION-MY-SUB
MY_LEVEL3	VARCHAR	12		Own accounts: Level3	LB-DESTINATION-MY-LV3
CURRENCY	VARCHAR	4		Transfer currency	LB-DESTINATION-CURNCY
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-DESTINATION-DOCTYPE
CURRENCY_CODE	VARCHAR	1		Document currency	LB-DESTINATION-DOCCURR
VALIDATION	VARCHAR	1	N	A/C validat'n method	LB-DESTINATION-VALID
RULE	VARCHAR	12		Structure name	LB-DESTINATION-RULE
FAILURE	VARCHAR	1	E	Failure action	LB-DESTINATION-FAIL
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-DESTINATION-SECLVL
<b>Version 6.6 and above</b>					
HIS_COMMENT	VARCHAR	1	N	Destination accounts: Comment on I/C acc.	LB-DESTINATION-HIS-COMM
MY_COMMENT	VARCHAR	1	N	Own accounts: Comment on I/C acc.	LB-DESTINATION-MY-COMM

Index name	No	Dup	Coll	Fields in index
DESTINATION_001	1	No	A	DESTINATION_NAME

Column name	Data type	Size	Def	CODA name	IASLINK name
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-DET-DOC-TYPE
DOCUMENT_NUMBER	INTEGER	4		Document number	LB-DET-DOC-NUM
LINE_NUMBER	SMALLINT	2		Line number	LB-DET-LINE
DOCUMENT_DATE	DATE	6		Document date	LB-DET-DOC-DATE
DOCUMENT_CURRENCY	VARCHAR	4		Currency	LB-DET-DOC-CURR
DOCUMENT_VALUE	DOUBLE	8		Value	LB-DET-VALUE
NOMINAL	VARCHAR	12		Nominal	LB-DET-NOMINAL
SUBACCOUNT	VARCHAR	12		Subaccount	LB-DET-SUBACC
LEVEL3	VARCHAR	12		Level3	LB-DET-LEVEL3
POST_FLAG	VARCHAR	1	N	[CODA internal]	LB-DET-POST
DOCUMENT_FLAG	VARCHAR	1		Document control flag	LB-DET-DOC-FLAG
HOME_VALUE	DOUBLE	8		Document value	LB-DET-DOC-HOME
DOCUMENT_YEAR	SMALLINT	2		Year and period	LB-DET-YEAR
DOCUMENT_PERIOD	SMALLINT	2		Year and period	LB-DET-PERIOD
DESTINATION	VARCHAR	12		Destination	LB-DET-DESTINATION
DST_FLAG	VARCHAR	1		Destination control flag	LB-DET-DST-FLAG
DST_VALUE	DOUBLE	8		Destination foreign value	LB-DET-DST-FVAL
NOMINAL_FLAG	VARCHAR	1		Nominal control flag	LB-DET-NOM-FLAG
NOMINAL_VALUE	DOUBLE	8		Nominal foreign value	LB-DET-NOM-FVAL
SUBACCOUNT_FLAG	VARCHAR	1		Subaccount control flag	LB-DET-SUB-FLAG
SUBACCOUNT_VALUE	DOUBLE	8		Subaccount foreign value	LB-DET-SUB-FVAL
LEVEL3_FLAG	VARCHAR	1		Level3 control flag	LB-DET-LV3-FLAG
LEVEL3_VALUE	DOUBLE	8		Level3 foreign value	LB-DET-LV3-FVAL
DESCRIPTION	VARCHAR	30		Description	LB-DET-DESCR
QUANTITY_1	DOUBLE	8		Quantity 1	LB-DET-QTY1
QUANTITY_2	DOUBLE	8		Quantity 2	LB-DET-QTY2
DUE_DATE	DATE	6		Due date	LB-DET-DUE-DATE
EXTERNAL_TEXT	VARCHAR	8		External reference	LB-DET-EXT-TEXT
EXTERNAL_NUMBER	INTEGER	4		External reference	LB-DET-EXT-NUM
DOCUMENT_REFERENCE	VARCHAR	24		Document reference	LB-DET-DOC-REF
AUTHOR	VARCHAR	4		Authorising user	LB-DET-AUTHOR
IF_VALUE	DOUBLE	8		If value	LB-DET-IF-VALUE
IF_DATE	DATE	6		If date	LB-DET-IF-DATE
BATCH	INTEGER	4		Batch	LB-DET-BATCH
DOCUMENT_COUNT	SMALLINT	2		[field not named]	LB-DET-COUNT
DOCUMENT_STATUS	VARCHAR	1		Status	LB-DET-DOCSTS
DETAIL_STATUS	VARCHAR	1		[field not named]	LB-DET-DETSTS
RECONCILIATION_STATUS	VARCHAR	1		Reconciliation status	LB-DET-RECSTS
JR_DATE	DATE	6		Reversing Document date	LB-DET-JR-DATE
JR_YEAR	SMALLINT	2		Reversing Year	LB-DET-JR-YEAR
JR_PERIOD	SMALLINT	2		Reversing Period	LB-DET-JR-PERIOD
REFERENCE_DATE_1	DATE	6		Date 1 Reference	LB-DET-DATE-REF1

Table continued on next page

<b>DETAILS column details (continued)</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
REFERENCE_DATE_2	DATE	6		Date 2 Reference	LB-DET-DATE-REF2
TAX_CODE_1	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_1	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_2	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_2	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_3	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_3	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_4	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_4	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
ACCOUNT_NAME	VARCHAR	40		Account name	LB-DET-ACTNAM
ADDRESS_1	VARCHAR	35		Address	LB-DET-ADRS
ADDRESS_2	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_3	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_4	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_5	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_6	VARCHAR	35		[Address]	LB-DET-ADRS
POST_CODE	VARCHAR	10		Code	LB-DET-CODE
COUNTRY	VARCHAR	35		Country	LB-DET-COUNTRY
PAY_APPLY	VARCHAR	1		Pay Apply	LB-DET-PAY-APPLY
PAY_BANK_INDEX	VARCHAR	1		Pay Bank Index	LB-DET-PAY-BANK-IDX
PAY_ADDRESS_INDEX	VARCHAR	1		Pay Address Index	LB-DET-PAY-ADRS-IDX
PAY_BANK_ACCOUNT	VARCHAR	12		Pay Bank Account	LB-DET-PAY-BANK-ACC
PAY_MEDIA	VARCHAR	12		Pay Media	LB-DET-PAY-MEDIA
PAY_DATE	DATE	6		Pay Date	LB-DET-PAY-DATE
PAY_REFERENCE	INTEGER	4		Pay Reference	LB-DET-PAY-REF
INPUT_DATE	DATE	6		Entered on	LB-RDET-INP-DATE
INPUT_USER	VARCHAR	20		By	LB-RDET-USER
PAYMENT_NUMBER [note 1]	SMALLINT	2		Payment number	LB-RDET-PAY-NUM
PAYMENT_DATE [note 1]	DATE	4		Date paid	LB-RDET-DATE-PAID
RS_NUMBER [note 1]	SMALLINT	2		Remittance/statement number	-
<b>Version 7.3</b>					
CURRENCY_SCAN_1	VARCHAR	1		[-]	LB-DET-CURR-SCAN1
CURRENCY_SCAN_2	VARCHAR	1		[-]	LB-DET-CURR-SCAN2
CURRENCY_SCAN_3	VARCHAR	1		[-]	LB-DET-CURR-SCAN3
CURRENCY_SCAN_4	VARCHAR	1		[-]	LB-DET-CURR-SCAN4
CURRENCY_SCAN_5	VARCHAR	1		[-]	LB-DET-CURR-SCAN5
RECONCILIATION_DATE	DATE	6		Reconciled	LB-DET-RECDATE

Note 1. This column is read-only.

The structure of the DETAILS table is similar to that of INTRAY. It is used to post a document directly to the books. On the other hand, INTRAY uses the CODA Inray facility - values are entered into the system, but the document does not need to balance, and so can be partially completed. Furthermore, INTRAY can be updated, whereas DETAILS cannot.

Index name	No	Dup	Coll	Fields in index
DETAILS_001	1	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, LINE_NUMBER
DETAILS_002	2	Yes	A	NOMINAL, DOCUMENT_DATE
DETAILS_003	3	Yes	A	SUBACCOUNT, DOCUMENT_DATE
DETAILS_004	4	Yes	A	LEVEL3, DOCUMENT_DATE
DETAILS_005	5	Yes	A	NOMINAL, DOCUMENT_YEAR, DOCUMENT_PERIOD
DETAILS_006	6	Yes	A	SUBACCOUNT, DOCUMENT_YEAR, DOCUMENT_PERIOD
DETAILS_007	7	Yes	A	LEVEL3, DOCUMENT_YEAR, DOCUMENT_PERIOD
DETAILS_008	8	Yes	A	DOCUMENT_YEAR, DOCUMENT_PERIOD
DETAILS_009	9	Yes	A	INPUT_DATE

Column name	Data type	Size	Def	CODA name	IASLINK name
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-COMMENT-DOC-TYPE
DOCUMENT_NUMBER	INTEGER	4		Document number	LB-COMMENT-DOC-NUM
DOCUMENT_LINE	SMALLINT	2		Line number	LB-COMMENT-DOC-LINE
COMMENT_TEXT	VARCHAR	78		[field not named]	LB-COMMENT-TEXT
COMMENT_NUMBER	INTEGER	4		-	-

Index name	No	Dup	Coll	Fields in index
DETAILS_COMMENT_001	1	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, DOCUMENT_LINE, COMMENT_NUMBER

Column name	Data type	Size	Def	CODA name	IASLINK name
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-DOCM-TYPE
FORMAT	VARCHAR	2		Document format	LB-DOCM-FMT
MNEMONIC	VARCHAR	3		Document mnemonic	LB-DOCM-MNEM
BANK_RECONCILIATION_CODE_1	VARCHAR	4		Bank type code	LB-DOCM-BANK1
BANK_RECONCILIATION_CODE_2	VARCHAR	4		Bank type code	LB-DOCM-BANK2
DESCRIPTION	VARCHAR	30		Description	LB-DOCM-DESCR
DUE_DATE_REQUIRED	VARCHAR	1	N	Due date required	LB-DOCM-DUEDATE
DEFAULT_STATUS	VARCHAR	1		Default posting sts	LB-DOCM-DEFSTS
TURNOVER_REQUIRED	VARCHAR	1	N	Adds to turnover	LB-DOCM-TRNOVR
POST_TO_INTRAY	VARCHAR	1	N	Must place on intray	LB-DOCM-INTRAY
INTER_COMPANY_ALLOWED	VARCHAR	1	N	Inter-company doc	LB-DOCM-INTERCOMP
POST_TO_DETAILS	VARCHAR	1	N	Autopost to details	LB-DOCM-AUTO
CHANGE_DOCUMENT_TYPE	VARCHAR	1	N	Change doc type	LB-DOCM-CHANGE
CHECK_EXTERNAL_REFERENCES	VARCHAR	1	*	Duplicate ext ref	LB-DOCM-CHK-EXT
DUPLICATE_EXTERNAL_REFERENCE	VARCHAR	1	N	Duplicate ext ref	LB-DOCM-DUPL-EXT
REQUIRE_AUTHORISING_USER	VARCHAR	1	N	Requires auth. user	LB-DOCM-AUTHOR
SIGN_RULE	VARCHAR	1	N	Sign rule	LB-DOCM-SIGN
SEQUENCING_ACTION	VARCHAR	1	U	Sequence rule	LB-DOCM-SEQ

**Table continued on next page**

<b>DOCUMENT_MASTER column details (continued)</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
START_DOCUMENT_NUMBER_1	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_2	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_3	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_4	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_5	INTEGER	4		Sequence start	LB-DOCM-START
START_DOCUMENT_NUMBER_6	INTEGER	4		Sequence start	LB-DOCM-START
END_DOCUMENT_NUMBER_1	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_2	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_3	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_4	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_5	INTEGER	4		Sequence end	LB-DOCM-END
END_DOCUMENT_NUMBER_6	INTEGER	4		Sequence end	LB-DOCM-END
NUMBER_RANGE_OPEN	VARCHAR	6		Open, in use	LB-DOCM-OPEN
SEQUENCE_IN_USE	VARCHAR	6	NNNNN	Open, in use	LB-DOCM-SEQ-INUSE
CURRENCY	VARCHAR	4		Document currency	LB-DOCM-CURRENCY
CHANGE_CURRENCY	VARCHAR	1	Y	Change currency	LB-DOCM-CURR-CHNG
GENERATE_TAX_LINES	VARCHAR	1	N	Generates tax lines	LB-DOCM-TAX
CHECK_INTRAY_DOC_BALANCES	VARCHAR	1	N	Balance intray docs	LB-DOCM-INTRAY-BAL
SELF_PROPORTIONING	VARCHAR	1	N	Self proportioning	LB-DOCM-PROP
REVERSE_QUANTITY_SIGNS	VARCHAR	1	Y	Reverse quantities	LB-DOCM-REV-QTY
DEFAULT_NOMINAL	VARCHAR	12		Account	LB-DOCM-NOM
DEFAULT_SUBACCOUNT	VARCHAR	12		Account	LB-DOCM-SUB
DEFAULT_LEVEL3	VARCHAR	12		Account	LB-DOCM-LV3
PRESERVE_PERIOD	VARCHAR	1	N	Preserve I/C period	LB-DOCM-PRSV-PRD
PRESERVE_EXTERNAL_REFERENCES	VARCHAR	1	Y	Preserve I/C ext. ref	LB-DOCM-PRSV-EXT
PRESERVE_BATCH_NUMBER	VARCHAR	1	N	Retain batch number	LB-DOCM-PRSV-BAT
PRESERVE_SOURCE_EXT_REF	VARCHAR	1	Y	Preserve i/c ext. ref	LB-DOCM-PRSV-SRC
DEFAULT_EXTERNAL_TEXT	VARCHAR	8		Ext. ref	LB-DOCM-EXT-TXT
DEFAULT_EXTERNAL_NUMBER	INTEGER	4		Ext. ref	LB-DOCM-EXT-NUM
DOCUMENT_REFERENCE	VARCHAR	24		Doc. ref	LB-DOCM-DOC-REF
CHECK_EXTERNAL_DOCUMENT	VARCHAR	1	E	Ext / doc ref check	LB-DOCM-EXT-DOC
CHECK_DOCUMENT	VARCHAR	1		Duplicate doc ref	LB-DOCM-CHK-DOC
DUPLICATE_DOCUMENT	VARCHAR	1	N	Duplicate doc ref	LB-DOCM-DUPL-DOC
ACCOUNT_MASKS	VARCHAR	1	N	Use account masks	LB-DOCM-USEMSK
ACCOUNT_EXTERNAL	VARCHAR	1	Y	Use references	LB-DOCM-USEEXT
USE_DOCUMENT_REFERENCE	VARCHAR	1	Y	Use references	LB-DOCM-USEDOC
USE_DATE_REFERENCE_1	VARCHAR	1	N	Use references	LB-DOCM-USEDT1
USE_DATE_REFERENCE_2	VARCHAR	1	N	Use references	LB-DOCM-USEDT2
COMPANY_MASK	VARCHAR	12		Company mask	LB-DOCM-CMP-MASK
NOMINAL_MASK	VARCHAR	12		Account masks	LB-DOCM-NOM-MASK
SUBACCOUNT_MASK	VARCHAR	12		Account masks	LB-DOCM-SUB-MASK
LEVEL3_MASK	VARCHAR	12		Account masks	LB-DOCM-LV3-MASK
DATE_REFERENCE_1	VARCHAR	11		Date 1 ref	LB-DOCM-DATE1
DATE_REFERENCE_2	VARCHAR	11		Date 2 ref	LB-DOCM-DATE2
SPECIAL_TAX	VARCHAR	1	N	Enter special tax	LB-DOCM-SPEC-TAX
<b>Version 6.6 and above</b>					
CLEAR_CURRENCY	VARCHAR	1	N	Reprompt currency	LB-DOCM-CLEAR-CUR
<b>Version 6.7 and above</b>					
PROPAGATE_PAY	VARCHAR	1	N	Propagate pay number	LB-DOCM-PROP-PAY
<b>Version 7.2</b>					
GROSS_EXPENSE	VARCHAR	1	N	Gross Expense	LB-DOCM-GROSS-NETT
<b>Version 7.3</b>					
DEFAULT_CURRENCY_LEVEL	VARCHAR	1		Currency level	LB-DOCM-CURR-LEVEL

<b>Table 27. DOCUMENT_MASTER index details</b>				
<b>Index name</b>	<b>No</b>	<b>Dup</b>	<b>Coll</b>	<b>Fields in index</b>
DOCUMENT_001	1	No	A	DOCUMENT_TYPE

<b>Table 28. GROUP_HIERARCHY column details</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
<b>PARENT</b>	VARCHAR	12		Parents	LB-GRPXREF-PARENT
<b>CHILD</b>	VARCHAR	12		Children	LB-GRPXREF-CHILD
CHILD_TYPE [note 1]	VARCHAR	1		-	LB-GRPXREF-CHILD-TYPE
PARENT_ID	INTEGER	4		-	-
CHILD_ID	INTEGER	4		-	-

See also ACCOUNT\_GROUP table.

Note 1: CHILD\_TYPE is a flag which may refer to an account group, a nominal, a subaccount or a level 3. For any given account which is to be included in more than one account group, the GROUP\_HIERARCHY table should be used; there should be one record for each account in the account group (PARENT), and the CHILD should be the account that is in the group.

<b>Table 29. GROUP_HIERARCHY index details</b>				
<b>Index name</b>	<b>No</b>	<b>Dup</b>	<b>Coll</b>	<b>Fields in index</b>
GROUP_HIERARCHY_001	1	No	A	PARENT_ID, CHILD_ID, CHILD_TYPE
GROUP_HIERARCHY_002	2	Yes	A	PARENT

<b>Table 30. INTRAY column details</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-DET-DOC-TYPE
DOCUMENT_NUMBER	INTEGER	4		Initial seq number	LB-DET-DOC-NUM
<b>LINE_NUMBER</b> [note 1]	SMALLINT	2		Line number	LB-DET-LINE
DOCUMENT_DATE	DATE	6		Document date	LB-DET-DOC-DATE
DOCUMENT_CURRENCY	VARCHAR	4		Currency	LB-DET-DOC-CURR
DOCUMENT_VALUE	DOUBLE	8		Value	LB-DET-VALUE
NOMINAL	VARCHAR	12		Nominal	LB-DET-NOMINAL
SUBACCOUNT	VARCHAR	12		Subaccount	LB-DET-SUBACC
LEVEL3	VARCHAR	12		Level3	LB-DET-LEVEL3
POST_FLAG	VARCHAR	1	N	[CODA internal]	LB-DET-POST
DOCUMENT_FLAG	VARCHAR	1		Document control flag	LB-DET-DOC-FLAG
HOME_VALUE	DOUBLE	8		Document value	LB-DET-DOC-HOME
DOCUMENT_YEAR	SMALLINT	2		Year and period	LB-DET-YEAR
DOCUMENT_PERIOD	SMALLINT	2		Year and period	LB-DET-PERIOD
DESTINATION	VARCHAR	12		Destination	LB-DET-DESTINATION
DST_FLAG	VARCHAR	1		Destination control flag	LB-DET-DST-FLAG
DST_VALUE	DOUBLE	8		Destination foreign value	LB-DET-DST-FVAL
NOMINAL_FLAG	VARCHAR	1		Nominal control flag	LB-DET-NOM-FLAG
NOMINAL_VALUE	DOUBLE	8		Nominal foreign value	LB-DET-NOM-FVAL
SUBACCOUNT_FLAG	VARCHAR	1		Subaccount control flag	LB-DET-SUB-FLAG
SUBACCOUNT_VALUE	DOUBLE	8		Subaccount foreign value	LB-DET-SUB-FVAL
LEVEL3_FLAG	VARCHAR	1		Level3 control flag	LB-DET-LV3-FLAG
LEVEL3_VALUE	DOUBLE	8		Level3 foreign value	LB-DET-LV3-FVAL
DESCRIPTION	VARCHAR	30		Description	LB-DET-DESCR
QUANTITY_1	DOUBLE	8		Quantity 1	LB-DET-QTY1
QUANTITY_2	DOUBLE	8		Quantity 2	LB-DET-QTY2
DUE_DATE	DATE	6		Due date	LB-DET-DUE-DATE
EXTERNAL_TEXT	VARCHAR	8		External reference	LB-DET-EXT-TEXT
EXTERNAL_NUMBER	INTEGER	4		External reference	LB-DET-EXT-NUM
DOCUMENT_REFERENCE	VARCHAR	24		Document reference	LB-DET-DOC-REF
AUTHOR	VARCHAR	4		Author	LB-DET-AUTHOR
IF_VALUE	DOUBLE	8		Discount	LB-DET-IF-VALUE
IF_DATE	DATE	6		If paid by	LB-DET-IF-DATE
BATCH	INTEGER	4		Batch	LB-DET-BATCH
DOCUMENT_COUNT	SMALLINT	2		[field not named]	LB-DET-COUNT
DOCUMENT_STATUS	VARCHAR	1		Status	LB-DET-DOCSTS
DETAIL_STATUS	VARCHAR	1		[field not named]	LB-DET-DETSTS
RECONCILIATION_STATUS	VARCHAR	1		Reconciliation status	LB-DET-RECSTS
JR_DATE	DATE	6		Reversing Document date	LB-DET-JR-DATE
JR_YEAR	SMALLINT	2		Reversing Year	LB-DET-JR-YEAR
JR_PERIOD	SMALLINT	2		Reversing Period	LB-DET-JR-PERIOD
REFERENCE_DATE_1	DATE	6		Date 1 Reference	LB-DET-DATE-REF1
REFERENCE_DATE_2	DATE	6		Date 2 Reference	LB-DET-DATE-REF2
TAX_CODE_1	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_1	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_2	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_2	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_3	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_3	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_4	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_4	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
ACCOUNT_NAME	VARCHAR	40		Account name	LB-DET-ACTNAM

Table continued on next page

<b>INTRAY column details (continued)</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
ADDRESS_1	VARCHAR	35		Address	LB-DET-ADRS
ADDRESS_2	VARCHAR	35		[field not named]	LB-DET-ADRS
ADDRESS_3	VARCHAR	35		[field not named]	LB-DET-ADRS
ADDRESS_4	VARCHAR	35		[field not named]	LB-DET-ADRS
ADDRESS_5	VARCHAR	35		[field not named]	LB-DET-ADRS
ADDRESS_6	VARCHAR	35		[field not named]	LB-DET-ADRS
POST_CODE	VARCHAR	10		Code	LB-DET-CODE
COUNTRY	VARCHAR	35		Country	LB-DET-COUNTRY
PAY_APPLY	VARCHAR	1		Pay Apply	LB-DET-PAY-APPLY
PAY_BANK_INDEX	VARCHAR	1		Pay Bank Index	LB-DET-PAY-BANK-IDX
PAY_ADDRESS_INDEX	VARCHAR	1		Pay Address Index	LB-DET-PAY-ADRS-IDX
PAY_BANK_ACCOUNT	VARCHAR	12		Pay Bank Account	LB-DET-PAY-BANK-ACC
PAY_MEDIA	VARCHAR	12		Pay Media	LB-DET-PAY-MEDIA
PAY_DATE	DATE	6		Pay Date	LB-DET-PAY-DATE
PAY_REFERENCE	INTEGER	4		Pay Reference	LB-DET-PAY-REF
INPUT_DATE	DATE	6		Entered on	LB-RDET-INP-DATE
INPUT_USER	VARCHAR	20		By	LB-RDET-USER
<b>DOCUMENT_ID</b>	INTEGER	4		[CODA internal]	-
<b>Version 7.3</b>					
CURRENCY_SCAN_1	VARCHAR	1		[-]	LB-DET-CURR-SCAN1
CURRENCY_SCAN_2	VARCHAR	1		[-]	LB-DET-CURR-SCAN2
CURRENCY_SCAN_3	VARCHAR	1		[-]	LB-DET-CURR-SCAN3
CURRENCY_SCAN_4	VARCHAR	1		[-]	LB-DET-CURR-SCAN4
CURRENCY_SCAN_5	VARCHAR	1		[-]	LB-DET-CURR-SCAN5

Note 1. The allowed values that this can take depend upon the sequencing rule (SEQUENCING\_ACTION) that is specified in the DOCUMENT\_MASTER table.

