Easysoft Data Access

ODBC-ODBC Bridge

Installation Guide and User Manual

Version 13.

This manual documents version 1.2.n to 1.4.n of the Easysoft ODBC-ODBC Bridge.

Publisher: Easysoft Limited

Thorp Arch Grange
Thorp Arch
Wetherby
LS23 7BA
United Kingdom

CONTENTS

List of Figures 9

Preface 13

Intended Audience 14
Displaying the Manual 14
Notational Conventions 15
Typographical Conventions 16
Contents 17
Trademarks 18

Chapter 1 Introduction. 19

Why ODBC? 20
Driver Managers 22
Why the Easysoft ODBC-ODBC Bridge? 23
Where the Easysoft ODBC-ODBC Bridge fits in 24

Chapter 2 Installation. 29

Obtaining the Easysoft ODBC-ODBC Bridge 30
What to install 31
Installing on Windows. 35
Uninstalling on Windows 56
Installing on Unix 58
Uninstalling on Unix 78

Chapter 3 Connection 85

The connection process 86

Setting up the OOB Server. 96
Windows server setup 97
Unix server setup 105
Testing the OOB Server 111
Setting up the OOB Client 125
Windows client setup 126
Unix client setup 142

Chapter 4 **Configuration 157**

The Web Administrator. 158
Configuring the OOB Server under Windows 2000/NT 181
Configuring the OOB Server under Unix 188

Chapter 5 **Interfacing 197**

Introduction. 198
A Simple OOB Client in C 199
Apache/PHP. 201
Applixware 203
ColdFusion 204
Perl DBI DBD::ODBC. 205
Rexx/SQL 206
StarOffice 207
mnoGoSearch (formerly UDMSearch) 208
mxODBC 209
SQLPlus_hsODBC 210
unixODBC 211

Chapter 6 **Enterprise 213**

Introduction. 214

Reporting and Statistics 215
Auditing 223
Security 228
Usability 231
Availability 237
Feature table 240

Appendix A Technical Reference. 243

ODBC versions supported 244
Unsupported ODBC 3.5 functionality 244
Modifications to the API 244
Understanding ODBC diagnostic messages 246
Implementing ODBC diagnostics 247
Tracing 253

Appendix B Glossary 257

LIST OF FIGURES

Figure 1: Before and after ODBC 21
Figure 2: The Driver Manager as a dynamic linker. 22
Figure 3: The typical ODBC configuration 25
Figure 4: The Easysoft ODBC-ODBC Bridge 26
Figure 5: Bridging the network with the Easysoft ODBC-ODBC Bridge 27
Figure 6: Client Server version compatability. 32
Figure 7: The Welcome dialog box 35
Figure 8: The Information dialog box 36
Figure 9: The User Information dialog box 37
Figure 10: The Choose Destination Location dialog box 38
Figure 11: The Installation Options dialog box 39
Figure 12: The Choose Setup Type dialog box 40
Figure 13: The Select Components dialog box 41
Figure 14: The Software License Agreement dialog box. 42
Figure 15: The Exit Setup dialog box 43
Figure 16: The OOB Server Configurable Parameters dialog box 44
Figure 17: The OOB Server Administrator dialog box. 45
Figure 18: The Initial Security dialog box 46
Figure 19: The Start Copying Files dialog box 47
Figure 20: The Easysoft ODBC-ODBC Bridge DSN dialog box. 48
Figure 21: The License Manager window 50
Figure 22: The License Type dialog box 51
Figure 23: Select the product you are licensing. 51
Figure 24: The Authorization Number dialog box 52
Figure 25: The License Application dialog box 52
Figure 26: The Post Install Message dialog box. 54
Figure 27: The Setup Complete dialog box. 55
Figure 28: Linking a Linux client to an NT server 88
Figure 29: The Easysoft ODBC-ODBC Bridge Database Logon dialog box 92