Also see the DETAILS and MOTHBALL tables.

<b>Table 31. INTRAY index details</b>				
<b>Index name</b>	<b>No</b>	<b>Dup</b>	<b>Coll</b>	<b>Fields in index</b>
INTRAY_001	1	No	A	DOCUMENT_ID, LINE_NUMBER
INTRAY_002	2	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, LINE_NUMBER
INTRAY_003	3	Yes	A	NOMINAL, DOCUMENT_DATE
INTRAY_004	4	Yes	A	SUBACCOUNT, DOCUMENT_DATE
INTRAY_005	5	Yes	A	LEVEL3, DOCUMENT_DATE
INTRAY_006	6	Yes	A	INPUT_DATE

<b>Table 32. INTRAY_COMMENT column details</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
<b>DOCUMENT_TYPE</b>	VARCHAR	4		Doc type	LB-COMMENT-DOC-TYPE
<b>DOCUMENT_NUMBER</b>	INTEGER	4		Initial Seq number	LB-COMMENT-DOC-NUM
<b>DOCUMENT_LINE</b>	SMALLINT	2		Line number	LB-COMMENT-DOC-LINE
COMMENT_TEXT	VARCHAR	78		[field not named]	LB-COMMENT-TEXT
<b>COMMENT_NUMBER</b>	INTEGER	4		-	-



Index name	No	Dup	Coll	Fields in index
INTRAY_COMMENT_001	1	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, DOCUMENT_LINE, COMMENT_NUMBER

Column name	Data type	Size	Def	CODA name	IASLINK name
LEDGER	VARCHAR	5		Ledger ident.	LB-LEDGER-CODE
NOMINAL	VARCHAR	12		Control account	LB-LEDGER-NOMINAL
DESCRIPTION	VARCHAR	30		Ledger name	LB-LEDGER-TITLE
CREATED_USER	VARCHAR	20		[CODA internal]	-
CREATED_DATE	DATE	6		[CODA internal]	-
MODIFIED_USER	VARCHAR	20		[CODA internal]	-
MODIFIED_DATE	DATE	6		[CODA internal]	-

Index name	No	Dup	Coll	Fields in index
LEDGER_001	1	No	A	LEDGER

Column name	Data type	Size	Def	CODA name	IASLINK name
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-DET-DOC-TYPE
DOCUMENT_NUMBER	INTEGER	4		Document number	LB-DET-DOC-NUM
LINE_NUMBER	SMALLINT	2		Line number	LB-DET-LINE
DOCUMENT_DATE	DATE	6		Document date	LB-DET-DOC-DATE
DOCUMENT_CURRENCY	VARCHAR	4		Currency	LB-DET-DOC-CURR
DOCUMENT_VALUE	DOUBLE	8		Value	LB-DET-VALUE
NOMINAL	VARCHAR	12		Nominal	LB-DET-NOMINAL
SUBACCOUNT	VARCHAR	12		Subaccount	LB-DET-SUBACC
LEVEL3	VARCHAR	12		Level3	LB-DET-LEVEL3
POST_FLAG	VARCHAR	1	N	[CODA internal]	LB-DET-POST
DOCUMENT_FLAG	VARCHAR	1		Document control flag	LB-DET-DOC-FLAG
HOME_VALUE	DOUBLE	8		Document value	LB-DET-DOC-HOME
DOCUMENT_YEAR	SMALLINT	2		Year and period	LB-DET-YEAR
DOCUMENT_PERIOD	SMALLINT	2		Year and period	LB-DET-PERIOD
DESTINATION	VARCHAR	12		Destination	LB-DET-DESTINATION
DST_FLAG	VARCHAR	1		Destination control flag	LB-DET-DST-FLAG
DST_VALUE	DOUBLE	8		Destination foreign value	LB-DET-DST-FVAL
NOMINAL_FLAG	VARCHAR	1		Nominal control flag	LB-DET-NOM-FLAG

**Table continued on next page**

<b>MOTHBALL column details (continued)</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
NOMINAL_VALUE	DOUBLE	8		Nominal foreign value	LB-DET-NOM-FVAL
SUBACCOUNT_FLAG	VARCHAR	1		Subaccount control flag	LB-DET-SUB-FLAG
SUBACCOUNT_VALUE	DOUBLE	8		Subaccount foreign value	LB-DET-SUB-FVAL
LEVEL3_FLAG	VARCHAR	1		Level3 control flag	LB-DET-LV3-FLAG
LEVEL3_VALUE	DOUBLE	8		Level3 foreign value	LB-DET-LV3-FVAL
DESCRIPTION	VARCHAR	30		Description	LB-DET-DESCR
QUANTITY_1	DOUBLE	8		Quantity 1	LB-DET-QTY1
QUANTITY_2	DOUBLE	8		Quantity 2	LB-DET-QTY2
DUE_DATE	DATE	6		Due date	LB-DET-DUE-DATE
EXTERNAL_TEXT	VARCHAR	8		External reference	LB-DET-EXT-TEXT
EXTERNAL_NUMBER	INTEGER	4		External reference	LB-DET-EXT-NUM
DOCUMENT_REFERENCE	VARCHAR	24		Document reference	LB-DET-DOC-REF
AUTHOR	VARCHAR	4		Authorising user	LB-DET-AUTHOR
IF_VALUE	DOUBLE	8		If value	LB-DET-IF-VALUE
IF_DATE	DATE	6		If date	LB-DET-IF-DATE
BATCH	INTEGER	4		Batch	LB-DET-BATCH
DOCUMENT_COUNT	SMALLINT	2		[field not named]	LB-DET-COUNT
DOCUMENT_STATUS	VARCHAR	1		Status	LB-DET-DOCSTS
DETAIL_STATUS	VARCHAR	1		[field not named]	LB-DET-DETSTS
RECONCILIATION_STATUS	VARCHAR	1		Reconciliation status	LB-DET-RECSTS
JR_DATE	DATE	6		Reversing Document date	LB-DET-JR-DATE
JR_YEAR	SMALLINT	2		Reversing Year	LB-DET-JR-YEAR
JR_PERIOD	SMALLINT	2		Reversing Period	LB-DET-JR-PERIOD
REFERENCE_DATE_1	DATE	6		Date 1 Reference	LB-DET-DATE-REF1
REFERENCE_DATE_2	DATE	6		Date 2 Reference	LB-DET-DATE-REF2
TAX_CODE_1	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_1	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_2	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_2	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_3	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_3	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
TAX_CODE_4	VARCHAR	4		Special Tax code	LB-DET-SPEC-TAX-CODE
TAX_VALUE_4	DOUBLE	8		Special Tax value	LB-DET-SPEC-TAX-VALUE
ACCOUNT_NAME	VARCHAR	40		Account name	LB-DET-ACTNAM
ADDRESS_1	VARCHAR	35		Address	LB-DET-ADRS
ADDRESS_2	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_3	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_4	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_5	VARCHAR	35		[Address]	LB-DET-ADRS
ADDRESS_6	VARCHAR	35		[Address]	LB-DET-ADRS
POST_CODE	VARCHAR	10		Code	LB-DET-CODE
COUNTRY	VARCHAR	35		Country	LB-DET-COUNTRY
PAY_APPLY	VARCHAR	1		Pay Apply	LB-DET-PAY-APPLY
PAY_BANK_INDEX	VARCHAR	1		Pay Bank Index	LB-DET-PAY-BANK-IDX
PAY_ADDRESS_INDEX	VARCHAR	1		Pay Address Index	LB-DET-PAY-ADRS-IDX
PAY_BANK_ACCOUNT	VARCHAR	12		Pay Bank Account	LB-DET-PAY-BANK-ACC
PAY_MEDIA	VARCHAR	12		Pay Media	LB-DET-PAY-MEDIA
PAY_DATE	DATE	6		Pay Date	LB-DET-PAY-DATE
PAY_REFERENCE	INTEGER	4		Pay Reference	LB-DET-PAY-REF
INPUT_DATE	DATE	6		Entered on	LB-RDET-INP-DATE
INPUT_USER	VARCHAR	20		by	LB-RDET-USER

**Table continued on next page**

<b>MOTHBALL column details (continued)</b>					
PAYMENT_NUMBER	SMALLINT	2		Payment number	LB-RDET-PAY-NUM
PAYMENT_DATE	DATE	4		Date paid	LB-RDET-DATE-PAID
RS_NUMBER	SMALLINT	2		Remittance/statement number	-
<b>Version 7.3</b>					
RECONCILIATION_DATE	DATE	6		Reconciled	-

Note 1. The MOTHBALL table is read-only.

Note 2: The structure of the MOTHBALL table is similar to that of INTRAY and DETAILS tables.

<b>Table 37. MOTHBALL index details</b>					
Index name	No	Dup	Coll	Fields in index	
MOTHBALL_001	1	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, LINE_NUMBER	
MOTHBALL_002	2	Yes	A	NOMINAL, DOCUMENT_DATE	
MOTHBALL_003	3	Yes	A	SUBACCOUNT, DOCUMENT_DATE	
MOTHBALL_004	4	Yes	A	LEVEL3, DOCUMENT_DATE	
MOTHBALL_005	5	Yes	A	NOMINAL, DOCUMENT_YEAR, DOCUMENT_PERIOD	
MOTHBALL_006	6	Yes	A	SUBACCOUNT, DOCUMENT_YEAR, DOCUMENT_PERIOD	
MOTHBALL_007	7	Yes	A	LEVEL3, DOCUMENT_YEAR, DOCUMENT_PERIOD	
MOTHBALL_008	8	Yes	A	DOCUMENT_YEAR, DOCUMENT_PERIOD	
MOTHBALL_009	9	Yes	A	INPUT_DATE	

<b>Table 38. MOTHBALL_COMMENT column details</b>						
Column name	Data type	Size	Def	CODA name	IASLINK name	
DOCUMENT_TYPE	VARCHAR	4		Document type	LB-COMMENT-DOC-TYPE	
DOCUMENT_NUMBER	INTEGER	4		Document number	LB-COMMENT-DOC-NUM	
DOCUMENT_LINE	SMALLINT	2		Line number	LB-COMMENT-DOC-LINE	
COMMENT_TEXT	VARCHAR	78		[field not named]	LB-COMMENT-TEXT	
COMMENT_NUMBER	INTEGER	4		-	-	

<b>Table 39. MOTHBALL_COMMENT index details</b>					
Index name	No	Dup	Coll	Fields in index	
MOTHBALL_COMMENT_001	1	No	A	DOCUMENT_TYPE, DOCUMENT_NUMBER, DOCUMENT_LINE, COMMENT_NUMBER	

<b>Table 40. NAME column details</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-SUB-CODE
LEDGER [note 1]	VARCHAR	5	0000	Subanalyse as	LB-SUB-LEDGER
ADDRESS_ID	SMALLINT	2		Address no (Address data)	LB-SUB-ADDID
TITLE	VARCHAR	40		Sub-account name	LB-SUB-NAME
SHORT_TITLE	VARCHAR	20		Short name	LB-SUB-SHORT
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-SUB-SECLVL
TAX_CODE	VARCHAR	4		Tax code	LB-SUB-TAX
EXPIRY_DATE	DATE	6		Expiry date	LB-SUB-EXPIRY
ACCOUNT_TYPE	VARCHAR	1	B	A/c type	LB-SUB-TYPE
CURRENCY_CODE	VARCHAR	4		Currency	LB-SUB-CURNCY
RECONCILABLE	VARCHAR	1	N	Reconcilable	LB-SUB-RECON
ONE_OFF_SUPPLIERS	VARCHAR	1	N	One off supplier	LB-SUB-EXTADR
ADDRESS_FORMAT	VARCHAR	2		Address format type	LB-SUB-AFMT
BANK_FORMAT	VARCHAR	2		Format code	LB-SUB-BFMT
PRIORITY	VARCHAR	1	1	Account priority	LB-SUB-PRIORITY
SEND_TO_HIGHER	VARCHAR	1		Higher	LB-SUB-HIGHER
SEND_TO_LOWER	VARCHAR	1		Lower	LB-SUB-LOWER
DEFAULT_POSTING_STATUS	VARCHAR	1	A	Default posting sts	LB-SUB-DFLT
CROSS_REFERENCE_ACCOUNT	VARCHAR	12		Cross reference	LB-SUB-XREF
DRAFT	VARCHAR	1	N	Draft	LB-SUB-DRAFT
CHEQUE	VARCHAR	1	Y	Cheque	LB-SUB-CHEQUE
BACS	VARCHAR	1	N	Bacs	LB-SUB-BACS
CASH	VARCHAR	1	Y	Cash	LB-SUB-CASH
SORT_CODE	VARCHAR	12		Sort code	LB-SUB-SRT-CODE
ACCOUNT_NUMBER	VARCHAR	20		Bank account number	LB-SUB-ACT-NUM
ACCOUNT_NAME	VARCHAR	40		Account name	LB-SUB-ACT-NAM
ACCOUNT_REFERENCE	VARCHAR	20		Account reference	LB-SUB-ACT-REF
CREDIT_LIMIT	DOUBLE	8		Credit lim	LB-SUB-CLIM
TRADING_LIMIT	DOUBLE	8		Trade lim	LB-SUB-TLIM
INDIRECT_ACCOUNT	VARCHAR	12		Indirect	LB-SUB-INDIRECT
MNEMONIC_1	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_2	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_3	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_4	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_5	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
MNEMONIC_6	VARCHAR	12		Account mnemonics	LB-SUB-MNEM
ACCOUNT_GROUP [note 2]	VARCHAR	12		Account groups	LB-SUB-GROUP
LEVEL3_ACCOUNT_GROUP	VARCHAR	12		Group as level3	LB-SUB-GROUP-LV3
ONE_CHEQUE_PER_TRANSACTION	VARCHAR	1	N	Single cheque / trans	LB-SUB-SINGCHQ
USE_SPECIAL_TAX	VARCHAR	1	N	Supply special tax	LB-SUB-TAX-FLAG
TAX_SORT_CODE	VARCHAR	12		Sort code	LB-SUB-TAX-SORT
TAX_ACCOUNT_NUMBER	VARCHAR	20		Account Number	LB-SUB-TAX-ACTNUM
TAX_ACCOUNT_NAME	VARCHAR	40		Account Name	LB-SUB-TAX-ACTNAM
TAX_ACCOUNT_REFERENCE	VARCHAR	20		Account Reference	LB-SUB-TAX-ACTREF
TAX_FORMAT	VARCHAR	2		General tax type	LB-SUB-TFMT
TYPE_OF_TAX_ID_NUMBER	VARCHAR	1		Type of TIN	LB-SUB-GENTAX-TYPE
TAX_ID_NUMBER	VARCHAR	25	00000 0000	Tax id number	LB-SUB-GENTAX-TIN

Table continued on next page

<b>NAME column details (continued)</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
<b>Version 6.5 and above</b>					
FOR_THE_ATTENTION_OF	VARCHAR	40		For atn of	LB-SUB-FAO
ADDRESS_LINE_1	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_2	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_3	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_4	VARCHAR	35		Address	LB-SUB-ADDR
TOWN	VARCHAR	35		Town	LB-SUB-TOWN
COUNTY	VARCHAR	35		County	LB-SUB-COUNTY
POST_CODE	VARCHAR	10		Post code	LB-SUB-POST
COUNTRY	VARCHAR	35		Country	LB-SUB-COUNTRY
STATEMENT_HEADER	VARCHAR	30		Stmt header	LB-SUB-HEADER
TELEPHONE_NUMBER	VARCHAR	20		Telephone	LB-SUB-PHONE
FAX_NUMBER	VARCHAR	20		Fax	LB-SUB-TELEX
ACCOUNT_LANGUAGE	VARCHAR	2		Account language	LB-SUB-ACC-LANG
<b>Version 6.6 and above</b>					
COMPANY_NUMBER	VARCHAR	25		Company number	LB-SUB-COMPANY-NUM
<b>Version 6.7 and above</b>					
PREFERRED_MEDIA	VARCHAR	12		Pref media	LB-SUB-MEDIA
BANK_ID	SMALLINT	2		Address no (Bank Address data)	LB-SUB-BADDID
BANK_ADDRESS_LINE_1	VARCHAR	35		Bank address	LB-SUB-BADDR
BANK_ADDRESS_LINE_2	VARCHAR	35		Bank address	LB-SUB-BADDR
BANK_ADDRESS_LINE_3	VARCHAR	35		Bank address	LB-SUB-BADDR
BANK_ADDRESS_LINE_4	VARCHAR	35		Bank address	LB-SUB-BADDR
BANK_TOWN	VARCHAR	35		Town	LB-SUB-BTOWN
BANK_COUNTY	VARCHAR	35		County	LB-SUB-BCOUNTY
BANK_POST_CODE	VARCHAR	10		Post code	LB-SUB-BPOST
BANK_COUNTRY	VARCHAR	35		Country	LB-SUB-BCOUNTRY
CREATED_USER	VARCHAR	20		[CODA internal]	-
CREATED_DATE	DATE	6		[CODA internal]	-
MODIFIED_USER	VARCHAR	20		[CODA internal]	LB-SUB-MUSER
MODIFIED_DATE	DATE	6		[CODA internal]	LB-SUB-MDATE
<b>Version 7.2</b>					
INTEREST_RATE	DOUBLE	8			LB-SUB-INTCALC-PCENT
<b>Version 7.3</b>					
CREDIT_TRADE_LIMITS	VARCHAR	1		Limits Home or Acc	LB-SUB-CURR-LIMIT
<b>Version 8.0</b>					
ON_HOLD	VARCHAR	1		On hold	LB-SUB-ONHOLD
IBAN_CODE	VARCHAR	36		IBAN Code	LB-SUB-IBAN

Note 1. The default for the ledger column is the space character followed by four zeros.

Note 2. If this account is associated with more than one account group, this column will contain the value “\*\* many \*\*”. If you want to insert account group data, and if there is more than one account group for the account, then you must use the GROUP\_HIERARCHY table to insert the data.

<b>Table 41. NAME index details</b>				
<b>Index name</b>	<b>No</b>	<b>Dup</b>	<b>Coll</b>	<b>Fields in index</b>
NAME_001	1	No	A	SUBACCOUNT
NAME_002	2	Yes	A	LEDGER

<b>Table 42. NAME_ADDRESS column details</b>					
Column name	Data type	Size	Def	CODA name	IASLINK name
<b>Version 6.5 and above</b>					
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-SUB-CODE
ADDRESS_ID	SMALLINT	2		Address no (Address data)	LB-SUB-ADDID
ADDRESS_FORMAT	VARCHAR	2		Address format type	LB-SUB-AFMT
FOR_THE_ATTENTION_OF	VARCHAR	40		For atn of	LB-SUB-FAO
ADDRESS_LINE_1	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_2	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_3	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_4	VARCHAR	35		Address	LB-SUB-ADDR
TOWN	VARCHAR	35		Town	LB-SUB-TOWN
COUNTY	VARCHAR	35		County	LB-SUB-COUNTY
POST_CODE	VARCHAR	10		Post code	LB-SUB-POST
COUNTRY	VARCHAR	35		Country	LB-SUB-COUNTRY

Note. The NAME\_ADDRESS table is read-only. It gives access to ADDRESS\_IDs greater than 0.