Figure 30: The Easysoft ODBC-ODBC Bridge Server Logon dialog box 93
Figure 31: The ODBC Data Source Administrator User DSN tab 98
Figure 32: The ODBC Data Source Administrator System DSN tab 99
Figure 33: The Create New Data Source dialog box 100
Figure 34: The ODBC Microsoft Access Setup dialog box 101
Figure 35: The Easysoft ODBC-ODBC Bridge Server Services entry 103
Figure 36: The Configuration dialog box for a PostgreSQL data source. 108
Figure 37: The Create New Data Source dialog box 127
Figure 38: A blank Easysoft ODBC-ODBC Bridge DSN dialog box 128
Figure 39: The DSN set up for the Easysoft demo data source 131
Figure 40: The Easysoft ODBC-ODBC Bridge Settings dialog box 133
Figure 41: The Easysoft ODBC-ODBC Bridge Test DSN dialog box 135
Figure 42: The Link dialog box in Microsoft Access 137
Figure 43: The Select Data Source dialog box Machine Data Source tab 138
Figure 44: The demo.exe program. 140
Figure 45: Local ODBC attribute settings 146
Figure 46: Global ODBC attribute settings 149
Figure 47: A Linux client connected to an NT server. 150
Figure 48: The Configure Data Source dialog box. 152
Figure 49: The demo.exe program on Unix. 154
Figure 50: The Enter Network Password dialog box 160
Figure 51: The Statistics screen of the Web Administrator 161
Figure 52: The Configuration screen of the Web Administrator. 165
Figure 53: The Change Configuration screen of the Web Administrator. 175
Figure 54: The Data Sources screen of the Web Administrator 176
Figure 55: The Security screen of the Web Administrator 177
Figure 56: The Information screen of the Web Administrator. 179
Figure 57: The Client Hosts screen of the Web Administrator 180
Figure 58: Server parameters in the Registry. 186
Figure 59: Registry Flag settings. 187
Figure 60: Easysoft ODBC-ODBC Bridge Server Flag settings 195

Figure 61: General data on the Enterprise Web Administrator Statistics screen . 215
Figure 62: DSN data on the Enterprise Web Administrator Statistics screen 216
Figure 63: The Enterprise Web Administrator Client Hosts screen 217
Figure 64: The Enterprise Web Administrator Licenses screen 222
Figure 65: The Enterprise Web Administrator Connections per Minute graph . . . 227
Figure 66: The Enterprise Web Administrator Connections per Hour graph 228
Figure 67: The Enterprise Web Administrator Access Control table 229
Figure 68: The Enterprise DSN Realm Enter Network Password dialog box 234
Figure 69: The Enterprise Web Administrator Data Sources screen DSNs 235
Figure 70: The Enterprise Web Administrator Data Sources screen DSN data . . 236
Figure 71: The Enterprise Web Administrator Data Sources screen row data . . . 236
Figure 72: A Standard/Enterprise Easysoft ODBC-ODBC Bridge Feature table . . 242

This page left blank intentionally

PREFACE

About this manual

This manual is intended to cover the full range of requirements for anyone wishing to install, use, or configure the Easysoft ODBC-ODBC Bridge (OOB). Supplementary information is provided for those wishing to build ODBC applications that link through a driver manager, but please note that this manual is not an ODBC programming manual.

Chapter Guide

- **Intended Audience**
- **Displaying the Manual**
- **Notational Conventions**
- **Typographical Conventions**
- **Contents**
- **Trademarks**

Intended Audience

Sections written for the Microsoft Windows platforms require some familiarity with the use of buttons, menus, icons and text boxes, but should present no difficulties if you have any experience of Apple Macintosh computers, Microsoft Windows or the X Window System.

The Unix-based sections require experience of using a Unix shell and basic functions like editing a file. More complex activities are detailed more clearly, but it helps to understand how your system handles dynamic linking of shared objects.