<b>Table 43. NAME_ADDRESS index details</b>				
Index name	No	Dup	Coll	Fields in index
NAME_ADDRESS_001	1	No	A	SUBACCOUNT, ADDRESS_ID

<b>Table 44. NAME_BANK column details</b>					
Column name	Data type	Size	Def	CODA name	IASLINK name
<b>Version 6.5 and above</b>					
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-SUB-CODE
BANK_ID	SMALLINT	2		Address no (Bank Address data)	LB-SUB-BADDID
BANK_FORMAT	VARCHAR	2		Format code	LB-SUB-BFMT
ADDRESS_LINE_1	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_2	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_3	VARCHAR	35		Address	LB-SUB-ADDR
ADDRESS_LINE_4	VARCHAR	35		Address	LB-SUB-ADDR
TOWN	VARCHAR	35		Town	LB-SUB-TOWN
COUNTY	VARCHAR	35		County	LB-SUB-COUNTY
POST_CODE	VARCHAR	10		Post code	LB-SUB-POST
COUNTRY	VARCHAR	35		Country	LB-SUB-COUNTRY
ACCOUNT_NUMBER	VARCHAR	20		Bank account number	LB-SUB-ACT-NUM

**Table continued on next page**

NAME_BANK column details (continued)					
ACCOUNT_NAME	VARCHAR	40		Account name	LB-SUB-ACT-NAM
ACCOUNT_REFERENCE	VARCHAR	20		Account reference	LB-SUB-ACT-REF
Version 8.0					
IBAN_CODE	VARCHAR	36		IBAN Code	LB-SUB-IBAN

Note. The NAME\_BANK table is read-only. It gives access to BANK\_IDs greater than 0.

Table 45. NAME_BANK index details				
Index name	No	Dup	Coll	Fields in index
NAME_BANK_001	1	No	A	SUBACCOUNT, BANK_ID

Table 46. NAME_COMMENT column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

Table 47. NAME_COMMENT index details				
Index name	No	Dup	Coll	Fields in index
NAME_COMMENT_001	1	No	A	SUBACCOUNT, COMMENT_NUMBER

Table 48. NAME_CONTACT column details					
Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

Table 49. NAME_CONTACT index details				
Index name	No	Dup	Coll	Fields in index
NAME_CONTACT_001	1	No	A	SUBACCOUNT, COMMENT_NUMBER

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

Index name	No	Dup	Coll	Fields in index
NAME_LEDGER_COMMENT_001	1	No	A	SUBACCOUNT, COMMENT_NUMBER

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-OWN-CODE
NOT_OR_ONLY	VARCHAR	1		Not (Nominal/ledger list)	LB-OWN-NOT
OWNER_TYPE	VARCHAR	1		Type (Nominal/ledger list)	LB-OWN-TYPE
OWNER_FROM	VARCHAR	12		From (Nominal/ledger list)	LB-OWN-FROM
OWNER_TO	VARCHAR	12		To (Nominal/ledger list)	LB-OWN-TO
LAST_YEAR_TURNOVER_HOME	DOUBLE	8		Turnover Last year	LB-OWN-LAST-HOME
PREVIOUS_YEAR_TURNOVER_HOME	DOUBLE	8		Turnover Previous year	LB-OWN-PREV-HOME
LAST_YEAR_TURNOVER_ACCOUNT	DOUBLE	8		Turnover Last year	LB-OWN-LAST-ACNT
PREVIOUS_YEAR_TURNOVER_ACCOUNT	DOUBLE	8		Turnover Previous year	LB-OWN-PREV-ACNT
TERMS_CODE	VARCHAR	1	R	Terms code	LB-OWN-TERMS
PAYMENT_DISCOUNT_METHOD	VARCHAR	1	N	Discount	LB-OWN-DISC-TYPE
EARLY_TERMS_DATE	VARCHAR	4		Early terms	LB-OWN-EDATE
EARLY_TERMS_DISCOUNT	INTEGER	4		Early terms	LB-OWN-EDISC
NORMAL_TERMS_DATE	VARCHAR	4	0100	Normal terms	LB-OWN-NDATE
NORMAL_TERMS_DISCOUNT	INTEGER	4		Normal terms	LB-OWN-NDISC
LIST_NUMBER	INTEGER	4		[field not named]	-

Index name	No	Dup	Coll	Fields in index
NAME_OWNER_001	1	No	A	SUBACCOUNT, LIST_NUMBER



Column name	Data type	Size	Def	CODA name	IASLINK name
NOMINAL	VARCHAR	12		Identification code	LB-NOM-CODE
LEDGER [note 1]	VARCHAR	5	0000	Ledger code	LB-NOM-LEDGER
FULL_NAME	VARCHAR	40		Nominal name	LB-NOM-NAME
SHORT_NAME	VARCHAR	20		Short name	LB-NOM-SHORT
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-NOM-SECLVL
TAX_CODE	VARCHAR	4		Tax code	LB-NOM-TAX
EXPIRY_DATE	DATE	6		Expiry date	LB-NOM-EXPIRY
ACCOUNT_TYPE	VARCHAR	1	T	Account type	LB-NOM-TYPE
CURRENCY	VARCHAR	4		Currency	LB-NOM-CURRENCY
DEFAULT_STATUS	VARCHAR	1	X	Default posting sts	LB-NOM-DEFSTS
POST_IF_ZERO	VARCHAR	1	N	Post details if zero	LB-NOM-PDIZ
DESCRIPTION_REQUIRED	VARCHAR	1	N	Description required	LB-NOM-DESCR
RECONCILABLE	VARCHAR	1	N	Reconcilable	LB-NOM-RECON
CHECK_CREDIT_LIMIT	VARCHAR	1	N	Check credit limits	LB-NOM-CHKCLIM
CHECK_TRADING_LIMIT	VARCHAR	1	N	Check credit limits	LB-NOM-CHKTLIM
QUANTITY_1_TITLE	VARCHAR	20		Quantity titles	LB-NOM-Q1TITL
QUANTITY_1_REQUIRED	VARCHAR	1	N	Quantity required	LB-NOM-Q1REQD
QUANTITY_1_BUDGET_CODE	VARCHAR	1		Quantity budg codes	LB-NOM-Q1BUDG
QUANTITY_2_TITLE	VARCHAR	20		Quantity titles	LB-NOM-Q2TITL
QUANTITY_2_REQUIRED	VARCHAR	1	N	Quantity required	LB-NOM-Q2REQD
QUANTITY_2_BUDGET_CODE	VARCHAR	1		Quantity budg codes	LB-NOM-Q2BUDG
ACCOUNT_GROUP [note 2]	VARCHAR	12		Account groups	LB-NOM-GROUP
CREATED_USER	VARCHAR	20		[CODA internal]	-
CREATED_DATE	DATE	6		[CODA internal]	-
MODIFIED_USER	VARCHAR	20		[CODA internal]	LB-NOM-MUSER
MODIFIED_DATE	DATE	6		[CODA internal]	LB-NOM-MDATE

Note 1. The default for the ledger column is the space character followed by four zeros.

Note 2: If this account is associated with more than one account group, this column will contain the value “\*\* many \*\*”. If you want to insert account group data, and if there is more than one account group for the account, then you must use the GROUP\_HIERARCHY table to insert the data.

Index name	No	Dup	Coll	Fields in index
NOMINAL_001	1	No	A	NOMINAL
NOMINAL_002	2	Yes	A	LEDGER

Column name	Data type	Size	Def	CODA name	IASLINK name
NOMINAL	VARCHAR	12		Identification code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

<b>Table 57. NOMINAL_COMMENT index details</b>				
Index name	No	Dup	Coll	Fields in index
NOMINAL_COMMENT_001	1	No	A	NOMINAL, COMMENT_NUMBER

<b>Table 58. SUBACCOUNT column details</b>					
Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-SUB-CODE
LEDGER	VARCHAR	5		Subanalyse as	LB-SUB-LEDGER
SUBACCOUNT_TYPE	VARCHAR	1		[CODA internal]	-
TITLE	VARCHAR	40		Title	LN-SUB-NAME
SHORT_TITLE	VARCHAR	20		Short name	LB-SUB-SHORT
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-SUB-SECLVL
TAX_CODE	VARCHAR	4		Tax code	LB-SUB-TAX
EXPIRY_DATE	DATE	6		Expiry date	LB-SUB-EXPIRY
ACCOUNT_TYPE	VARCHAR	1	T	A/c type	LB-SUB-TYPE
CURRENCY_CODE	VARCHAR	4		Currency	LB-SUB-CURNCY
RECONCILABLE	VARCHAR	1	N	Reconcilable	LB-SUB-RECON
ONE_OFF_SUPPLIERS	VARCHAR	1	N	One off Suppliers	LB-SUB-EXTADR
ACCOUNT_GROUP [note 1]	VARCHAR	12		Account groups	LB-SUB-GROUP
LEVEL3_ACCOUNT_GROUP	VARCHAR	12		Account groups Lvl3	LB-SUB-GROUP-LV3
CREATED_USER	VARCHAR	20		[CODA internal]	-
CREATED_DATE	DATE	6		[CODA internal]	-
MODIFIED_USER	VARCHAR	20		[CODA internal]	-
MODIFIED_DATE	DATE	6		[CODA internal]	-

Note 1: If this account is associated with more than one account group, this column will contain the value "\*\*\* many \*\*\*". If you want to insert account group data, and if there is more than one account group for the account, then you must use the GROUP\_HIERARCHY table to insert the data.

<b>Table 59. SUBACCOUNT index details</b>				
Index name	No	Dup	Coll	Fields in index
SUBACCOUNT_001	1	No	A	SUBACCOUNT
SUBACCOUNT_002	2	Yes	A	LEDGER

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-COMM-CODE
COMMENT	VARCHAR	78		Account comments	LB-COMM-TEXT
COMMENT_NUMBER	INTEGER	4		[field not named]	-

Index name	No	Dup	Coll	Fields in index
SUBACCOUNT_COMMENT_001	1	No	A	SUBACCOUNT, COMMENT_NUMBER

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-SUB-CODE
LEDGER	VARCHAR	5		Subanalyse as	LB-SUB-LEDGER
TITLE	VARCHAR	40		Title	LN-SUB-NAME
SHORT_TITLE	VARCHAR	20		Short name	LB-SUB-SHORT
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-SUB-SECLVL
TAX_CODE	VARCHAR	4		Tax code	LB-SUB-TAX
EXPIRY_DATE	DATE	6		Expiry date	LB-SUB-EXPIRY
ACCOUNT_TYPE	VARCHAR	1	T	A/c type	LB-SUB-TYPE
CURRENCY_CODE	VARCHAR	4		Currency	LB-SUB-CURNCY
RECONCILABLE	VARCHAR	1	N	Reconcilable	LB-SUB-RECON
ONE_OFF_SUPPLIERS	VARCHAR	1	N	One off Suppliers	LB-SUB-EXTADR
ACCOUNT_GROUP	VARCHAR	12		Account groups	LB-SUB-GROUP
LEVEL3_ACCOUNT_GROUP	VARCHAR	12		Account groups Lv13	LB-SUB-GROUP-LV3
CREATED_USER	VARCHAR	20		[CODA internal]	-
CREATED_DATE	DATE	6		[CODA internal]	-
MODIFIED_USER	VARCHAR	20		[CODA internal]	-
MODIFIED_DATE	DATE	6		[CODA internal]	-
SUBACCOUNT_TYPE	VARCHAR	1		[CODA internal]	-

Note: The SUBACCOUNT\_LOOKUP table is read-only.

Index name	No	Dup	Coll	Fields in index
SUBACCOUNT_LOOKUP_001	1	No	A	SUBACCOUNT
SUBACCOUNT_LOOKUP_002	2	Yes	A	LEDGER

Column name	Data type	Size	Def	CODA name	IASLINK name
SUBACCOUNT	VARCHAR	12		Sub-account code	LB-OWN-CODE

<b>LIST_NUMBER</b>	INTEGER	4		[field not named]	-
NOT_OR_ONLY	VARCHAR	1		Not (Nominal/ledger list)	LB-OWN-NOT
OWNER_TYPE	VARCHAR	1		Type (Nominal/ledger list)	LB-OWN-TYPE
OWNER_FROM	VARCHAR	12		From (Nominal/ledger list)	LB-OWN-FROM
OWNER_TO	VARCHAR	12		To (Nominal/ledger list)	LB-OWN-TO

**Table 65. SUBACCOUNT\_OWNER index details**

Index name	No	Dup	Coll	Fields in index
SUBACCOUNT_OWNER_001	1	No	A	SUBACCOUNT, LIST_NUMBER

**Table 66. TAX\_CODE column details****a. CODA versions 6.4 to 7.1 only**

Column name	Data type	Size	Def	CODA name	IASLINK name
<b>TAX_CODE</b>	VARCHAR	4		Identification code	LB-TAXCODE-CODE
TITLE	VARCHAR	30		Tax name	LB-TAXCODE-TITLE
TAXRATE	DOUBLE	8		Tax rate	LB-TAXCODE-RATE
NOMINAL	VARCHAR	12		Associated account	LB-TAXCODE-NOM
SUBACCOUNT	VARCHAR	12		Associated account	LB-TAXCODE-SUB
LEVEL3	VARCHAR	12		Associated account	LB-TAXCODE-LV3
SECURITY_LEVEL	VARCHAR	1		Security level	LB-TAXCODE-SECLVL

**Table continued on next page**

TAX_CODE column details (continued)					
Column name	Data type	Size	Def	CODA name	IASLINK name
TAX_TYPE	VARCHAR	1	N	Tax type	LB-TAXCODE-TYPE
<b>TAX_CODE</b>	VARCHAR	4		Identification code	LB-TAXCODE-CODE
TITLE	VARCHAR	30		Tax name	LB-TAXCODE-TITLE
SECURITY_LEVEL	VARCHAR	1	3	Security level	LB-TAXCODE-SECLVL
REVERSE	VARCHAR	1	N	Reversible	LB-TAXCODE-REVERSE
NOMINAL_1	VARCHAR	12		[note 1]	LB-TAXCODE-NOM
SUBACCOUNT_1	VARCHAR	12		[note 1]	LB-TAXCODE-SUB
LEVEL3_1	VARCHAR	12		[note 1]	LB-TAXCODE-LV3
NOMINAL_2	VARCHAR	12		[note 1]	LB-TAXCODE-NOM
SUBACCOUNT_2	VARCHAR	12		[note 1]	LB-TAXCODE-SUB
LEVEL3_2	VARCHAR	12		[note 1]	LB-TAXCODE-LV3
NOMINAL_3	VARCHAR	12		[note 1]	LB-TAXCODE-NOM
SUBACCOUNT_3	VARCHAR	12		[note 1]	LB-TAXCODE-SUB
LEVEL3_3	VARCHAR	12		[note 1]	LB-TAXCODE-LV3
ACCOUNT_DESC_1	VARCHAR	30		[note 1]	LB-TAXCODE-ACCOUNT-DESC
ACCOUNT_DESC_2	VARCHAR	30		[note 1]	LB-TAXCODE-ACCOUNT-DESC
ACCOUNT_DESC_3	VARCHAR	30		[note 1]	LB-TAXCODE-ACCOUNT-DESC
TAXRATE_1	DOUBLE	8		Tax Rate	LB-TAXCODE-RATE
TAXPERCENT_1	DOUBLE	8		Non-recoverable %	LB-TAXCODE-PERCENT
TAX_EFF_DATE_1	DATE	6		Effective date	LB-TAXCODE-EFF-DATE
TAXRATE_2	DOUBLE	8		Tax Rate	LB-TAXCODE-RATE
TAXPERCENT_2	DOUBLE	8		Non-recoverable %	LB-TAXCODE-PERCENT
TAX_EFF_DATE_2	DATE	6		Effective date	LB-TAXCODE-EFF-DATE
TAXRATE_3	DOUBLE	8		Tax Rate	LB-TAXCODE-RATE
TAXPERCENT_3	DOUBLE	8		Non-recoverable %	LB-TAXCODE-PERCENT
TAX_EFF_DATE_3	DATE	6		Effective date	LB-TAXCODE-EFF-DATE
TAXRATE_4	DOUBLE	8		Tax Rate	LB-TAXCODE-RATE
TAXPERCENT_4	DOUBLE	8		Non-recoverable %	LB-TAXCODE-PERCENT
TAX_EFF_DATE_4	DATE	6		Effective date	LB-TAXCODE-EFF-DATE
VALUE_ROUND_TYPE	VARCHAR	1		Value rounding type	LB-TAXCODE-VAL-ROUND-TYPE
VALUE_ROUND_DIGIT	VARCHAR	1	2	Value rounding digit	LB-TAXCODE-VAL-ROUND-DIGIT
RATE_ROUND_TYPE	VARCHAR	1		Gross rounding type	LB-TAXCODE-RATE-ROUND-TYPE
RATE_ROUND_DIGIT	VARCHAR	1	9	Gross rounding digit	LB-TAXCODE-RATE-ROUND-DIGIT

Note 1. There are three account description columns (ACCOUNT\_DESC\_n). Related to each of these is a set of N/S/L3 columns. The ACCOUNT\_DESC column contains the field label that CODA uses to identify a N/S/L3 combination. These field labels vary depending upon the context. For example, ACCOUNT\_DESC\_1 can contain “Associated account” or “VAT account”. From the CODA Tax maintenance - List screen, these fields are labelled “Account descriptions”.

<b>Table 67. TAX_CODE index details</b>				
<b>Index name</b>	<b>No</b>	<b>Dup</b>	<b>Coll</b>	<b>Fields in index</b>
TAXCODE_001	1	No	A	TAX_CODE

<b>Table 68. TEXT column details</b>					
<b>Column name</b>	<b>Data type</b>	<b>Size</b>	<b>Def</b>	<b>CODA name</b>	<b>IASLINK name</b>
TEXT_KEY	INTEGER	4		Number, To	LB-TEXT-KEY
LANG	VARCHAR	2		Language	LB-TEXT-LANG
TEXT	VARCHAR	80		[field not named]	LB-TEXT-TEXT

<b>Table 69. TEXT index details</b>				
<b>Index name</b>	<b>No</b>	<b>Dup</b>	<b>Coll</b>	<b>Fields in index</b>
TEXT_001	1	No	A	TEXT_KEY, LANG

## CHAPTER 7

# Using Easysoft ODBC

The aims of this tutorial are to show you how to:

- connect to CODA data using an ODBC-compliant application
- set up masters and account groups using Easysoft ODBC and an application of your choice
- post data to the books using Easysoft ODBC and an application of your choice

It is assumed that you know the rudiments of CODA and that you can use an ODBC-compliant application such as Microsoft Access or Microsoft Excel. (This tutorial uses Access, but you can use any ODBC-compliant application that you are familiar with).

The following prerequisites are needed:

- read and write privileges on CODA data (if you have only read privileges, then you can only view the data, so only the first section is relevant to you)
- sufficient privileges to create a CODA company (or this should be set up for you), and the existence of a CODA bank which can be used by the accounts (in this tutorial, we set up such a bank using CODA)
- the Easysoft Server Component installed and licensed on the server
- Easysoft Client Component installed on the PC and a data source configured. Here we use a data source called CODA (see Table 2 in chapter 5))

## *Connecting to CODA*

This section gives a step-by-step example of how to access server data using Microsoft Access 7.0, which is an ODBC-compliant application.

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 24) appears.

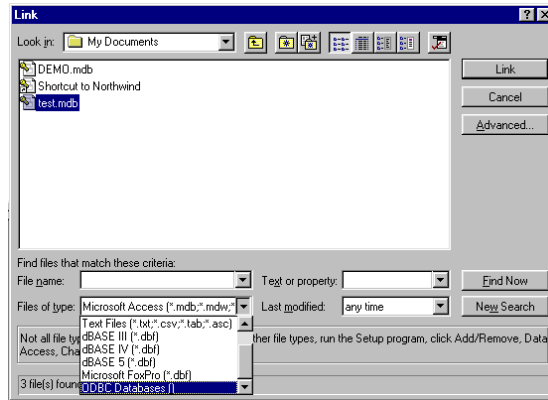


Figure 24.

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 25) appears.

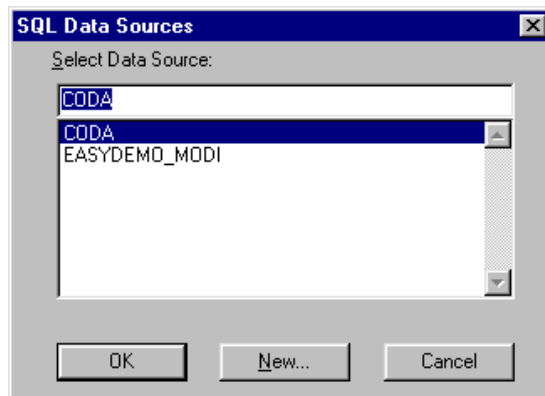
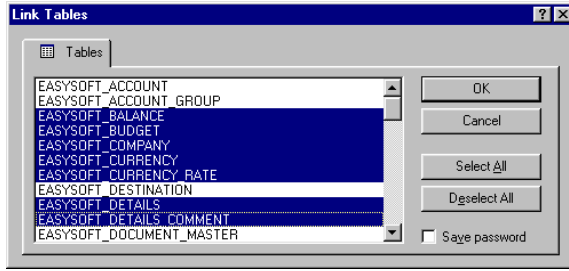


Figure 25.

SQL Data Sources dialog box

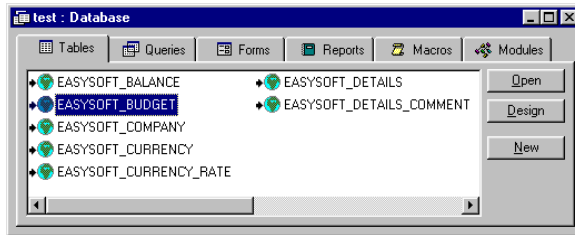
4. Select **CODA** as the data source and click **OK**. (If the Easysoft ODBC Login Prompt dialog box appears, click **OK**). The Link Tables dialog box (Figure 26) appears.





**Figure 26.**  
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 27) contains a list of linked tables.



**Figure 27.**  
Database dialog box

6. 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 CODA data in Microsoft Access (Figure 28).

	NOMINA	SUBACCOUNT	LEVEL3	BUDGET_YEAR	BUDGET_COD	ACCOUNT_BAL	ADJUSTMEN	FINAL_VALUE	OPENING_VAL	TOTAL_VALUE	VALUE_0001
	PARADIX	95 A	N			0	0	0		-2135	0
	ENICANTA	95 A	N			0	0	0		-6618	0
	QUATROPRO	95 A	N			0	0	0		-5040	0
	FORESTATREI	95 A	N			0	0	0		-52540	0
	POWBOOKS	95 A	N			0	0	0		-969184	0
	IMPROMPTU	95 A	N			0	0	0		-2999.9	0
	CORELERAW	95 A	N			0	0	0		-20640	0
	POWERBILDI	95 A	N			0	0	0		-6499.67	0
	WORDPERFEC	95 A	N			0	0	0		-2100	0
	VISUALBASIC	95 A	N			0	0	0		-7130	0
	VISUALC++	95 A	N			0	0	0		-4500	0
	MYSUB	96 F	N	4		0.3	20	124		9.95	
	MYSUB	96 G	N			0	0	0		100	9.95
	MYSUB	96 H	N			0	0	0		200	6.15
	PURCHA	95 A	N			0	0	0		-145675	-60000
	PURCHA FRONTLINE	95 A	Y			0	0	0		-145675	-60000
	PURCHA FRONTLINE	95 B	Y			0	0	0		-594.49	-50
	PURCHA FRONTLINE	96 B	Y			0	0	0		-163992.5	-66000
	PURCHA FRONTLINE	97 B	Y			0	0	0		-180390.75	-72000
	PURCHA FRONTLINE	98 B	Y			0	0	0		-198418.63	-79860
	PURCHA FRONTLINE	99 B	Y			0	0	0		-216260.71	-87846
	Sales	95	N			0	0	0		-365660.34	-145675

**Figure 28.**  
CODA data in Access

## Creating the Master Files

Before using Easysoft ODBC for CODA to set up the Master files and input data, there are a few preparatory steps which must be taken.

### Initialise the System

- Create a new CODA company called NEWCO. You must do this from within CODA (MA GM CO AD). It is not permitted to create a CODA company using Easysoft.

If you don't want to create a new company, you can use an existing company as long as you ensure that there is no conflict between the names used in this tutorial, and names already known to the system; you may have to modify the chart of accounts (see next section), but the principles remain the same.

- To create some of the masters, you need access to a bank that is defined in the CODA system. Either use an existing bank, or create a new one (MA GM BA AD). All the examples below use a bank called NEWBANK.
- Before data can be posted to the books (in non-accounting terms, input), one or more document masters must be created. These define the way in which data is input into the CODA system. You are not allowed to insert or delete records in the DOCUMENT\_MASTER table using Easysoft ODBC for CODA, so create a new document master using CODA (MM DO AD). If a purchase invoice called PINV already exists, you can use this.

On the first CODA screen enter the following:

Document type:	<b>PINV</b>	(this is the name of the invoice)
Format:	<b>IN</b>	(it is an invoice)
Description:	<b>Purchase invoice for NEWCO</b>	

Accept all the other defaults on this and the following screens.

- Once these steps have been completed, use Easysoft ODBC for CODA to check that the new data exists. Connect to the data source which holds CODA data (see previous section) and open the COMPANY table.

*NEWCO doesn't appear.* This is because the Easysoft catalog does not know about it. Run the Codacat routine (see "Adding CODA Companies", page 37) and add the company. Now look at the data in the COMPANY table again -

NEWCO should be visible (you may have to close the table and re-open it, or refresh the data, depending upon the nature of the application you are using).

To see the PINV invoice open the DOCUMENT\_MASTER table. You should see the PINV document master as one of the records in the table.

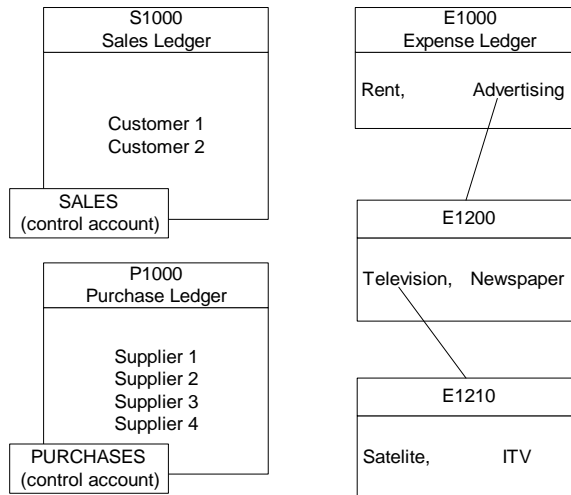
There is no table corresponding directly to the bank data, so this cannot be seen using Easysoft ODBC for CODA at the current stage.

## Master Files

We will set up the Master files for a CODA company called NEWCO using Easysoft ODBC. Its chart of accounts is shown in the table and figure below. This chart of accounts has been designed to show most of the features available using CODA and Easysoft; it is not necessarily representative of a real-world chart of accounts.

### NEWCO Chart of Accounts

E1000 EXPENSES	P1000 PURCHASES
E1200 ADVERTISING	SUPP1      (customer 1)
NEWSPAPER	SUPP2      (customer 2)
E1210 TELEVISION	SUPP3      (customer 3)
ITV	SUPP4      (customer 4)
SATELITE	
RENT	S1000 SALES
	CUST1      (customer 1)
	CUST2      (customer 2)

**Figure 29.****NEWCO Chart of Accounts**

In the examples, in the general case only the minimum information needed to create the objects in the system is shown. You can, if you wish, fill in all the details (for example, full addresses, telephone numbers, comments).

- Create the ledgers. Open the LEDGER table and enter the following information in the fields.

Ledger	Nominal	Description
S1000	SALES	SALES LEDGER
P1000	PURCHASES	PURCHASE LEDGER
E1000		EXPENSES LEDGER
E1200		ADVERTS LEDGER
E1210		TV LEDGER

Nominal refers to the Control Account associated with Sales and Purchase ledgers. Since neither of these nominals yet exist, they will be created automatically by the system. Expense ledgers do not have control accounts, so this column is left empty for the E type ledgers. The other fields in the table (CREATED\_USER, etc.) should not be completed, as the information is entered automatically by CODA. To check this, ensure the details you entered are saved, then re-open the table, and they will be seen.

- Create the Name and Address subaccounts for the SALES and PURCHASE ledgers. Open the NAME table, and enter the following:

Subaccount	Address_ID	Account_name
CUST1	0	NEWBANK
CUST2	0	NEWBANK
SUPP1	0	NEWBANK
SUPP2	0	NEWBANK
SUPP3	0	NEWBANK
SUPP4	0	NEWBANK

SUBACCOUNT is the name of the CODA Name and Address subaccount, ADDRESS\_ID is the Address number (there can be more than one address for a Name and Address account, so CODA uses these numbers). For our purposes, we use just one address (we haven't bothered to enter the details!), and in CODA terms, the first address number is labelled zero. ACCOUNT\_NAME is the name of the bank account associated with the Name and Address account. Either use a bank that exists in your CODA group, or create a new one.

- Review: Look at the chart of accounts in CODA (MA GM CA). Sort the output by NOMINAL, SUBACC and LEVEL3, and in the Print Options show the Groups, Ledgers and Titles. You should see:

Chart of accounts sorted by -		
NOMINAL	SUBACC	LEVEL3
PURCHASES		P1000
SALES		S1000

Only the Purchases and Sales nominals are seen. The Name and Address accounts are not seen because we haven't completed the ownership lists for these. The NAME\_OWNER table holds this information, along with financial information about the account (we won't bother with this).

- Create ownership lists for the customer and supplier Name and Address accounts. Open the NAME\_OWNER table and enter the following:

Subaccount	Owner_type	Owner_from	List_number
CUST1	L	S1000	1
CUST2	L	S1000	1
SUPP1	L	P1000	1
SUPP2	L	P1000	1
SUPP3	L	P1000	1
SUPP4	L	P1000	1

When ownership lists are entered using CODA, you do not see anything on screen which is equivalent to List\_number. However, behind the scenes, the lists must be identified; here we explicitly state the number of each line for the

ownership list of a given subaccount. We must do this, because in some cases, there can be more than one owner.

Save the data, and check the chart of accounts in CODA. You should see:

Chart of accounts sorted by -			
NOMINAL	SUBACC		LEVEL3
PURCHASES			P1000
SUPP1	SUPPLIER 1		
SUPP2	SUPPLIER 2		
SUPP3	SUPPLIER 3		
SUPP4	SUPPLIER 4		
SALES			S1000
CUST1	CUSTOMER 1		
CUST2	CUSTOMER 2		

- Create the RENT and ADVERTISING nominals. Open the NOMINAL table (you will see the Purchase and Sales nominals). Add the following:

Nominal	Ledger	Full_name
RENT		RENT ACCOUNT
ADVERTISING	E1200	ADVERTS ACCOUNT

Ledger refers to the ledger which is used for sub-analysis. RENT cannot be sub-analysed, so leave this blank; on entry, the default ( 0000) is automatically entered. (You can, if you wish, enter this manually). The ADVERTISING account is to be sub-analysed; we use the ledger E1200 to do this.

- Create Subaccounts and Level3s. Open the SUBACCOUNT and enter the following:

Subaccount	Ledger	Short_title
TELEVISION	E1210	TV ADS ACCOUNT
NEWSPAPER		NEWSPAPER
SATELITE		SATELITE TV ADVERTS
ITV		ITV ADVERTS

Ledger refers to the ledger which is used for sub-analysis.

- Create ownership lists for the new subaccounts and level3s. Open SUBACCOUNT\_OWNER and enter the following:

Subaccount	Owner_type	Owner_from	List_number
TELEVISION	L	E1200	1
NEWSPAPER	L	E1200	1
SATELITE	L	E1210	1
ITV	L	E1210	1

- Review: Look at the chart of accounts in CODA (MA GM CA). Sort the output by NOMINAL, SUBACC and LEVEL3, and in the Print Options show the Groups, Ledgers and Titles. You should see:

```

Chart of accounts sorted by -
NOMINAL                SUBACC                LEVEL3
-----
ADVERTISING  ADVERTS ACCOUNT                E1200
  NEWSPAPER  .....
  TELEVISION .....                E1210
    ITV      .....
    SATELITE .....

PURCHASES                P1000
  SUPP1          SUPPLIER 1
  SUPP2          SUPPLIER 2
  SUPP3          SUPPLIER 3
  SUPP4          SUPPLIER 4

RENT                RENT ACCOUNT

SALES                S1000
  CUST1          CUSTOMER 1
  CUST2          CUSTOMER 2

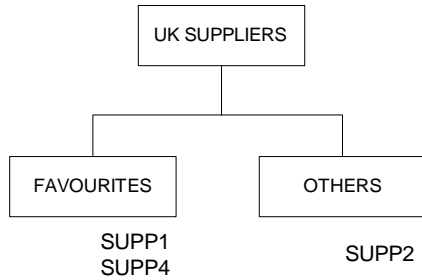
```

All the Masters needed to input data have been created, and assuming that one or more Document Masters has been created using CODA, you could now input to the books. However, before inputting data, let's create a few Hierarchical Account Groups.

## Create Hierarchical Account Groups

This stage is optional; if you wish, you can now enter data into CODA using your ODBC-compliant application in conjunction with Easysoft ODBC for CODA. If you want to set up account groups, follow the instructions in this section. Since the function of this tutorial is to show you how to use Easysoft ODBC, we present just the simplest account group structure; the principles are identical in more complex cases.

In this tutorial, we will create the hierarchical account group structure shown in Figure 30.



**Figure 30.**

**NEWCO account groups**

- Create the account groups. Open ACCOUNT\_GROUP and enter the following:

Account_group	Description	Group_type
UK SUPPLIERS	UK suppliers	S
FAVOURITES	Favourite UK suppliers	S
OTHERS	Other UK suppliers	S

- Now create the hierarchical structure for these account groups and then associate accounts with account groups. Open GROUP\_HIERARCHY and enter the following:

Parent	Child	Child_type	
UK SUPPLIERS	FAVOURITES	G	These lines define the hierarchical structure.
UK SUPPLIERS	OTHERS	G	
FAVOURITES	SUPP4	S	These lines link accounts to groups.
FAVOURITES	SUPP1	S	
OTHERS	SUPP4	S	

Although FAVOURITES and OTHERS are account groups which hold S type accounts, the Child\_type of their parent is G. This is because the children are account groups, not accounts.

Note: in CODA the accounts are included in account groups using the account group list structure in the relevant nominal, subaccount or name and address master.

- Review: Look at the account group hierarchy using CODA (MA AG BH). You should see:



Group	Group Title	Group Type	Relations
FAVOURITES	Favourite UK suppliers	Subaccount	
Parents			
UK SUPPLIERS	UK Suppliers	(Group)	
Children			
SUPP1	SUPPLIER 1	(Subaccount)	
SUPP4	SUPPLIER 4	(Subaccount)	
Group	Group Title	Group Type	Relations
OTHERS	Other UK Suppliers:	Subaccount	
Parents			
UK SUPPLIERS	UK Suppliers	(Group)	
Children			
SUPP2	SUPPLIER 2	(Subaccount)	
Group	Group Title	Group Type	Relations
UK SUPPLIERS	UK Suppliers	Subaccount	
Children			
FAVOURITES	Favourite UK suppliers	(Group)	
OTHERS	Other UK Suppliers:	(Group)	

If you now re-open the ACCOUNT\_GROUP table you will see additional records (Figure 31). The application will not allow you to delete them, and you cannot see them in CODA.

PARENT	CHILD	CHILD_TYPE	PARENT_ID	CHILD_ID
FAVOURITES	SUPP1	S	3	4
FAVOURITES	SUPP4	S	3	18
OTHERS	SUPP2	S	6	5
UK SUPPLIERS	FAVOURITES	G	7	3
UK SUPPLIERS	SUPP1	S	7	4
UK SUPPLIERS	SUPP4	S	7	18
UK SUPPLIERS	OTHERS	G	7	6
UK SUPPLIERS	SUPP2	S	7	5
*				

Record: 1 of 8

Figure 31.

**NEWCO account group data in Access**

*These three records are read only, created by Easysoft.*

So, what are these records, and where do they come from? To aid the production of reports and other forms of analysis, all the accounts in children of an account group are considered to be children of the parent. Therefore, in our example, we see that SUPP1, SUPP2 and SUPP4 are listed as children of UK SUPPLIERS (this kind of analysis is not supported in CODA - furthermore, it is forbidden to mix child types in an account group). This categorisation of accounts is recursive, so, for example, in a three-level hierarchy, the children of the lowest level account group would also be listed as children of the top level account group.

## Post to Books

You can either post to the Intray or directly to the books. In this example, we'll do the latter using the DETAILS table. (To enter data into the CODA Inray, use the INTRAY table).

- Open DETAILS and enter the following:

DT	DN	LN	DD	DV	Nominal	Subaccount	Level3	PF
PINV	1	1	<date>	100	ADVERTISING	TELEVISION	SATELIT E	N
PINV	1	2		-30	PURCHASES	SUPP2		N
PINV	1	3		-20	PURCHASES	SUPP2		N
PINV	1	4		-50	PURCHASES	SUPP2		Y

DT = Document\_type  
 DN = Document\_number  
 LN = Line\_number  
 DD = Document\_date  
 DV = Document\_value  
 PF = Post\_flag

The first line entered above contains the header record information; detail lines are all the other lines. The sum of the values in the detail lines must equal the value specified in the Document\_value column for the header line.

Document\_date only needs one entry on the header record. This date will then be used automatically for the other entries, although initially, you will not see this.

For all but the last line entered, the Post\_flag value should be "N" - if you enter "Y", and then try to add another line, an error occurs. The value for Post\_flag on the last line that you enter must be "Y", otherwise the data will not be entered into the CODA system. Note that the Post\_flag values will later be automatically changed by the system; we will see this in a moment.

- Look at the data in CODA (IN LD). You should see something like this (we've included descriptions here).

```

PINV      2      Status ""
          Doc date 12-Jun-96      Batch      11      Period 96  1
          Entered on 12-Jun-96      by STEVE
          Ext ref      0      Doc ref
          Currency GBP      Author.
    
```

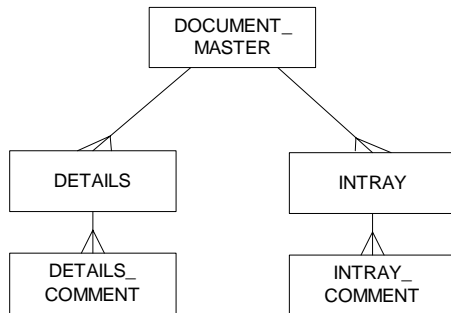
Nominal	Subaccount	Level3 acct	Description	Value	Quantity 1/du	due date	Quantity 2
ADVERTISING	TELEVISION	SATELITE	Satelite account purchases	100.00			X
PURCHASES	SUPP1		Chocolate biscuits	-30.00	31-Jul-96		A
PURCHASES	SUPP1		Ginger biscuits	-20.00	31-Jul-96		A
PURCHASES	SUPP3		Custard creams	-50.00	31-Jul-96		A
Totals:	Home		0.00 Document				0.00

- Refresh the data shown in the DETAILS table (in Microsoft Access close and re-open the table). In addition to the various default values, you will see that all the Post\_flag values are now “Y”. You cannot update or delete any of the records, and neither can you add new records with the same document number as the ones previously entered.

### Intray

The use of the INTRAY table is similar to that of the DETAILS table. The only difference is that you can update existing records (but not the Post\_flag column).

Figure 32 shows the tables that are used for posting to the books.



**Figure 32.**  
Tables used for posting to the books

## Budgets

Use CODA (BU LI) to look at the data you entered to the books using Easysoft. Here we see the first four periods for this:

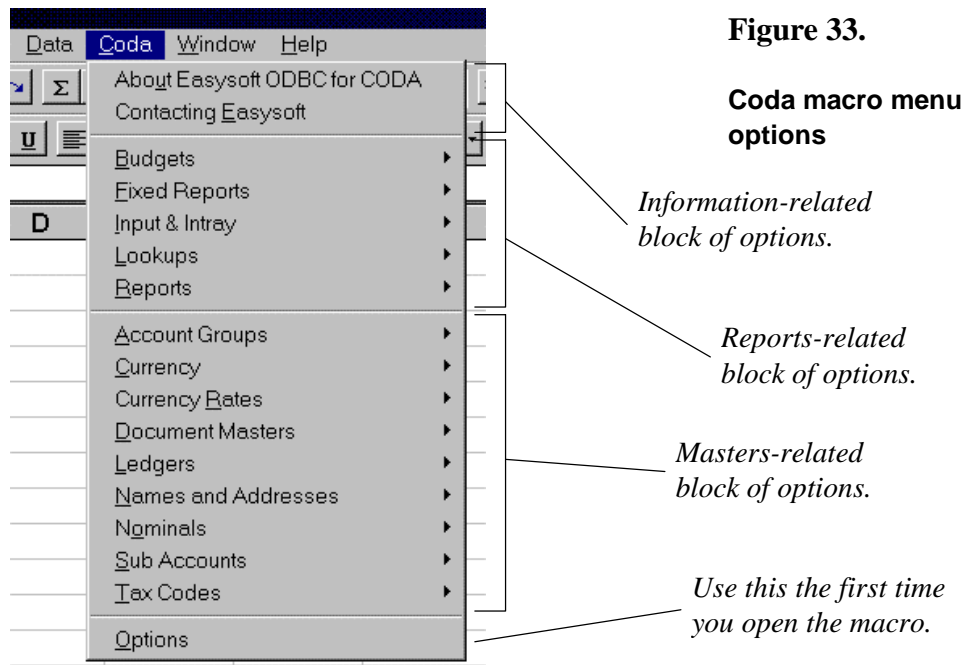
Nominal	ADVERTISING	ADVERTS	ACCOUNT				
Sub-account	TELEVISION	.....					
Level-3	SATELITE	.....					
Budget	A						
Year & period range	96 01	- 4		Total value			100.00
Value	Prd	Value	Prd	Value	Prd	Value	Prd
100.00	1	0.00	2	0.00	3	0.00	4

Open the BUDGET table. You will see records for each combination of year, account and budget code. Period information is stored in the columns labelled VALUE\_n, where n is a period number.

# Easysoft Excel Macro for CODA

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

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



The **Coda** menu option on the menu bar gives access to the top level pulldown shown here. There are four groups of options, related to information about the macro, reports, masters and initialisation. They are discussed later in the following order: initialisation, masters-related options, reports-related options.

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

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

Before the macro can be used, it must be installed, and this is described in the next section.

## Macro Installation

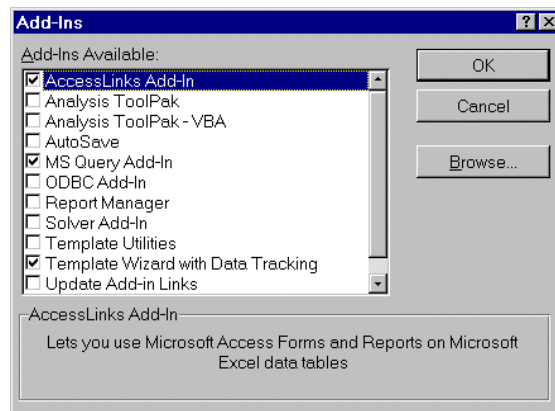
To install the macro, you must add the software as an Excel add-in. Take the following steps:

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, simply copy it to the directory.

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



**Figure 34.**

**Add-ins dialog box**

3. Click the **Browse...** button. The Browse dialog box appears. Navigate to the `c:\easysoft\excel` directory, and highlight the `.XLA` file. Click the **OK** button.

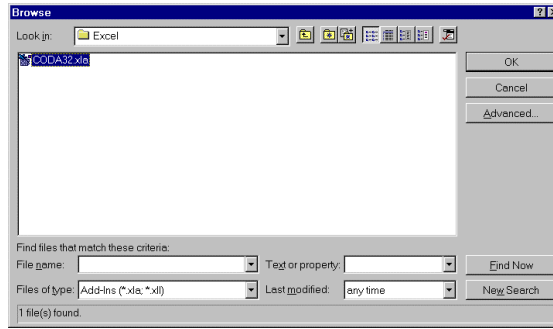
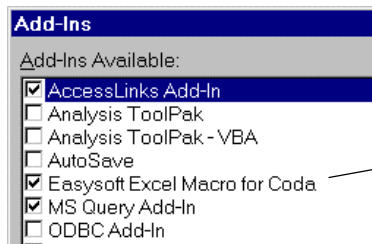


Figure 35.