NB Several technical documents are installed in addition to this manual, including a detailed list of Frequently Asked Questions (`FAQ.txt`), installation, configuration and interfacing information. These are located in `<InstallDir>/easysoft/oob/doc` under Unix and `<InstallDir>\Docs` under Windows. Please check all documentation thoroughly before contacting Easysoft with a query.

Displaying the Manual

This manual is available in the following formats:

- Portable Document Format (PDF), which can be displayed and printed using the Acrobat Reader, available free from Adobe at **http://www.adobe.com**.
- HTML (the format Easysoft recommend for viewing onscreen).

Notational Conventions

Across the range of Easysoft manuals you will encounter passages that are emphasized with a box and a label.

A *note box* provides additional information that may further your understanding of a particular procedure or piece of information relating to a particular section of this manual:

NB Note boxes often highlight information that you may need to be aware of when using a particular feature.

A *reference box* refers to resources external to the manual, such as a useful website or suggested reading:

REF For more manuals that use this convention, see the rest of the Easysoft documentation.

A *platform note* provides platform-specific information for a particular procedure step:

Linux In Linux you must log on as the `root` user in order to make many important changes.

A *caution box* is used to provide important information that you should check and understand, prior to starting a particular procedure or reading a particular section of this manual:

Caution! Be sure to pay attention to these paragraphs because Caution boxes are important!

Information has also been grouped within some chapters into two broad classes of operating system, Windows and Unix, for which side tabs are used to help you turn to the section relevant to you.

Typographical Conventions

To avoid ambiguity, typographic effects have been applied to certain types of reference:

- User interface components such as icon names, menu names, buttons and selections are presented in bold, for example:

  Click **Next** to continue.

  Where there is a chain of submenus, the following convention is used:

  Choose **Start > Programs > Command Prompt**.

- Commands to be typed are presented using a `monotype` font, for example:

  At the command prompt type `admin`.

- Keyboard Commands

  It is assumed that all typed commands will be committed by pressing the *<Enter>* key, and as such this will not normally be indicated in this manual. Other key presses are italicized and enclosed by angle brackets, for example:

  Press *<F1>* for help.

- File listings and system names (such as file names, directories and database fields) are presented using the `monotype` plain text style.

Contents

- **Introduction**

  An overview of the ODBC architecture and what the Easysoft ODBC-ODBC Bridge brings to it.

- **Installation**

  A step-by-step guide to installing the software.

- **Connection**

  Explains the connection process and shows how to set up an ODBC connection across the network.

- **Configuration**

  Describes the configuration options for the server in Windows and Unix, and the server configurable parameters.

- **Interfacing**

  Provides information about third-party programming languages, tools and applications that can be integrated with the Easysoft ODBC-ODBC Bridge.

- **Enterprise**

  Details the enhanced facilities offered within the Easysoft ODBC-ODBC Bridge Enterprise Edition.

- Appendices

  Comprising a Technical Reference and Glossary.

Trademarks

Throughout this manual, *Windows* refers generically to Microsoft Windows 95, 98, 2000, NT, XP, ME or 2003 Server, which are trademarks of the Microsoft Corporation. The X Window system is specifically excluded from this and is referred to as *The X Window System* or just *X*.

Note also that although the name UNIX is a registered trademark of The Open Group, the term has come to encompass a whole range of UNIX-like operating systems, including the free, public Linux and even the proprietary Solaris. Easysoft use Unix (note the case) as a general term covering the wide range of Open and proprietary operating systems commonly understood to be Unix ‘flavors’.

Easysoft and Easysoft Data Access are trademarks of Easysoft Limited.

CHAPTER 1 INTRODUCTION

Introducing the Easysoft ODBC-ODBC Bridge

Easysoft Data Access is a suite of programs that add significant value to your investment in ODBC. With Easysoft software you can connect applications on more platforms to more database systems than ever before.

The Easysoft ODBC-ODBC Bridge (OOB) enables ODBC calls to be made across the network from any Windows or Unix system to an ODBC data source running under Windows or Unix, permitting heterogeneous client-server operation and allowing client access to data wherever it is stored.