Browse dialog box

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

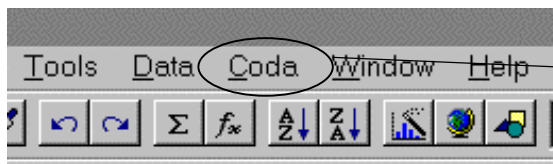


*The Easysoft macro is now accessible.*

Select the CODA add-in.

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

- Click the **OK** button on the Add-Ins dialog box.
- The CODA macro is now available for use by clicking the newly added **Coda** menu which appears on the Excel menu bar.

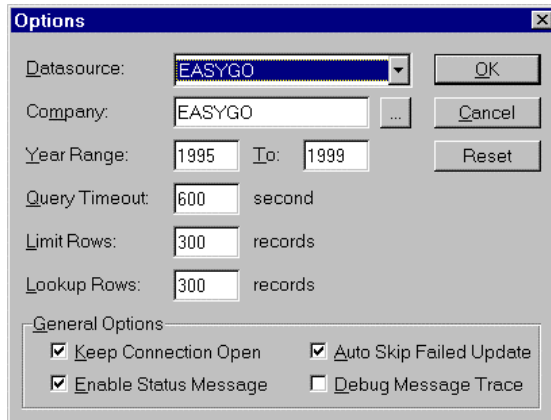


*New Coda option.*

## 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 **Coda** menu. The Options dialog box appears. The various settings you can choose are described in the remainder of this section.



**Figure 36.**

**Options dialog box**

2. Select the required data source from the dropdown list box (in these examples we use EASYGO). Only data sources which use Easysoft drivers are shown in the dropdown list.
3. Enter the CODA company on which you want to operate. You cannot enter data directly into the entry box. Click the  button (lookup button) to see the available companies in a new dialog box.
4. In this example, there is just one CODA company in the database, and it happens to have the same name as the data source. Highlight the company you want and click the **OK** button. The company will be entered into the list box on the Options dialog box.



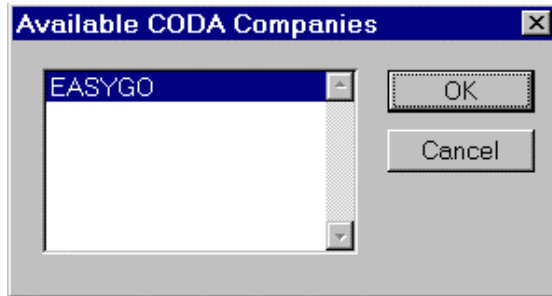
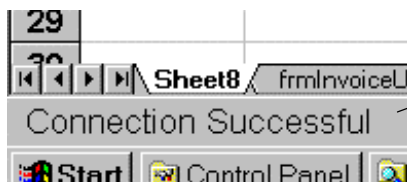


Figure 37.

**Available CODA  
Companies dialog  
box**

5. The year range can be changed by typing in the upper and lower bounds of the years you want to see. By default, the lower value is the current year minus two, and the upper value is the current year plus two. For example, if the current year is 1997, then the range is 1995 to 1999.
6. 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.
7. 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.
8. Lookup Rows is used to limit the number of rows initially returned by lookup queries (“Lookups”, page 132).
9. 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 CODA data, there may not be enough licence slots available. By default, this option is selected.
10. Enable Status Message is used to show messages at the bottom of the worksheet, for example:

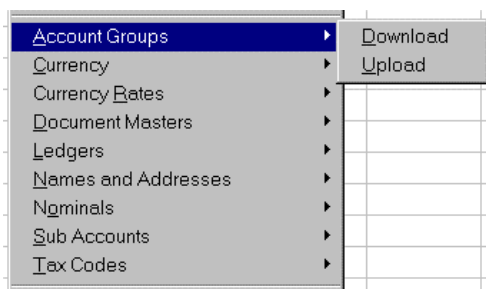


*This is where you will see  
messages.*

By default, this option is selected.

11. Auto Skip Failed update is used to prevent error messages which result from attempts at invalid updates being shown on screen. However, at the end of the update operation, all rows which the user wanted to update, but which were not updated will be flagged. By default, this option is selected.
12. 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.
13. When all the information has been entered, press 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. (Press 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).

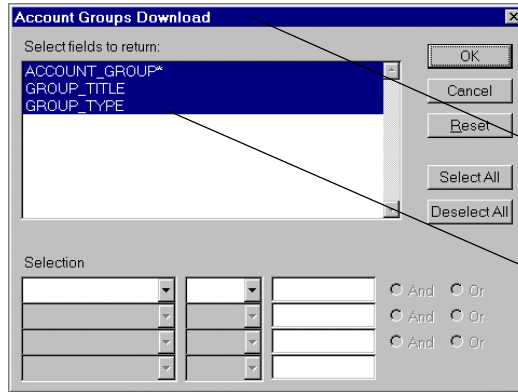
## ***Masters-related options***



All the menu options contained in this group follow a similar pattern. You can either download the data in the workbook, or you can upload it. These options are obtained by selecting the appropriate option as shown below.

### **Download**

If you download data, you will be presented with the Table Download dialog box. The operation of this dialog box is identical for all the masters, the only differences being the name of table and the columns that are available in the list box. For example, when you download account groups, the title of the dialog box is Account Groups Download. In the list box you will see the columns that are in the ACCOUNT\_GROUP table.

**Figure 38.****Download dialog box**

*This changes depending on the table to download.*

*The columns you see here are those available in the table.*

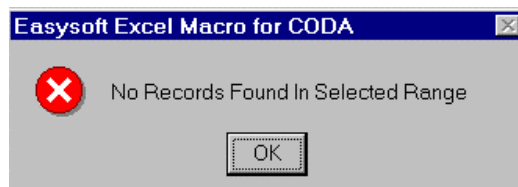
1. Highlight the column(s) that you want to download.

The **Select All** button can be used to select all the columns, and the **Deselect All** button is used to clear the selection.

An asterisk after the column name indicates that the column is part of the primary key.

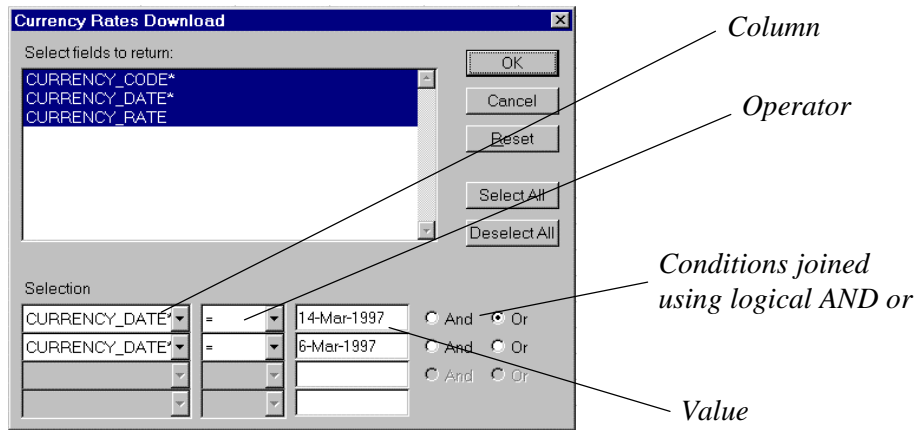
2. Click the **OK** button to proceed with the download, or **Cancel** to abandon the operation.
3. The data will be returned to the worksheet.

If the master does not contain any data, you will see the following message (click **OK** to return to the worksheet).

**Selection criteria**

You can set selection criteria to restrict the rows which are returned. Up to four simple condition clauses can be combined using logical AND and OR. This example uses Currency Rates: we will download all the fields for all currency rates dated either 6<sup>th</sup> March 1997 or 14<sup>th</sup> March 1997.

1. Select Coda, Currency Rates, Download to display the Currency Rates Download dialog box.



2. Select the fields you want to download.
3. Enter the following for the first selection criterion:

Column:        **CURRENCY\_DATE**        (select using dropdown)  
 Operator:     =                            (select using dropdown)  
 Value:        **6-Mar-1997**                (type required value)

4. Select **or** to combine with the second criterion.
5. Enter the following for the second selection criterion:

Column:        **CURRENCY\_DATE**        (select using dropdown)  
 Operator:     =                            (select using dropdown)  
 Value:        **14-Mar-1997**                (type required value)

6. Click the **OK** button to download the data.

## Upload

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

<b>row colour</b>	<b>indicates</b>
red	update failed

blue            row inserted successfully  
 green          row updated successfully

This example is for an upload of currency rates.

	A	B	C	D	E	F	G	H
1	CURRENCY_CODE	CURRENCY_DATE	CURRENCY_RATE					
2	USD	24-Mar-1997	246.24					
3	USD	14-Mar-1997	192.00					
4	USD	1-Jan-1997	205.20					
5	USD	25-Jan-2020	999.00					
6	UDS	2-Jan-1997	203.30					Easysoft Error: IASLINK, I'm sorry, that currency does not exist (7011)
7								

The rates for USD were first downloaded (**Coda, Currency\_Rates, Download**), which resulted in the first three rows on the worksheet. Then two additional rows were created - the second additional row contains a typographical error. After the update was performed, the colours on the worksheet changed. The first three rows are green, indicating successful update. The fourth row is blue, indicating successful insert, the final row is red, indicating the attempted update was not successful. Additionally, the IASLINK error message is shown.

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)

## ***Reports-related options***



The following options are available: Budgets, Fixed Reports, Input and Intray, Lookups and Reports. Within each of these there are a number of options. All of these are described here.

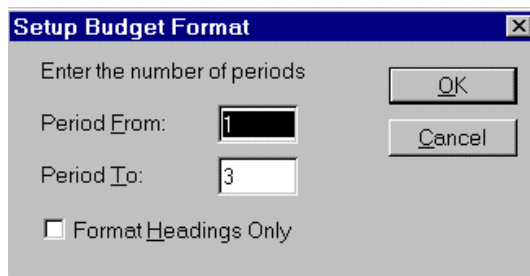
## Budgets



There are three options under the Budgets menu, namely Setup, Download, and Upload.

## Setup

This option is used to set up the upload options.



**Figure 39.**

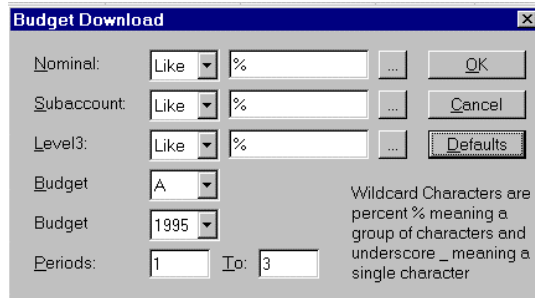
**Setup Budget Format dialog box**

Period From and Period To limit the range of the periods that you want to upload.

The Format Headings Only check box can be used when a change is made to the period range. It prevents existing data in the worksheet from being deleted, since only the column heading is changed.

## Download

The columns on which you can base selection criteria are shown in the dialog box.



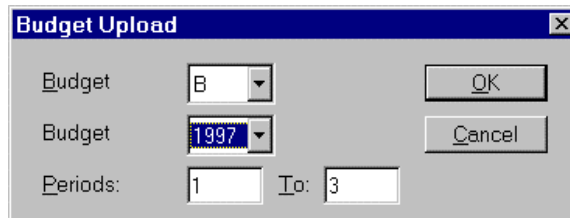
**Figure 40.**

**Budget Download  
dialog box**

The **Defaults** option resets the selection criteria to the default values.

## Upload

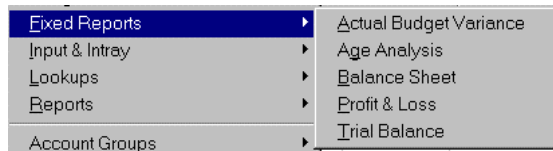
Set the upload options, then click **OK** to proceed, or **Cancel** to quit.



**Figure 41.**

**Budget Upload  
dialog box**

## Fixed Reports



There are five fixed reports: Actual Budget Variance, Age Analysis, Balance Sheet, Profit & Loss and Trial Balance.

## Actual Budget Variance (ABV)

The ABV Report is used to compare the variation between Actuals and Budgets. Set any selection criteria that you require, then click the **OK** button.

 A screenshot of a dialog box titled 'Budget Variance Selection'. The dialog box contains several fields and buttons:
 

- Nominal:** A dropdown menu set to 'Like', followed by a text box containing '6%' and a button with three dots.
- Subaccount:** A dropdown menu set to '<>', followed by an empty text box and a button with three dots.
- Level3:** A dropdown menu set to '=', followed by an empty text box and a button with three dots.
- Budget:** A dropdown menu set to '1997'.
- Periods:** Two text boxes, the first containing '0' and the second containing '3', with 'To:' between them.
- Main Code:** A dropdown menu set to 'A'.
- Comparison:** A dropdown menu set to 'B'.
- Buttons: 'OK', 'Cancel', and 'Defaults'.
- Text: 'Wildcard Characters are percent % meaning a group of characters and underscore \_ meaning a single character'.

**Figure 42.**

**Budget Variance Selection dialog box**

The data is downloaded, and will appear similar to the report shown here.





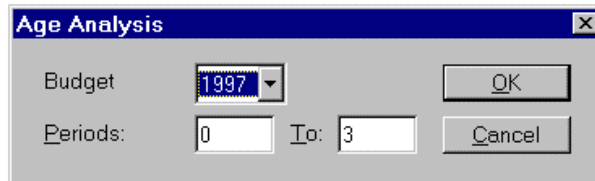
	A	B	C	D	E	F	G	
1	17-Mar-97	<b>Easysoft Travel Company</b>						
2	16:17	<b>Actual Budget Variance</b>						
3		As of March 17, 1997						
6		<b>Nominal</b>	<b>Subaccount</b>	<b>Level 3</b>		<b>Actual (A)</b>	<b>Budget (B)</b>	
7		8100	B0005			-1,281.50	-500.00	
8		8100	B0006			-4,785.00	-4,500.00	
9		8100	B0013			-1,342.00	-2,000.00	
10		8100	B0024			-8,470.00	-7,000.00	
11		8100	B0033			-1,694.00	-100,000.00	
12		8100	B0040			-1,122.00	-400.00	
13		8100	B0049			-830.50	-400.00	
14		8100	B0053			-661.00	-800.00	
15		8100	B0057			-3,355.00	-7,000.00	
16		8100	B0063			-1,540.00	-3,500.00	
17		8100	B0067			-2,300.00	-5,000.00	
18		8100	B0068			-1,100.00	-2,500.00	
19		8100	B0069			-1,850.00	-4,000.00	
20		8100	B0073			-1,760.00	-5,000.00	
21								
22					<b>Totals -</b>	<b>-30,809.50</b>	<b>-142,100.00</b>	

Figure 43.

## Example ABV Report

## Age Analysis

This displays a list of customers who have not paid. Set any selection criteria that you require, then click the **OK** button.



**Age Analysis**

Budget: 1997

Periods: 0 To: 3

OK Cancel

Figure 44.

### Age Analysis dialog box

Two sheets are created; a worksheet and a pivot table. The report defaults to the pivot table sheet, and you will see something similar to this.

	A	B	C	D	E	F
1	17-Mar-97		Easysoft Travel Company			
2	16:28		Age Analysis			
3			As of March 17, 1997			
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						

Sum of VALUE	PERIOD				Grand Total
SUBACCOUNT	1	2	3		
H0001	-1100	-2250	-215		-3565
H0002	-2000	-820	-400		-3220
H0003	-7700	0	0		-7700
H0004	-1540	0	0		-1540
H0005	-320	-700	0		-1020
H0006	-355	0	-400		-755
H0007	-210	-300	0		-510
H0008	-2500	0	-550		-3050
H0009	-1400	0	0		-1400
H0010	-2070	0	0		-2070
H0011	-2500	0	0		-2500

Figure 45.

Example Age Analysis Report

### Balance Sheet

Each category has its own selection criteria.

The dialog box titled "Balance Sheet Selection Criteria" contains the following fields and options:

- Financial Year: 1997 (dropdown)
- Period Range: 0 (From) To: 3 (To)
- Categories list: Fixed Assets, Current Assets, Capital (highlighted), Long Term Liabilities, Current Liabilities
- Selection section with three rows:
  - Row 1: NOMINAL (dropdown), Like (dropdown), 1% (text),  And  Or
  - Row 2: (dropdown), (dropdown), (text),  And  Or
  - Row 3: (dropdown), (dropdown), (text),  And  Or
- Buttons: OK, Cancel, Reset

Figure 46.

Balance Sheet Selection Criteria dialog box

The data is downloaded, and will appear similar to the report shown here.

	A	B	C	D	E	F	G
1	18-Mar-97			Easysoft Travel Company			
2	10:33			<b>Balance Sheet</b>			
3				As of March 18, 1997			
4							
5							
6			ASSETS				
7			Fixed Assets				
8			1110 Fixtures & Fitting Costs			2,000.00	
9			1120 Fixtures & Fitting Deprecia			-200.00	
10			1310 Computer Equipment Costs			3,500.00	
11			1320 Computer Equipment Depr			-600.00	
12			1410 Motor Vehicles Costs			9,980.00	
13			1420 Motor Vehicles Depreciati			-2,000.00	
14			1510 Office Equipment Costs			1,700.00	
15			1520 Office Equipment Deprecia			-600.00	
16							
17			Total Fixed Assets			13,780.00	
18							
19			Current Assets				
20			1110 Fixtures & Fitting Costs			2,000.00	
21			1120 Fixtures & Fitting Deprecia			-200.00	

Figure 47.

**Example Balance Sheet Report**

**Profit and Loss**

Use this report to see the profits and losses. Enter the required criteria, if any, then click the **OK** button.

Figure 48.

**Profit & Loss Selection dialog box**

The example report below was generated using the following criteria:

Category	Selection	Operator	Value
Sales	NOMINAL	like	6%
Purchases	NOMINAL	like	7%
Expenses	NOMINAL	like	8%

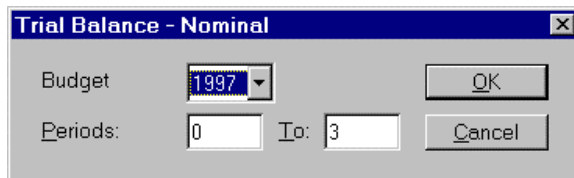
	A	B	C	D	E	F	G	H
1	18-Mar-97	<b>Easysoft Travel Company</b>						
2	10:45	<b>Profit &amp; Loss</b>						
3		As of March 18, 1997						
4								
5								
6								
7		<b>Sales</b>						
8		6100 Sales Analysis - Holidays						-31,891.00
9								
10		<b>Total Sales</b>						-31,891.00
11								
12		<b>Purchases</b>						
13		7100 Purchase Analysis - Holidays						28,930.00
14								
15		<b>Total Purchases</b>						28,930.00
16								
17		<b>Expenses</b>						
18		8200 Salaries						5,680.00
19		8300 Motor Vehicles						135.00
20		8610 Accountant						820.00
21								
22		<b>Total Expenses</b>						6,635.00
23								
24		<b>TOTAL:-</b>						3,674.00

**Figure 49.**

**Example Profit and Loss Report**

### Trial Balance

This displays a trial balance for the selected year and period range. Set the selection criteria that you require, then click the **OK** button.



**Figure 50.**

**Trial Balance dialog box**

The data is downloaded, and will appear similar to the report shown here.

	A	B	C	D	E	F	G
1	17-Mar-97			<b>Easysoft Travel Company</b>			
2	16:58			<b>Trial Balance</b>			
3				As of March 17, 1997			
4				Standard Trial Balance for Year - 1997 Start Period - 0 End Period - 3			
5							
6							
7				<b>Nominal</b>	<b>Description</b>	<b>Debit</b>	<b>Credit</b>
8				1110	Fixtures & Fitting Costs	2,000.00	
9				1120	Fixtures & Fitting Depreciation		200.00
10				1310	Computer Equipment Costs	3,600.00	
11				1320	Computer Equipment Depreciation		600.00
12				1410	Motor Vehicles Costs	9,980.00	
13				1420	Motor Vehicles Depreciation		2,000.00
14				1510	Office Equipment Costs	1,700.00	
15				1520	Office Equipment Depreciation		600.00
16				2100	Sales	31,891.00	
17				2210	Bank - Current Account	32,040.00	
18				3100	Capital		50,000.00
19				5100	Purchases		30,052.12
20				5200	Payroll		1,500.00
21				5300	VAT Control Account	167.12	
22				6100	Sales Analysis - Holidays		31,891.00
23				7100	Purchase Analysis - Holidays	28,930.00	
24				8200	Salaries	5,680.00	
25				8300	Motor Vehicles	136.00	
26				8610	Accountant	820.00	
27							
28					<b>Totals -</b>	<b>116,843.12</b>	<b>116,843.12</b>

**Figure 51.**

**Example Trial Balance**

### Input and Intray



There are four options available: Details Download, Intry Download, Invoice Setup and Post Documents. The operation of the Details Download and Intry Download options is similar, so only the first is shown.

### Details Download

Use this option to download details of the documents. Set any selection criteria you want, then click the **OK** button.

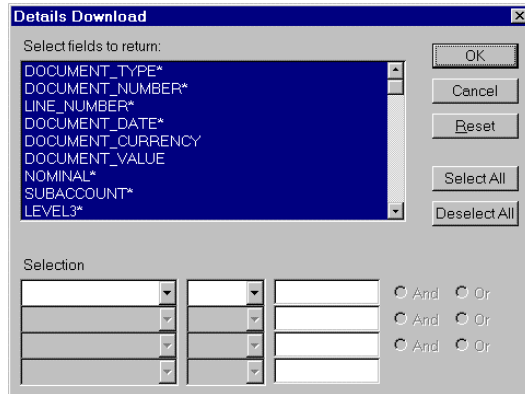


Figure 52.

**Details Download  
dialog box**

## Document Setup

This is a preparatory step, needed before an upload can take place. Select the fields you want to update in CODA.

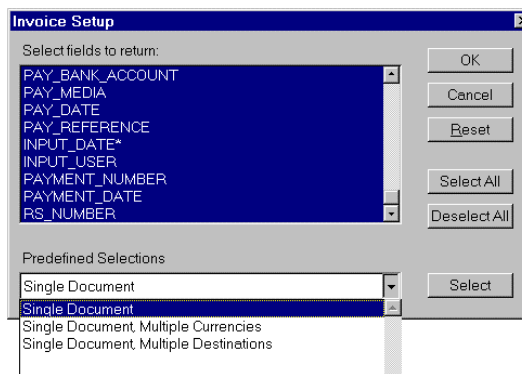


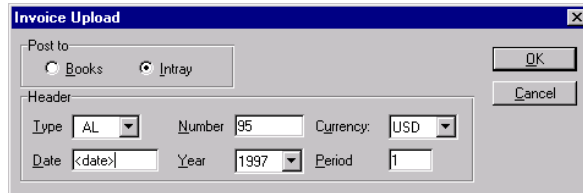
Figure 53.

**Invoice Setup dialog  
box**

The Predefined Selections option can be used to select fields rapidly. Choose the option you require, then click the **Select** button. The fields are automatically selected in the Select fields to return list box.

## Post Documents

When you select this option, the Invoice Upload dialog box appears. Select the options you require, then click the **OK** button.

**Figure 54.****Invoice Upload  
dialog box**

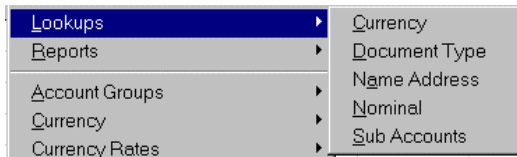
The values in the dialog box are defaults, and are only used if any required details are missing from the worksheet.

Number refers to the document number. It is automatically entered if autosequence rules are used in CODA, otherwise, it must be entered manually.

If LINE\_NUMBER was not selected in the Invoice Setup, then line numbers are automatically formatted such that if there is a change in the value of a DOCUMENT\_TYPE or DOCUMENT\_NUMBER column in the worksheet, then the LINE\_NUMBER value will be reset to correspond to these changes.

If the document format in CODA is JN, then the first line of the worksheet data should be left blank.


## Lookups

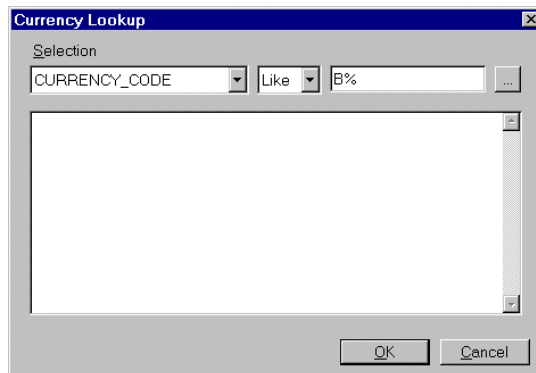


Lookups allows you to see all the values for a given category. This lets you ascertain, for example, whether a given value you want to enter is valid.

There are five lookup options: Currency, Document Type, Name Address, Nominal and Sub Accounts. The operation of these is identical, and so only one example, using Currency, is shown.

In the Selection dropdown enter the column on which you want to search. From the central dropdown, select the comparison operator. In the rightmost entry field type the characters you want to use in the search. You may use the percent character (%) as a wildcard to represent zero or more characters, and the underscore character (\_) to represent one character.

Press the  button (lookup button) button to display the required values.



**Figure 55.**

**Company Lookup  
dialog box**

Press **OK** (or **Cancel**) to exit.



## Reports



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

### Setup New Report

	A	B	C	D	E	F	G	H
1	Report Name	Author	Company					
2	simple	Mike	Easygo					
3								
4	Worksheet	Row	Column	Headings	Format	SQL	Last Rows	Last Time
5	Sheet1		1	1	YES	YES	SELECT * FROM EASYGO_LEDGER	
6								

**Figure 56. Setup New Report example**

1. Enter values for Report Name, Author and Company 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 Setup Report resides. 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, in which case 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 there is no formatting.
6. SQL. This is the SQL that is sent to the server. Multiple rows of SQL are allowed.
7. Save the worksheet if you will want to use the report again. 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 finishing time of the report.

## Run Report

1. Select the worksheet that contains the report you wish to run.
2. Run the report (**Coda, Reports, Run Report**). Example output:

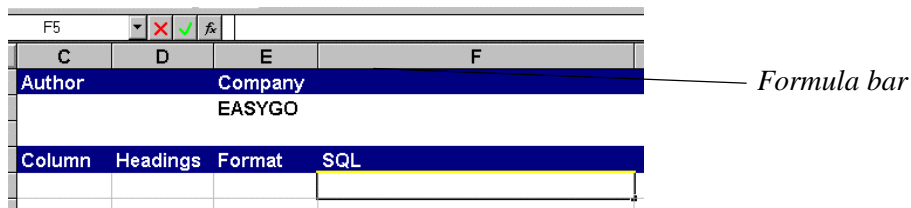
	A	B	C	D	E	F	G
1	LEDGER	NOMINAL	DESCRIPTION	CREATED_USER	CREATED_DATE	MODIFIED_USER	MODIFIED_DATE
2	E0001		VAT Ledger	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
3	E0002		Fixtures & Fittings	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
4	E0003		Holidays	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
5	E0004		Brochures	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
6	E0005		Destination	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
7	P0001	5100	Purchase Ledger	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997
8	S0001	2100	Sales Ledger	CAROLYN	4-Mar-1997	TRAINER	5-Mar-1997

## Advanced Techniques

To make the report easily transferable for any company, you can use a cell reference to the name of the company. If you change the company name, then all the references would change, so you would only need to make one global change to the worksheet. This is useful if there is much SQL.

A simple example follows (for clarity, only the relevant information is shown on the screen shots). Say we want this SQL query: `SELECT * FROM <company>_LEDGER`, where `<company>` is the name of the company that is displayed in the Company cell. When the company name is changed, the name will change in the SQL.

1. Enter the name of the company in the Company cell.
2. Select the cell in which you want to write the SQL.
3. Move the cursor to the Formula bar.

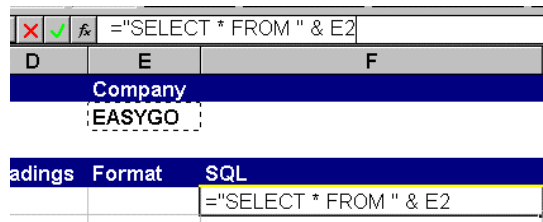


4. Press the equals (=) key (this indicates that a formula is to follow).

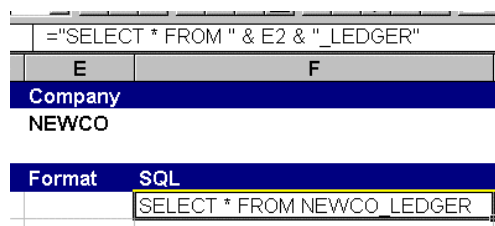
5. Type: "SELECT \* FROM "

Ensure you type the quote marks, and leave a space before the closing quote. Notice how the text also appears in the SQL cell (see next screen shot).

6. Press the space bar.
7. Press the ampersand (&) key.
8. Press the space bar.
9. Select the cell (E2) containing the company name.
10. Notice how the value E2 is added to the formula (you could have typed it directly).



11. Press the space bar, the ampersand (&) key, and the space bar.
12. Type the remainder of the SQL: "\_LEDGER"
13. Press the Enter key. The SQL is ready to run.
14. Notice how the formula contains a reference to cell E2 which contains the company name. If you change the name, then this change will automatically be reflected in the SQL.





# Troubleshooting

This chapter deals with various troubleshooting issues related to the installation and use of Easysoft ODBC for CODA. 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 "Edit the System Startup File", page 31).

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 tape

Back up the catalog and the data. You can back up to TK50 or DAT and send it in the post. For example,

```
$ BACKUP/REWIND/VERIFY EASYSOFT_SQL_CATALOG:*.*,DATA:*. * -
MKA700:EASYSOFT/LOG
```

- 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 then use one of the following methods:

Internet Mail:	support@easysoft.com
Internet FTP:	ftp.easysoft.com

- 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 25 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][ESODBC32]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] [ESODBC32]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 58.

- 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 19).

- 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 17); 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.

## ***CODA-specific Issues***

- Can I set up a CODA company using Easysoft ODBC for CODA?

No, you can't do this. Easysoft is designed to work on existing CODA companies.

- Chapter six describes indexes that are available on CODA tables. I can't see them when I access CODA tables, so where are they and how do I use them?

These are available for use by the application; whether or not you can see them, and exactly how they appear depend upon the application.

- Why can't I update/delete/insert to CODA data?

It is possible that you do not have the required privileges. See your CODA Administrator.

- Using CODA I created a new user. How do I insert this information into the catalog?

Run the Codacat routine.

- How do I know whether the Codaxref routine has been run for a particular company?

Codaxref generates a file called EASYF001.DAT. On the server platform go to the IAS\$GROUP directory. For each of the companies in this directory, check for the existence of EASYF001.DAT; if it exists, then Codaxref has been run (but note that if additional nominals, subaccounts or level3s have been added after Codaxref was run, then it should be run again).

- My system has more than one CODA version. How do I know which version I am using?

The Codacat routine ("Adding CODA Companies", page 37) shows the CODA version, so run this, and then quit without making changes.

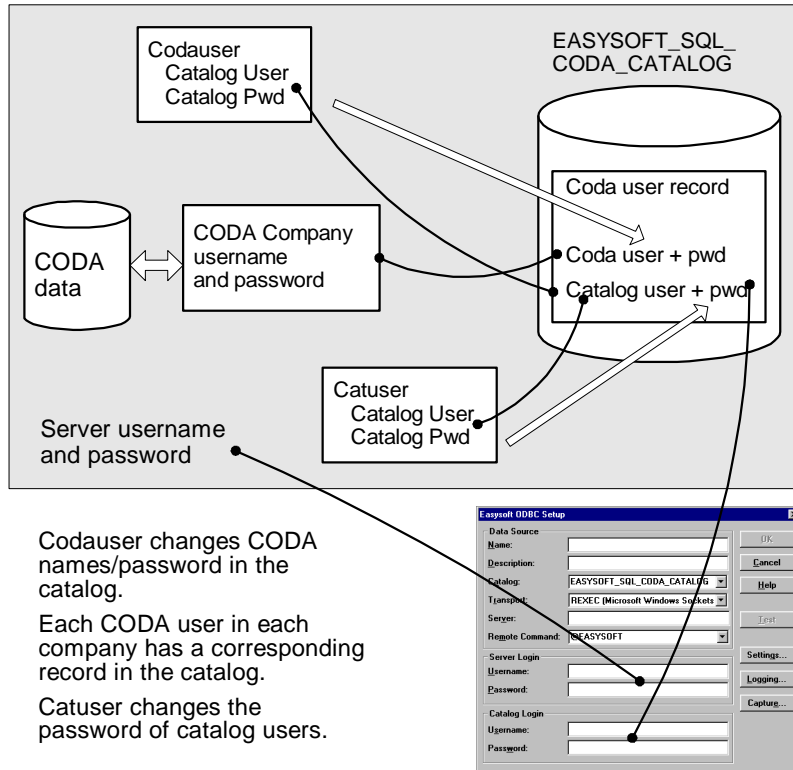
# Appendices



## APPENDIX A

# Names and Passwords

Figure 57 shows the password relationships between components in the Easysoft system.



**Figure 57. Password relationships**

The Server Login Username and Password in the Easysoft ODBC Setup dialog box are those needed to log on to the server. Each Catalog Login Username and its associated Password corresponds to a CODA user and associated password. See “Adding CODA Companies”, page 37 for more information.

**Note:** When a 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.

Table 70 lists the restrictions associated with Easysoft usernames, passwords and other names used in the Easysoft system.

<b>Table 70. Naming restrictions in the Easysoft system</b>			
<b>Name</b>	<b>Min. chars</b>	<b>Max. chars</b>	<b>Allowed characters and/or restrictions</b>
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
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 CODA.

## ***ODBC Files***

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 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.INI

This file controls Easysoft ODBC. An example is shown below, and following this, settings and options pertaining to Easysoft ODBC for CODA are described. If you have an Easysoft product which uses the Easysoft Administrator (Easysoft ODBC for CODA does not), then this file already exists in the Windows directory. If it does not exist, and if you want to use any of the options described here, then create the file using any text editor of your choice.

```
[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 164. The default is 0 (that is, the line can be omitted).

## **CODAXREF.COM**

CODAXREF.COM is a DCL command procedure which controls the operation of the batch job which runs the Codaxref routine on the server. It is located in the EASYSOFT\_SQL\_TEMP directory. The default options are shown below.

```

$
SUBMIT/AFTER=TOMORROW/KEEP/NOPRINT/LOG=EASYSOFT_SQL_TEMP:CODAXREF.LOG
/NAME=CODAXREF EASYSOFT_SQL_TEMP:CODAXREF.COM
$ SHOW TIME
$ DEFINE IAS$GROUP <device><directory>
$ CODAXREF:=$EASYSOFT_SQL_SYSTEM:CODAXREF.EXE
$ CODAXREF EASYSOFT_SQL_CODA_CATALOG ADMIN ADMIN *
$ SHOW TIME
$ CURRENT_DIRECTORY = F$ENVIRONMENT("DEFAULT")
$ SET DEF IAS$GROUP
$ PURGE [...]EASY*.DAT
$ SET PROTECTION=(WORLD:R) [...]EASY*.DAT
$ SET DEF 'CURRENT_DIRECTORY'
$ PURGE/KEEP=5 EASYSOFT_SQL_TEMP:CODAXREF.LOG$

```

You can change any of the lines to suit your particular requirements.

The Codaxref routine is run by the following line in the command procedure:

```
$ CODAXREF EASYSOFT_SQL_CODA_CATALOG ADMIN ADMIN *
```

This line accepts three arguments:

1. ADMIN (this is the catalog administrator's name and cannot be changed)
2. the catalog administrator's password (by default, this is also ADMIN. Ideally, you should have changed it).
3. the CODA companies for which Codaxref should operate. You can either run Codaxref on a single named company or on all the companies known to the Easysoft system (that is, all the companies for which details are recorded in the Easysoft Catalog). The asterisk character (\*) represents all the companies known to the catalog.

A special file called EASYF001.DAT which knows of the combinations of Nominal/Subaccount/Level3 that have budget data associated with them is generated by running the Codaxref routine. This file is automatically purged by the line:

```
$ PURGE [...]EASY*.DAT
```

## APPENDIX C

# Glossary

**\$** DCL prompt.

**!** DCL comment separator.

**account** (accounting term) Details of transactions of a given type (that is, items in a journal). Each account has a name (Account title) and a number (account number).

**Alpha** A type of computer produced by Digital Equipment Corporation (DEC) which can use the VMS and OpenVMS operating systems.

**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.

**BACS** Banks Automated Clearing System.

**balance sheet** Reports the financial condition of a business at a given point in time.

**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.

**book of original entry** Another name for a journal.

**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.

**chart of accounts** A list of account titles and account numbers of the accounts in a general ledger, in other words, a description of the structure of a general ledger.

**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.

- 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.
- contiguous** A technical term meaning adjacent, next to or continuous.
- control account** (accounting term) An account which is used to check the accuracy of a ledger. It summarises the information in the accounts contained in a given ledger.
- 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.
- entry** (accounting term) An item recorded in a journal.
- field** A segment of a data record.
- 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.

**financial statement** (accounting term) A summary of the financial information contained in the accounts.

**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.

**general ledger** (accounting term) A ledger that contains all accounts needed to prepare financial statements. Contains all accounts except those for customers and suppliers. Also known as a nominal ledger.

**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

**index** (in DEC terms) The structure that allows retrieval of records in an indexed file by key value. See key.

**interoperability** An application that is interoperable is one that can access many different databases.

**journal** (accounting term) A record of accounting information in chronological order.

**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.

**L1 nominal** (CODA) Account code of the nominal on the first line of the document that contains accounts.

**ledger** (accounting term) (also see general, purchase and sales ledgers) A group (also known as book) of related accounts.

**Ledger nominal (subaccount)** (CODA) The ledger specified in the nominal (subaccount) master record.

**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.

**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.

**Nominal** (CODA) The account number of the nominal as defined in the nominal master record.

**nominal ledger** (accounting) Another name for the general ledger.

**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.

**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.

**posting** (accounting term) The transfer of information from journal entries to ledger accounts.

**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.

**predicate** A predicate is a simple statement which can be true, false or sometimes unknown.

**primary key** (in RMS terms) The mandatory key within the records of an indexed file.

**privilege** A privilege is the right to perform a particular action (for example, INSERT) on an object (for example, a table).

**purchase ledger** (accounting) A ledger for suppliers' accounts.

**qualifier** A portion of a command string that modifies a command verb or a command parameter. It has the format: /qualifier[=option].

**record** A collection of related data items treated as a unit. A record contains one or more fields.

**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.

**sales ledger** (accounting) A ledger for customers' accounts.

**scalar value** A scalar value is in principle any value that can be assigned to a column.

**scale** Scale refers to the maximum number of digits to the right of the decimal point in numeric fields

**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.

**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).

**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 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  $10^{38}$ ).

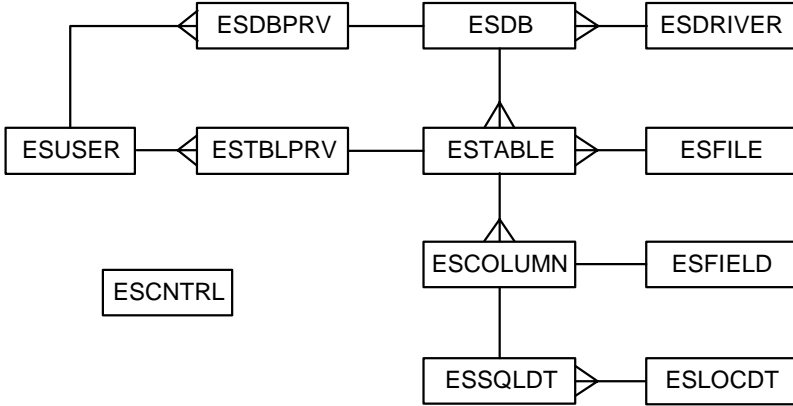
### Datetime Data Types

**DATE** Date data.

**TIME** Time data.

**TIMESTAMP** Date/time data.

# Easysoft Catalog



**Figure 58. Easysoft Catalog**

Figure 58 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

### ESCNTL

Filename ESCNTL.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

<u>Column</u>	<u>Data Type</u>	<u>Length</u>
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

<u>Column</u>	<u>Data Type</u>	<u>Length</u>
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

**ESLOC DT**

Filename ESLOC DT.DAT  
 Record Type FIXED  
 Record Size 712

<u>Column</u>	<u>Data Type</u>	<u>Length</u>
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

<u>Column</u>	<u>Data Type</u>	<u>Length</u>
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

<u>Column</u>	<u>Data Type</u>	<u>Length</u>
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

<u>Column</u>	<u>Data Type</u>	<u>Length</u>
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

<u>Column</u>	<u>Data Type</u>	<u>Length</u>
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.

# 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	COBOL	DEFERRED
ACTION	COLLATE	DELETE
ADA	COLLATION	DEPTH
ADD	COLLATION_CATALOG	DESC
AFTER	COLLATION_NAME	DESCRIBE
ALIAS	COLLATION_SCHEMA	DESCRIPTOR
ALL	COLUMN	DIAGNOSTICS
ALLOCATE	COLUMN_NAME	DICTIONARY
ALTER	COMMAND_FUNCTION	DISCONNECT
AND	COMMIT	DISTINCT
ANY	COMMITTED	DOMAIN
ARE	COMPLETION	DOUBLE
AS	CONDITION_NUMBER	DROP
ASC	CONNECT	DYNAMIC_FUNCTION
ASSERTION	CONNECTION	EACH
ASYNC	CONNECTION_NAME	ELSE
AT	CONSTRAINT	ELSEIF
AUTHORIZATION	CONSTRAINT_CATALOG	END
AVG	CONSTRAINT_NAME	END-EXEC
BEFORE	CONSTRAINT_SCHEMA	EQUALS
BEGIN	CONSTRAINTS	ESCAPE
BETWEEN	CONTINUE	EXCEPT
BIT	CONVERT	EXCEPTION
BIT_LENGTH	CORRESPONDING	EXEC
BOOLEAN	COUNT	EXECUTE
BOTH	CREATE	EXISTS
BREADTH	CROSS	EXTERNAL
BY	CURRENT	EXTRACT
C	CURRENT_DATE	FALSE
CALL	CURRENT_TIME	FETCH
CASCADE	CURRENT_TIMESTAMP	FIRST
CASCADED	CURRENT_USER	FLOAT
CASE	CURSOR	FOR
CAST	CURSOR_NAME	FOREIGN
CATALOG	CYCLE	FORTRAN
CATALOG_NAME	DATA	FOUND
CHAR	DATE	FROM
CHAR_LENGTH	DATETIME_INTERVAL_CODE	FULL
CHARACTER	DATETIME_INTERVAL_PRECISION	GENERAL
CHARACTER_LENGTH	DAY	GET
CHARACTER_SET_NAME	DEALLOCATE	GLOBAL
CHARACTER_SET_SCHEMA	DEC	GO
CHECK	DECIMAL	GOTO
CLASS_ORIGIN	DECLARE	GRANT
CLOSE	DEFAULT	GROUP
COALESCE	DEFERRABLE	HAVING



HOUR	NO	RESTRICT
IDENTITY	NONE	RETURN
IF	NOT	RETURNED_LENGTH
IGNORE	NOTRACEPOINT	RETURNED_OCTET_LENGTH
IMMEDIATE	NULL	RETURNED_SQLSTATE
IN	NULLABLE	RETURNS
INCLUDE	NULLIF	REVOKE
INDEX	NUMBER	RIGHT
INDICATOR	NUMERIC	ROLE
INITIALLY	OBJECT	ROLLBACK
INNER	OCTET_LENGTH	ROUTINE
INPUT	OF	ROW
INSENSITIVE	OFF	ROW_COUNT
INSERT	OID	ROWS
INT	OLD	SAVEPOINT
INTEGER	ON	SCALE
INTERSECT	ONLY	SCHEMA
INTERVAL	OPEN	SCHEMA_NAME
INTO	OPERATION	SCROLL
IS	OPERATORS	SEARCH
ISOLATION	OPTION	SECOND
JOIN	OR	SECTION
KEY	ORDER	SELECT
LANGUAGE	OTHERS	SENSITIVE
LAST	OUTER	SEQUENCE
LEADING	OUTPUT	SERIALIZABLE
LEAVE	OVERLAPS	SERVER_NAME
LEFT	PAD	SESSION
LENGTH	PARAMETERS	SESSION_USER
LESS	PARTIAL	SET
LEVEL	PASCAL	SIGNAL
LIKE	PENDANT	SIMILAR
LIMIT	PLI	SIZE
LOCAL	POSITION	SMALLINT
LOOP	PRECISION	SOME
LOWER	PREORDER	SPACE
MATCH	PREPARE	SQL
MAX	PRESERVE	SQLCA
MESSAGE_LENGTH	PRIMARY	SQLCODE
MESSAGE_OCTET_LENGTH	PRIOR	SQLERROR
MESSAGE_TEXT	PRIVATE	SQLEXCEPTION
MIN	PRIVILEGES	SQLSTATE
MINUTE	PROCEDURE	SQLWARNING
MODIFY	PROTECTED	STRUCTURE
MODULE	PUBLIC	SUBCLASS_ORIGIN
MONTH	READ	SUBSTRING
MORE	REAL	SUM
MUMPS	RECURSIVE	SYSTEM
NAME	REF	SYSTEM_USER
NAMES	REFERENCES	TABLE
NATIONAL	REFERENCING	TABLE_NAME
NATURAL	RELATIVE	TEMPORARY
NCHAR	REPEATABLE	TEST
NEW	REPLACE	THEN
NEXT	RESIGNAL	THERE

TIME	UNDER	VIEW
TIMESTAMP	UNION	VIRTUAL
TIMEZONE_HOUR	UNIQUE	VISIBLE
TIMEZONE_MINUTE	UNKNOWN	WAIT
TO	UNNAMED	WHEN
TRACEPOINT	UPDATE	WHENEVER
TRAILING	UPPER	WHERE
TRANSACTION	USAGE	WHILE
TRANSLATE	USER	WITH
TRANSLATION	USING	WITHOUT
TRIGGER	VALUE	WORK
TRIM	VALUES	WRITE
TRUE	VARCHAR	YEAR
TYPE	VARIABLE	ZONE
UNCOMMITTED	VARYING	

# Import Export Formats

The IMPORT CSV and EXPORT CSV commands in the Easysoft Host Administrator use the following CSV formats.

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>
```