Please note that if you want to connect to local data sources from a remote Java application, you will need the Easysoft JDBC-ODBC Bridge *instead of* this software.

Chapter Guide

- **Why ODBC?**
- **Driver Managers**
- **Why the Easysoft ODBC-ODBC Bridge?**
- **Where the Easysoft ODBC-ODBC Bridge fits in**

Why ODBC?

Historically, corporate data was held on large, centralized computing resources that performed all the processing required on it. Changes to the business practice meant changes had to be made to the corporate mainframe system. Worse still was the problem of integrating two or more of these highly individual systems, for example in the event of a corporate merger.

As the desktop computer improved in power, users began to want to access corporate data in order to process it on their own desktop. The client-server architecture became a popular goal: the central computing resource (server) would produce a subset of its data for a user-friendly tool (client). The client would use desktop computing power to format and present the data.

Database application writers and their customers found themselves with a key problem: it was necessary to produce one version of a piece of software for each DataBase Management System (DBMS) they wished to use it with. Relational databases and SQL went part way toward alleviating the problem and for the first time there was a defined, open, standard language for querying databases.

In theory at least, it was possible to use the same language in dealing with databases from a variety of manufacturers and the X/Open consortium went on to define a Call Level Interface (CLI) so that programmers could effectively use SQL within their own programs.

ODBC is an API definition, compliant with ANSI SQL and X/Open’s SQL CLI, which allows an application to be written without considering the intricacies of the particular database engine to which it connects.

An ODBC *driver* takes care of all the database-specific code, if necessary transforming the structure of the underlying system into a relational framework.

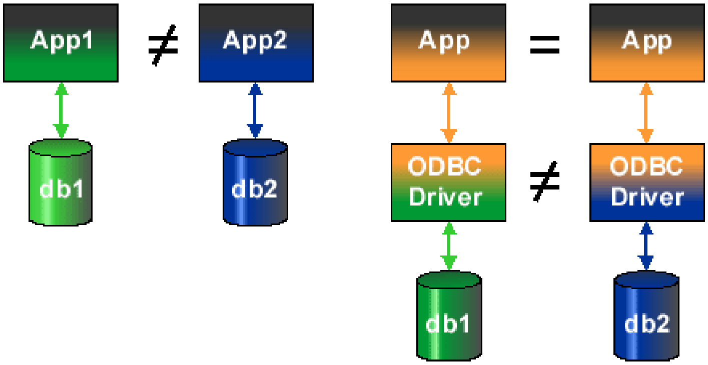


Figure 1: Before and after ODBC

Figure 1 on page 21 illustrates the principle of separating the driver from the application.

The left-hand side of the diagram shows how before ODBC, even if App1 and App2 were functionally equivalent, two programs were reuqired, one for each DBMS.

The right-hand side of the diagram shows how ODBC permits the DBMS-specific parts of a program to be separated from the part that fulfils the functional requirement, enabling the completed application to be attached to any DBMS that has a corresponding driver.

Driver Managers

The barest ODBC system would include an ODBC-conformant driver accessing some data, and an ODBC-conformant application, linked to the driver library.

If commercial applications were distributed in this way, users would need to re-link their applications to their chosen driver whenever they wanted to access a different data source.

Instead, the application program is linked to a *driver manager*, which loads the required driver at runtime. The driver manager also initializes the driver with a stored set of attributes.