**Example of Comma Separated Values**

```
"FILE" , "ACCOUNT_GROUP" , "INDEXED" , "FIXED" , 43
"FIELD" , "ACCOUNT_GROUP" , "ACCOUNT_GROUP" , "STRING" , 0 , 12 , 12 , 0 , 0 , " , "
"FIELD" , "ACCOUNT_GROUP" , "GROUP_TITLE" , "STRING" , 12 , 30 , 30 , 0 , 0 , " , "
"FIELD" , "ACCOUNT_GROUP" , "GROUP_TYPE" , "STRING" , 42 , 1 , 1 , 0 , 0 , " , "N"
"DB" , "CODADEM" , "DKA0:[ IAS.V670.GROUP.CODADEM]" , "CODA-
RMS" , "COMPANY=CODADEM;GROUP=DKA0:[ IAS.V670.GROUP];TEXT=DKA0:[ IAS.V670.TEXT];SYSTEM=DKA0:[ IAS.V670.SYSTEM];TIMEOUT=300;"
"TABLE" , "CODADEM" , "ACCOUNT_GROUP" , "TABLE" , "ACCOUNT_GROUP" , "1" , "
"COLUMN" , "CODADEM" , "ACCOUNT_GROUP" , "ACCOUNT_GROUP" , "ACCOUNT_GROUP" ,
"VARCHAR" , 12 , 1 , 1 , "
"COLUMN" , "CODADEM" , "ACCOUNT_GROUP" , "GROUP_TITLE" , "GROUP_TITLE" , "VAR
CHAR" , 30 , 1 , 1 , "
"COLUMN" , "CODADEM" , "ACCOUNT_GROUP" , "GROUP_TYPE" , "GROUP_TYPE" , "VARCH
AR" , 1 , 1 , 1 , "N"
"DBPRV" , "MIKE" , "CODADEM" , "MIKE" , " , " , 1 , 1 , 1 , 1 , "TABLE"
"TBLPRV" , "MIKE" , "CODADEM" , "ACCOUNT_GROUP" , 1 , 1 , 1 , 1
```

## 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.

<b>Function</b>	<b>Description</b>
ASCII( <i>str_exp</i> )	Returns the ASCII code value of the leftmost character of <i>str_exp</i> as an integer.
CHAR( <i>code</i> )	Returns the character that has the ASCII code value specified by <i>code</i> . The value of <i>code</i> should be between 0 and 255; otherwise, the return value is data source dependent.
CONCAT( <i>str_exp1</i> , <i>str_exp2</i> )	Returns a character string that is the result of concatenating <i>str_exp2</i> to <i>str_exp1</i> . The resulting string is DBMS dependent.
DIFFERENCE( <i>str_exp1</i> , <i>str_exp2</i> )	Returns an integer value that indicates the difference between the values returned by the SOUNDEX function for <i>str_exp1</i> and <i>str_exp2</i> .
INSERT( <i>str_exp1</i> , <i>start</i> , <i>length</i> , <i>str_exp2</i> )	Returns a character string where <i>length</i> characters have been deleted from <i>str_exp1</i> beginning at <i>start</i> and where <i>str_exp2</i> has been inserted into <i>str_exp1</i> , beginning at <i>start</i> .
LCASE( <i>str_exp</i> )	Converts all upper case characters in <i>str_exp</i> to lower case.
LEFT( <i>str_exp</i> , <i>count</i> )	Returns the leftmost <i>count</i> of characters of <i>str_exp</i> .
LENGTH( <i>str_exp</i> )	Returns the number of characters in <i>str_exp</i> , excluding trailing blanks and the string termination character.
LOCATE( <i>str_exp1</i> , <i>str_exp2</i> [, <i>start</i> ])	Returns the starting position of the first occurrence of <i>str_exp1</i> within <i>str_exp2</i> . The search for the first occurrence of <i>str_exp1</i> begins with the first character position in <i>str_exp2</i> unless the optional argument, <i>start</i> , is specified. If <i>start</i> is specified, the search begins with the character position indicated by the value of <i>start</i> . The first character position in <i>str_exp2</i> is indicated by the value 1. If <i>str_exp1</i> is not found within <i>str_exp2</i> , the value 0 is returned.
LTRIM( <i>str_exp</i> )	Returns the characters of <i>str_exp</i> , with leading blanks removed.
REPEAT( <i>str_exp</i> , <i>count</i> )	Returns a character string composed of <i>str_exp</i> repeated <i>count</i> times.
REPLACE( <i>str_exp1</i> , <i>str_exp2</i> , <i>str_exp3</i> )	Replaces all occurrences of <i>str_exp2</i> in <i>str_exp1</i> with <i>str_exp3</i> .
RIGHT( <i>str_exp</i> , <i>count</i> )	Returns the rightmost <i>count</i> of characters of <i>str_exp</i> .
RTRIM( <i>str_exp</i> )	Returns the characters of <i>str_exp</i> with trailing blanks removed.
SOUNDEX( <i>str_exp</i> )	Returns a data source dependent character string representing the sound of the words in <i>str_exp</i> .
SPACE( <i>count</i> )	Returns a character string consisting of <i>count</i> spaces.

SUBSTRING( <i>str_exp</i> , <i>start</i> , <i>length</i> )	Returns a character string that is derived from <i>str_exp</i> beginning at the character position specified by <i>start</i> for <i>length</i> characters.
UCASE( <i>str_exp</i> )	Converts all lower case characters in <i>str_exp</i> 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.

<b>Function</b>	<b>Description</b>
ABS( <i>numeric_exp</i> )	Returns the absolute value of <i>numeric_exp</i> .
ACOS( <i>float_exp</i> )	Returns the arccosine of <i>float_exp</i> as an angle, expressed in radians.
ASIN( <i>float_exp</i> )	Returns the arcsine of <i>float_exp</i> as an angle, expressed in radians.
ATAN( <i>float_exp</i> )	Returns the arctangent of <i>float_exp</i> as an angle, expressed in radians.
ATAN2( <i>float_exp1</i> , <i>float_exp2</i> )	Returns the arctangent of the x and y co-ordinates, specified by <i>float_exp1</i> and <i>float_exp2</i> , respectively, as an angle, expressed in radians.
CEILING( <i>numeric_exp</i> )	Returns the smallest integer greater than or equal to <i>numeric_exp</i> .
COS( <i>float_exp</i> )	Returns the cosine of <i>float_exp</i> , where <i>float_exp</i> is an angle expressed in radians.
COT( <i>float_exp</i> )	Returns the cotangent of <i>float_exp</i> , where <i>float_exp</i> is an angle expressed in radians.

DEGREES( <i>numeric_exp</i> )	Returns the number of degrees converted from <i>numeric_exp</i> radians.
EXP( <i>float_exp</i> )	Returns the exponential value of <i>float_exp</i> .
FLOOR( <i>numeric_exp</i> )	Returns the largest integer less than or equal to <i>numeric_exp</i> .
LOG( <i>float_exp</i> )	Returns the natural logarithm of <i>float_exp</i> .
LOG10( <i>float_exp</i> )	Returns the base 10 logarithm of <i>float_exp</i> .
MOD( <i>int_exp1</i> , <i>int_exp2</i> )	Returns the remainder (modulus) of <i>int_exp1</i> divided by <i>int_exp2</i> .
PI()	Returns the constant value of pi as a floating point value.
POWER( <i>numeric_exp</i> , <i>int_exp</i> )	Returns the value of <i>numeric_exp</i> to the power of <i>int_exp</i> .
RADIANS( <i>numeric_exp</i> )	Returns the number of radians converted from <i>numeric_exp</i> degrees.
RAND([ <i>int_exp</i> ])	Returns a random floating point value using <i>int_exp</i> as the optional seed value.
ROUND( <i>numeric_exp</i> , <i>int_exp</i> )	Returns <i>numeric_exp</i> rounded to <i>int_exp</i> places right of the decimal point. If <i>int_exp</i> is negative, <i>numeric_exp</i> is rounded to $ int\_exp $ places to the left of the decimal point.
SIGN( <i>numeric_exp</i> )	Returns an indicator or the sign of <i>numeric_exp</i> . If <i>numeric_exp</i> is less than zero, -1 is returned. If <i>numeric_exp</i> equals zero, 0 is returned. If <i>numeric_exp</i> is greater than zero, 1 is returned.
SIN( <i>float_exp</i> )	Returns the sine of <i>float_exp</i> , where <i>float_exp</i> is an angle expressed in radians.
SQRT( <i>float_exp</i> )	Returns the square root of <i>float_exp</i> .
TAN( <i>float_exp</i> )	Returns the tangent of <i>float_exp</i> , where <i>float_exp</i> is an angle expressed in radians.
TRUNCATE( <i>numeric_exp</i> , <i>int_exp</i> )	Returns <i>numeric_exp</i> truncated to <i>int_exp</i> places right of the decimal point. If <i>int_exp</i> is negative, <i>numeric_exp</i> 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.

<b>Function</b>	<b>Description</b>
CURDATE( )	Returns the current date as a date value.
CURTIME( )	Returns the current local time as a time value.
DAYNAME( <i>date_exp</i> )	Returns a character string containing the data source-specific name of the day for the day portion of <i>date_exp</i> . Examples: Sunday for a data source that uses English, or Sonntag for a data source that uses German.
DAYOFMONTH( <i>date_exp</i> )	Returns the day of the month in <i>date_exp</i> as an integer value in the range 1 to 31.
DAYOFWEEK( <i>date_exp</i> )	Returns the day to the week in <i>date_exp</i> as an integer value in the range 1 to 7, where 1 represents Sunday.
DAYOFYEAR( <i>date_exp</i> )	Returns the day of the year in <i>date_exp</i> as an integer value in the range 1 to 366.
HOUR( <i>time_exp</i> )	Returns the hour in <i>time_exp</i> as an integer value in the range 0 to 23.
MINUTE( <i>time_exp</i> )	Returns the minute in <i>time_exp</i> as an integer value in the range 0 to 59.
MONTH( <i>date_exp</i> )	Returns the month in <i>date_exp</i> as an integer value in the range 1 to 12.

MONTHNAME( <i>date_exp</i> )	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 <i>date_exp</i> .
NOW()	Returns the current date and time as a timestamp value.
QUARTER( <i>date_exp</i> )	Returns the quarter in <i>date_exp</i> as an integer value in the range 1 to 4, where 1 represents January 1 to March 31.
SECOND( <i>time_exp</i> )	Returns the second in <i>time_exp</i> as an integer value in the range 0 to 59.
WEEK( <i>date_exp</i> )	Returns the week of the year in <i>date_exp</i> as an integer value in the range 1 to 53.
YEAR( <i>date_exp</i> )	Returns the year in <i>date_exp</i> 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 ( <i>hdbc</i> ). (The name of the database is also available by calling <b>SQLGetConnectOption</b> with the CURRENT_QUALIFIER connection option.)
IFNULL( <i>exp</i> , <i>value</i> )	If <i>exp</i> is null, <i>value</i> is returned. If <i>exp</i> is not null, <i>exp</i> is returned. The possible data type(s) of <i>value</i> must be compatible with the data type of <i>exp</i> .
USER()	Returns the user's authorisation name. (The user's authorisation name is also available via <b>SQLGetInfo</b> 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.

<b>Function</b>	<b>Description</b>
CONVERT( <i>value_exp</i> , <i>data_type</i> )	Returns the value specified by <i>value_exp</i> ; the value is converted to the <i>data_type</i> specified.

# 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

## ***CODA Errors***

<string> cannot contain lowercase characters  
 CODA IAS login invalid  
 Comment record already exists  
 Date record already exists  
 File is not currently open  
 Function <string> is not available in CODA version <numeric value>  
 Function <string> is not supported in the CODA driver  
 Invalid account group type <character>  
 Invalid CODA version <numeric value>  
 Invalid comment number. The correct value for the selected <string> is <numeric value>  
 Invalid currency location flag <character>  
 Invalid list number. The correct value for the selected <string> is <numeric value>  
 Invalid start mode <numeric value>  
 Key descriptor contains an invalid key number <numeric value>  
 Maximum Number of Files Open  
 Minimum value for the comment number is 1  
 Minimum value for the list number is 1  
 Owner record already exists  
 Posted Documents Cannot Be Amended.  
 Table <string> is not available in CODA version <numeric value>  
 The current table is read only  
 This table is not available to user <string>  
 Unable to allocate stack  
 Unable to delete stack  
 Unable to get error information from IASLINKSHR  
 Unable to get top of stack  
 Unable to initialise RMS driver  
 Unable to logoff CODA IAS  
 Unable to pop stack  
 Unable to push stack  
 Unable to set logical <string>  
 Unable to terminate RMS driver  
 Unsupported CODA table number <numeric value> in <string> function

## **Codacat Errors**

\*\*\* Error: Unidentified field datatype: <string> \*\*\* in field definition for <string>  
 <string> contains a reserved name <string>  
 <string> contains an incorrect value <value>  
 <string> is invalid (Allowed values - <string>, <string>, <string>)  
 <string> is out of range (Allowed range - <number> <=> <number>)  
 Admin password is invalid  
 Admin Password not verified  
 Catalog directory <string> is invalid  
 Catalog not found in directory <string>  
 Company <string> is the name of an existing CODA company  
 Company ES is reserved for Easysoft and cannot be used within the Easysoft catalogs  
 Could not locate CODA-RMS driver entry in Catalog



---

default directory pathname larger than <number> characters '<string>'  
Error - Unable to open <string>  
ERROR : - CODACAT failed.  
Error building user <string>'s privileges  
Error Checking Logicals  
Error deleting Catalog definitions for <string>  
Error initializing timer  
Error padding USER record  
Error refreshing users definitions in catalog  
Error retrieving CODA user information  
Error retrieving CODA version information  
Error showing timer  
Error writing DB files to catalog  
Error writing to ESDB.DAT  
Error writing to file ESCOLUMN.DAT  
Failed to create new catalog in directory <string>  
Failed to translate logical 'IAS\$TEXT'  
Field <string> is duplicated in the key definition  
File Name (<string>)/Key Number (<number>) or Key Name (<string>) already exists  
Invalid entity <string>  
Invalid entry '<string>'  
Length of <string> exceeds allowable limit of <number> characters  
Maximum number of fields allowed is 8  
Name contains invalid character  
Name contains non-alphabetic first character  
Name must not be blank  
No Companies present  
Outputting <object> definitions  
Password Cannot Be blank  
Please select an alternative name for ES to be used within the catalogs (Max 9 characters)  
Remove connections to database. Catalog still in use.  
The company name ES cannot be used within the Easysoft catalogs  
This catalog already contains definitions for CODA version <version>  
Unable to detect catalog CODA version  
Unable to find company <string> in CSV file  
Unable to lock new catalog in directory <string>  
Unable to read <string>  
Unable to remove exclusive lock to catalog in directory <string>  
Unable to update catalog with CODA alias  
Unable to update catalog with CODA version  
Unrecognized keyword - check validity and spelling.  
Warning : too many budget periods, <number> (limit = <number>), contact Easysoft  
Warning : Unable to retrieve user <string>'s privileges continuing  
Warning : Value parameter ignored <string>

## 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 185).
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
```

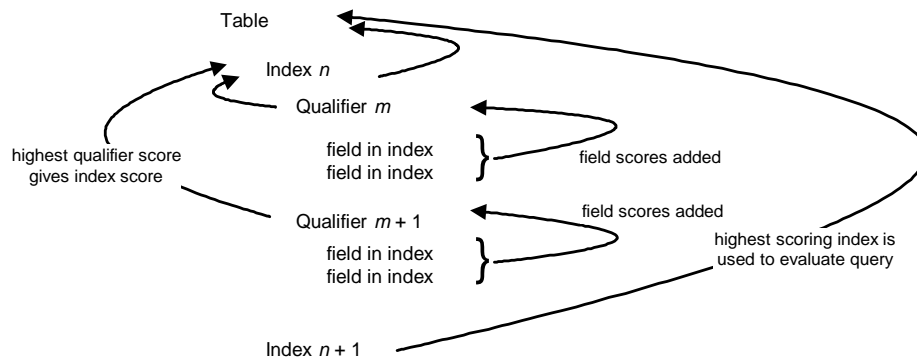
In this case <table2> will be accessed by the query first, and then <table1> will be accessed.

11. 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 59. 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:

$$= \frac{1}{n} \text{Field score} \times \text{Field length}(k) \times \text{Field factor}(k) \times \exp((\text{key offset}) / (\text{index factor}))$$

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 Example

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 use the BUDGET table (see Table 12 and Table 13).

**Note:** index names are 1-based. However, the QEP log is 0-based. Thus, for example, the index named BUDGET\_002 is shown as Index 1 in the QEP. Similarly, the first qualifier in the QEP is shown as qualifier 0.

### Query

SQL condition: WHERE SUBACCOUNT = 'B0005'

Scoring for Index 2 (BUDGET\_003), qualifier 0 (SUBACCOUNT = 15)

Field score	4	due to equal (=) operator
Index field length	12	SUBACCOUNT field is 12 bytes
Field factor	1	for all operators (except LIKE) = 1
Index field offset	0	offset of SUBACCOUNT field in index BUDGET_003
Key offset	255	maximum key length (255) - index field offset (0)
Index factor	25.5	maximum key length / 10

score =  $\boxed{\phantom{000000}}$  Field score  $\times$  Field length(k)  $\times$  Field factor(k)  $\times$  exp ((key offset) / (index factor))

$$= 4 \times 12 \times 1 \times \exp((255-0) / 25.5) = 1057270$$

There is only one field to evaluate in the index.

There is only one qualifier in this query. It scores 660407 on index BUDGET\_001 (index 0 in the QEP) and zero on the other indexes, thus index LASER\_003 (index number 2) will be used to evaluate the query.

## SQL and QEP log for query

```

=====
EXECUTE STATEMENT

SELECT EASYSOFT_BUDGET.BUDGET_CODE, EASYSOFT_BUDGET.BUDGET_YEAR, EASYSOFT_BUDGET.LEVEL3, EASYSOFT_BUDGET.NOMINAL, EASYSOFT_BUDGET.SUBACCOUNT, EASYSOFT_BUDGET.TOTAL_VALUE, EASYSOFT_BUDGET.BUDGET_CODE, EASYSOFT_BUDGET.BUDGET_YEAR, EASYSOFT_BUDGET.LEVEL3, EASYSOFT_BUDGET.SUBACCOUNT, EASYSOFT_BUDGET.NOMINAL
FROM EASYSOFT_BUDGET EASYSOFT_BUDGET
WHERE
  (EASYSOFT_BUDGET.SUBACCOUNT='B0005')
=====
QUERY EXECUTION PLAN

Index Scores

Table : EASYSOFT_BUDGET          (EASYSOFT_BUDGET ) Qualifier : 0   Index : 0
Score : 660407
Column 1 - Score                0 - NOMINAL
Column 2 - Score                660407 - SUBACCOUNT
Column 3 - Score                0 - LEVEL3
Column 4 - Score                0 - BUDGET_YEAR
Column 5 - Score                0 - BUDGET_CODE
Key Usage: 30%

Fast mode available: Yes

Table : EASYSOFT_BUDGET          (EASYSOFT_BUDGET ) Qualifier : 0   Index : 1
Score : 0
Column 1 - Score                0 - NOMINAL
Key Usage: 0%

Fast mode available: No

Table : EASYSOFT_BUDGET          (EASYSOFT_BUDGET ) Qualifier : 0   Index : 2
Score : 1057270
Column 1 - Score                1057270 - SUBACCOUNT
Key Usage: 100%

Fast mode available: No

Table : EASYSOFT_BUDGET          (EASYSOFT_BUDGET ) Qualifier : 0   Index : 3
Score : 0
Column 1 - Score                0 - LEVEL3
Key Usage: 0%

Fast mode available: No

Qualifier 0

Table      : EASYSOFT_BUDGET (EASYSOFT_BUDGET)
Index      : 2              Start Mode : 7 (Greater than or equal)
Start with :
  COLUMN(EASYSOFT_BUDGET.SUBACCOUNT) = VALUE(STRING("B0005"))
Finish with :
  COLUMN(EASYSOFT_BUDGET.SUBACCOUNT) = VALUE(STRING("B0005"))
Qualifiers:
  COLUMN(EASYSOFT_BUDGET.SUBACCOUNT) = VALUE(STRING("B0005"))

Summary: Tables [1]   Index Sort [No]   Quals [1]
$

```

*Index number 2 is used for this query. It has the highest score.*

---

# Bibliography

This bibliography lists (by title) a few sources of information relevant to information in this manual.

*Book-keeping & Accounts*, F. Wood. Published by Pitman, 1992. ISBN 0-273-03770-6.

*CODA Documentation*, Coda Limited.

*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.

*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.

*Using ODBC 2*, R. Gryphon, L. Charpentier, J. Oelschlaeger, A. Shoemaker, J. Cross, A. W. Lilley. QUE Corporation, 1995. ISBN 0-7897-0015-8.



---

# 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.

## A

access rights. *See under* CODA  
account  
    defined, 9  
account groups  
    restrictions on inserting data, 89, 93, 94  
Account groups  
    creating with Easysoft ODBC, 107  
Account Groups, 13, 14, 71  
ACCOUNT table, 41  
    generating, 71  
    set up, 29  
ACCOUNT\_GROUP table, 108  
accounting structure, 10  
Actual Budget Variance report. *See under*  
    CODA macro  
Add button  
    greyed out, 141  
ADD LICENCE command, 33, 44  
Add-ins dialog box, 114  
adjustment period, 72  
ADMIN user, 62  
ADMIN> prompt. *See under* Host  
    Administrator  
Age Analysis dialog box. *See under* CODA  
    macro  
Age Analysis report. *See under* CODA macro  
Analysis Titles, 14  
API conformance. *See* ODBC conformance  
application  
    using ODBC, 2  
approximate numeric data types. *See* data  
    types, SQL  
architecture  
    Easysoft. *See* Easysoft architecture  
    ODBC. *See under* ODBC  
Attach Tables dialog box, 101

Available CODA Companies dialog box. *See under* CODA macro

## B

Balance Sheet report. *See under* CODA  
    macro  
Balance Sheet Selection Criteria dialog box.  
    *See under* CODA macro  
BALANCE table, 41  
batch job for Codaxref. *See under* Codaxref  
    routine  
book of original entry. *See* journal  
Browse dialog box, 115  
Budget Download dialog box. *See under*  
    CODA macro  
budget periods  
    number supported, 73  
BUDGET table, 41, 112  
    set up, 29  
Budget Upload dialog box. *See under* CODA  
    macro  
Budget Variance Selection dialog box. *See under*  
    CODA macro  
buttons  
    togglng between, xix

## C

catalog, 6  
catalog driver, 6  
Catalog Login  
    changing password, 35  
    password, 39  
    username, 39  
Catuser routine, 27, 35  
cell reference, 134  
CHANNELCNT, 30  
character string data types. *See* data types,  
    SQL

- chart of accounts
  - CODA, 12
  - CODA restrictions, 13
  - defined, 10
  - example, 103
- checking installation. *See under* Easysoft Server Component
- client users
  - checking status of, 44
  - number of, 44
  - removing. *See under* users
- clients. *See* client users
- CODA
  - access rights, 13, 40
  - company. *See* Company
  - determining version, 144
  - errors, 180
  - identification number, referencing, 71
  - menu name
    - correspondence to column name, 70
  - security level. *See* CODA access rights
  - supported versions, 18
- CODA data
  - allowed operations, 6
- CODA directory, 26
- CODA driver, 6
- CODA macro
  - Actual Budget Variance report, 124
  - Age Analysis dialog box, 125
  - Age Analysis report, 125
  - Available CODA Companies dialog box, 117
  - Balance Sheet report, 126
  - Balance Sheet Selection Criteria dialog box, 126
  - Budget Download dialog box, 123
  - Budget Upload dialog box, 123
  - Budget Variance Selection dialog box, 124
  - Budgets, 122
  - changing period range, 122
    - prevent deletion of worksheet data, 122
  - contact information, 113
  - Currency Lookup dialog box, 132
  - customised reports, 133
    - creating, 133
    - generalising the report, 134
    - running, 134
  - debug options, 118
  - define Budget upload format, 122
  - Details Download dialog box, 130
  - document details download, 129
  - download Budgets, 123
  - Download dialog box, 119
  - failed update colour, 120
  - Fixed reports, 124
  - font colour after upload, 120
  - IASLINK errors, 121
  - initialisation, 116
  - Input and Intry, 129
  - installation, 114
  - Invoice Setup dialog box, 130
  - keeping connection open, 117
  - location of add-in software, 114
  - lookups, 132
  - Options dialog box, 116
  - overview, 113
  - post documents, 130
  - prevent display of error messages, 118
  - Profit & Loss Selection dialog box, 127
  - Profit and Loss report, 127
  - restricting number of rows, 117
  - selecting CODA company, 116
  - selecting year range, 117
  - selection criteria, 119
  - setting timeout value, 117
  - Setup Budget Format dialog box, 122
  - status message, 117
  - successful insert colour, 121
  - successful update colour, 121
  - Table Download dialog box, 118
  - Trial Balance dialog box, 128
  - Trial Balance report, 128
  - update sequence rules, 121
  - upload, 120
  - upload Budgets, 123
  - Upload dialog box, 131
  - upload invoice. *See* CODA macro, post documents
    - preparation, 130
  - upload status, 120
  - version information, 113
- Coda menu option, 113
- CODA setup routine, 27
- CODA tables
  - defined, 69
- CODA user, 13
  - adding to catalog, 38, 144
  - deleting from catalog, 38
- CODA utilities, 6
- Codacat routine, 27, 28, 37–39, 102

- Codauser routine, 27
    - function of, 39
    - location of, 39
  - Codaxref routine, 41
    - checking usage of, 144
    - modifying batch job options, 151
    - set up as batch job, 30
    - when to re-run, 42
  - CODAXREF.COM, 151
  - company
    - adding to catalog, 37
    - creating, 102, 143
    - defined, 11
    - structure, 12
  - Company Number, 32
    - entering, 33
  - conformance. *See* ODBC conformance
  - connect timeout. *See* timeouts
  - connection to server
    - rejection of, 44
  - control account, 10
  - Control Account
    - creating with Easysoft ODBC, 104
  - convert data types. *See under* scalar functions
  - CREATE CATALOG command, 45
  - Create New Data Source dialog box, 59
  - Currency Lookup dialog box. *See under* CODA macro
  - Currency Masters, 14
  - Customised reports. *See under* CODA macro
- D**
- data
    - sending to Easysoft, 137
    - update problems with CODA, 144
  - data sources
    - configuring, 8, 66
    - defined, 3
    - defining. *See* data sources, configuring
    - function of, 7
    - managing, 57–68, 57–68
    - method of definition, 7
    - naming restrictions, 148
    - removing, 66
    - settings, 63
    - unable to add, 142
    - validation of, 62
  - data type conversion functions. *See* scalar functions
  - data types
    - SQL, 158
  - Database dialog box, 101
  - databases
    - defined, 7
  - date format
    - default, 70
  - date functions. *See* scalar functions
  - datetime data types. *See* data types, SQL
  - DCL continuation symbol, 31
  - DEC C Runtime components kit, 19
  - DECCRTL, 19
  - DECCRTL.ZIP, 19
  - DECNET, 25
    - errors, 179
    - node address not found, 140
    - not correctly installed, 139
    - ODBC-call failed, 140
    - test for VAX/Alpha process, 141
  - detail lines, 14
  - Details Download dialog box. *See under* CODA macro
  - DETAILS table, 110
    - difference between INTRAY, 79
  - disable Winsock. *See under* Winsock
  - disk space requirements. *See under specific components*
  - DOC directory, 26
  - document
    - typographical conventions, xviii
  - document headers, 14
  - document masters
    - creation restriction, 102
  - Document masters, 14
  - Download dialog box. *See under* CODA macro
  - Driver Manager, 2
  - drivers. *See* ODBC driver
  - drop-down combo-box, defined, xix
  - drop-down list box, defined, xix
  - dynamic tracing. *See* function calls, tracing
- E**
- EASYDEMO data source
    - ADMIN user's password, 62
    - creating, 62
    - purpose of, 57
  - EASYF001.DAT, 144, 151
  - Easysoft architecture, 5

- Easysoft catalog
    - maintaining, 34
  - Easysoft Catalog
    - creating, 45
    - entity diagram, 159
    - exporting, 45
    - file and table definitions, 159
    - importing data to server, 46
    - number of, 8
    - purpose of, 8
    - specifying location of, 61
    - upgrading, 18, 50
  - Easysoft Catalog (CODA)
    - directory, 26
  - Easysoft Client Component, 5
  - Easysoft Excel Macro for CODA. *See* CODA macro
  - Easysoft Host Administrator. *See* Host Administrator
  - Easysoft ODBC
    - cancelling installation, 54
    - files used, 149
    - installation, 53
      - overall process, 1
    - memory requirement, 51
    - storage requirement, 51
    - system requirement, 51
    - upgrading, 51
  - Easysoft ODBC Capture dialog box, 66
  - Easysoft ODBC driver
    - installation, 51–55
    - name of file, 149
    - upgrading, 18
  - Easysoft ODBC for CODA
    - files used, 149
    - purpose of, xvii, 1
  - Easysoft ODBC Logging dialog box, 65
  - Easysoft ODBC login prompt
    - hiding and showing, 64
  - Easysoft ODBC Login Prompt, 62, 63
  - Easysoft ODBC Settings dialog box, 63
  - Easysoft ODBC Setup dialog box, 55, 59
  - Easysoft product licence. *See* licence
  - Easysoft Server Component, 5
    - checking installation, 33
    - installation
      - cancelling, 22
      - detecting problems, 23
      - directories created, 26
      - help, 22
      - procedure, 23
      - installation and upgrade, 18
        - privileges required, 18
        - precautionary back up, 23
        - preserving previous version, 24
        - recommended UIC, 25
        - re-installation restrictions, 23
        - storage space required, 18
        - transferring files to server, 21
        - unzipping files, 22
  - Easysoft Server Config Routine, 32
  - Easysoft SQL, 5
    - errors, 176
  - EASYSOFT.INI, 150
  - EASYSOFT.SQL directory, 31
  - EASYSOFT\_SQL\_ADMIN, 43
  - EASYSOFT\_SQL\_CATALOG, 61
  - EASYSOFT\_SQL\_CODA\_CATALOG
    - logical, 26, 30
  - EASYSOFT\_SQL\_DEMO\_CATALOG, 61, 62
  - EASYSOFT\_SQL\_SERVER logical, 20
  - EASYSOFT\_SQL\_TEMP, 151
  - editing in Windows. *See* Windows editing
  - ENQLM, 30
  - entry field. *See* text box
  - ESNET.LOG, 150
  - ESNET32.DLL, 149
  - ESODBC.LOG, 150
  - ESODBC32.DLL, 149
  - exact numeric data types. *See* data types, SQL
  - EXIT command, 45
  - EXPORT CSV command, 45, 167
  - extending product licences. *See under* licence
- ## F
- Fast Mode, 183
  - file data source, 58
  - files
    - accessed by a query, 65
  - final period, 72
  - Fixed assets, 14
  - fonts
    - used in document, xviii
  - Formula bar, 134
  - function calls
    - tracing, 67
  - functions. *See* scalar functions

**G**

General ledger. *See under* ledger  
 General Protection Fault. *See* GPF  
 GPF  
   problems due to, 142  
 GROUP\_HIERARCHY table, 108

**H**

HELP command, 43  
 Hierarchical Account Groups. *See* Account Groups  
 Host Administrator, 6, 43  
   ADMIN> prompt, 43  
   commands, 43–50  
   specifying current directory, 43

**I**

I/Os per query  
   number of, 65  
 IAS group, 11  
 IAS\$GROUP directory, 144  
 IAS\$SCRATCH logical  
   undefined, 26  
 IAS\$TEXT directory, 15  
 IASLINK, 6, 15, 69  
 IMPORT CSV command, 46, 167  
 index scoring mechanism. *See under* Query Execution Plan  
 indexes for CODA, 143  
 Input and Inray. *See under* CODA macro  
 installed image, removing, 20  
 Inray, 79, 111  
 INTRAY table, 111  
   difference between DETAILS. *See under* DETAILS table  
 Invoice Setup dialog box. *See under* CODA macro  
 Invoice Upload dialog box. *See under* CODA macro

**J**

journal, 11

**L**

ledger

  control account. *See* control account  
   creating with Easysoft ODBC, 104  
   defined, 9  
   General, 9  
   Nominal, 9  
   Purchase, 9  
   Sales, 9  
 Ledger  
   links, 14  
 Ledger identifier, 12  
 Ledger link  
   defined, 13  
 Ledger name, 12  
 LEDGER table, 104  
 ledger type  
   Expense, 12  
   Fixed asset, 12  
   Purchase, 12  
   Sales, 12  
 Level3, 15  
   creating with Easysoft ODBC, 106  
   defined, 13  
   relationship to Codaxref, 42  
 LEVEL3\_ID, 70  
 licence  
   adding, 44  
   directory, 26  
   expiry message. *See under* messages  
   extending, 34  
   installing, 31  
   removing, 34, 46  
 LICENCE directory, 26  
 licence file, 44  
 Licence Key  
   entering, 33  
   factors affecting, 34  
   valid characters, 44  
 Licence Server, 20  
 line 1, 14  
 Link dialog box, 100  
 links  
   increasing number of, 140  
 list box, defined, xix  
 LNKARGBLK.H, 15  
 LNKARGBLK.TXT, 15  
 LOG directory, 26  
 log files  
   directory, 26  
   server, 65  
 logicals

CODA, setting up, 28  
 determining, 141  
 IAS\$SCRATCH, undefined, 26

## M

masters  
   creating with Easysoft ODBC, 103  
 Masters, 14. *See also under individual names*  
 Media Headers, 15  
 messages  
   hiding, 150  
   licence expiry, 34  
 Microsoft Access  
   example of use, 99  
 Microsoft Excel  
   Formula bar. *See* Formula bar  
 Microsoft ODBC Administrator, 55  
 Microsoft Query  
   set table order, 183  
 modifying data sources. *See* data sources,  
   configuring  
 MOTHBALL table  
   difference between INTRAY and  
   DETAILS, 87  
 MULTINET, 25  
 multi-table queries  
   QEP score, 183  
   table order, 183

## N

NAME table, 104  
 NAME\_OWNER table, 105  
 names  
   restrictions, 147  
 names and addresses, 13  
 Names and addresses  
   adding to account groups, 108  
   creating with Easysoft ODBC, 104  
 Names and Addresses, 14  
 network connection  
   defining packet size. *See* packet size  
   packet acknowledgement. *See* packet  
   acknowledgement  
 network connectivity  
   file containing details, 149  
 network errors, 178  
 network object, 25  
 network protocol. *See* network transport

network statistics, 65  
 network transport  
   not available, 143  
   PC options, 61  
   server options, 25  
 Nominal accounts. *See* Nominals  
 Nominal ledger. *See under* ledger  
 NOMINAL table, 106  
 Nominal/ledger list, 14  
 NOMINAL\_ID, 70  
 Nominals, 15  
   creating with Easysoft ODBC, 106  
   relationship to Codaxref, 42  
 numeric functions. *See* scalar functions

## O

object number, 139  
 ODBC  
   architecture, 2  
   components  
     application, 2  
     data source, 2  
     determining details of, 67  
     driver, 2  
     driver manager, 2  
     purpose of, 1  
 ODBC Administrator icon, 55  
   location of, 142  
 ODBC conformance, 4  
   appropriate level of, 4  
   determining level of, 4  
 ODBC driver  
   information about, 67  
   listing, 67  
   multi-tier, 3  
   single-tier, 3  
 ODBC Driver  
   unable to select, 142  
 ODBC Drivers tab, 67  
 ODBC Setup  
   unexpected timeout, 142  
 ODBC Setup (Welcome) dialog box, 53  
 ODBC.INI, 149  
 ODBCAD32.EXE, 149  
 ODBCINST.INI, 149  
 on-line documentation  
   directory, 26  
 opening period, 72  
 Options dialog box. *See under* CODA macro

order data  
 by N/S/L3 (CODA), 41  
 override index selection, 184  
 ownership list, 14  
 creating with Easysoft ODBC, 105, 106

## P

packet acknowledgement  
 enabling, 64  
 packet size  
 defining, 64  
 password  
 not storing on PC, 147  
 relationship between components, 147  
 restrictions, 147  
 passwords and usernames  
 ADMIN user, 45  
 for Easysoft Catalog, 61  
 server, 61  
 PATHWAY, 25  
 PERIOD field  
 valid range, 72  
 PGFLQUOTA, 30  
 port number, 26, 139  
 posting, 11  
 pre-requisites to, 102  
 using DETAILS table, 79  
 primary key  
 indicated by font, 70  
 privileges. *See* CODA access rights  
 Profit & Loss Selection dialog box. *See under*  
 CODA macro  
 Profit and Loss report. *See under* CODA  
 macro  
 protocol. *See* network transport  
 Purchase ledger. *See under* ledger

## Q

QEP. *See* Query Execution Plan  
 Query Execution Plan, 65  
 index scoring mechanism, 182  
 quotas  
 recommended values, 30

## R

receive timeout. *See* timeouts  
 re-licensing. *See* licence, extending

remote command  
 specifying name of, 61  
 remote object  
 specifying name of, 61  
 remote service, 25  
 specifying name of, 61  
 Removal Key, 46  
 REMOVE LICENCE command, 46  
 REMOVE USER command, 47  
 removing  
 installed image. *See under* installed image  
 product licences. *See under* licence  
 reserved words, 164–66

## S

Sales ledger. *See under* ledger  
 scalar functions  
 defined, 169  
 supported by Easysoft, 169–75  
 syntax for use, 169  
 security. *See* CODA access rights  
 sequencing rule dependencies, 84  
 server  
 operating system versions, 18  
 server users, 44  
 service  
 creating, 138  
 SET COMPANY command, 47  
 Setup Budget Format dialog box. *See under*  
 CODA macro  
 SHOW CLIENTS command, 20  
 SHOW COMPANY command, 48  
 SHOW LICENCES command, 48  
 SHOW USERS command, 20, 49  
 shut down Licence Server. *See* Licence Server  
 Site Number, 32  
 generating, 32  
 sockets  
 increasing number of, 140  
 removing all, 140  
 Software License Agreement dialog box, 53  
 SQL Data Sources dialog box, 100  
 SQL data types. *See under* data types  
 SQL grammar conformance. *See* ODBC  
 conformance  
 SQL query, 65  
 SQL statements  
 supported by Easysoft SQL, 163  
 SQLGetFunctions, 4

SQLGetInfo, 4  
 SQLGetTypeInfo, 4  
 Start Copying Files dialog box, 55  
 string functions. *See* scalar functions  
 Subaccount, 15  
     creating with Easysoft ODBC, 106  
     defined, 13  
     relationship to Codaxref, 42  
 SUBACCOUNT table, 106  
 SUBACCOUNT\_ID, 70  
 SUBACCOUNT\_OWNER table, 106  
 SYSS\$MANAGER:SYSTARTUP\_V5.COM,  
     31  
 SYSS\$MANAGER:SYSTARTUP\_VMS.COM  
     , 31  
 system data source, 58  
 SYSTEM directory, 26  
 system files  
     directory, 26  
 system functions. *See* scalar functions  
 system startup file  
     editing, 31

**T**

Table Download dialog box. *See under*  
     CODA macro  
 tables  
     differentiating between databases, 64  
 Tax codes, 15  
 TCP/IP, 26  
     implementations of, 25  
 TCPWARE, 25  
 TEMP directory, 26  
 temporary files  
     directory, 26  
 text box, defined, xix  
 time functions. *See* scalar functions  
 timeouts, 64  
 toggling between buttons. *See under* buttons  
 trace of Easysoft network protocols, 150  
 trace of ODBC calls, 150

Tracing tab, 67  
 transmit timeout. *See* timeouts  
 Trial Balance dialog box. *See under* CODA  
     macro  
 Trial Balance report. *See under* CODA macro

## U

UCX, 25, 137  
     connection refused, 139  
 UIC. *See under* Easysoft Server Component  
     underneath lines. *See* detail lines  
 unexpected timeout. *See under* ODBC Setup  
 un-installing Easysoft software  
     on the server, 18  
 UPGRADE CATALOG command, 50  
 user data source, 58  
 User DSN tab, 57  
 User Information dialog box, 54  
 username  
     restrictions, 147  
 users  
     checking status of, 47  
     determining active users, 20  
     removing client user, 47

## V

VAX C Runtime library, 19  
 VAXCTRL, 19  
 VMSINSTAL, 18, 22

## W

Windows directory, 149  
 Windows editing, xix  
 Windows Socket errors, 179  
 Windows System directory, 149  
 Winsock  
     disable, 150