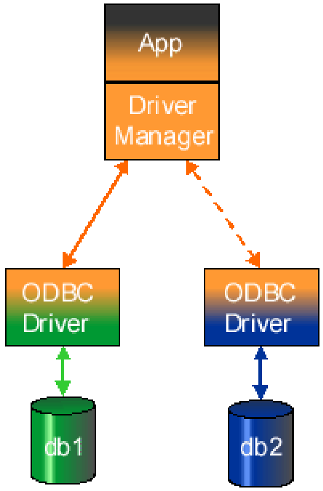


Figure 2: The Driver Manager as a dynamic linker

This approach provides three key results:

- Once developers have written applications to satisfy a business requirement, the application can be ‘plugged in’ to whatever database management system satisfies the technical demands.
- Administrators can connect a variety of applications (such as generic query tools) to their databases to browse and investigate the data.
- *Data access middleware* can be inserted between the ODBC application and driver to add strategic functionality such as joining heterogeneous databases into one data source or bridging a network.

Why the Easysoft ODBC-ODBC Bridge?

The Easysoft ODBC-ODBC Bridge is data access middleware that allows an application running on one platform to access an ODBC data source on another platform.

An Example Business Requirement

A sales manager keeps the company sales figures in an Excel spreadsheet on his networked Windows NT PC. He enjoys the consistent user interface and flexibility of the Windows system.

The company also has an intranet running on an Apache server under Linux, which displays the latest sales figures and updates them daily.

The system administrator values the round-the-clock stability of the Linux platform and the ability to freely tailor and tune the server software.

At present the sales manager emails the latest figures to the website administrator at the end of each day and then the website administrator updates the intranet.

A Solution using the Easysoft ODBC-ODBC Bridge

The Easysoft ODBC-ODBC Bridge allows a script on the web server to run a SQL select query directly on the NT machine, removing the need for regular human intervention.

A small script along with a simple client program submits SQL to the Easysoft ODBC-ODBC Bridge, which passes it across to the spreadsheet software on the NT machine. The script is set up to trigger automatically at the end of the day.

Where the Easysoft ODBC-ODBC Bridge fits in

The Easysoft ODBC-ODBC Bridge comes in two parts, a client and a server.

The OOB Client appears to the ODBC application just as any other ODBC driver, and the OOB Server connects to the database engine as an ODBC application.

This terminology may seem a little confusing at first, so it is important to consider each element step by step.

First, **Figure 3 on page 25** shows the typical ODBC set up. The application ("App") connects through a driver manager to an ODBC driver, which interfaces to the DBMS. The interface between the driver manager and the driver is defined by the ODBC API.

The Easysoft ODBC-ODBC Bridge can be seen as a black box that takes ODBC calls in at the top and passes them straight out again at the bottom.

This box can be inserted at *A* and as far as the application or the database are concerned, nothing has changed:

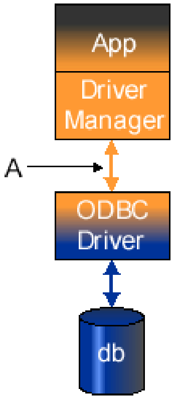


Figure 3: The typical ODBC configuration

What *has* changed is that now the driver manager, loads the OOB Client *and* the application and database no longer need to be on the same machine or software platform.

NB Connecting an ODBC application to a networked database is not new. For example, Microsoft SQL Server allows ODBC client applications to connect to a remote SQL Server data source. With the Easysoft ODBC-ODBC Bridge, however, the database can be any ODBC data source and the client may be on Windows, Linux, Solaris or any other supported platform.

Inside this ‘black box’ is a client-server pair that handles all the network operations, passing function references and parameters one way and return values the other:

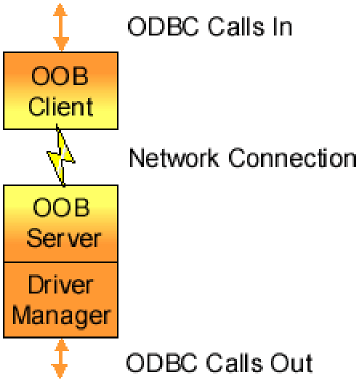


Figure 4: The Easysoft ODBC-ODBC Bridge

Figure 5 on page 27 shows the system set up across a network.

Comparing it with the model in **Figure 3 on page 25**, the database is replaced by a network connection on the client side and the OOB *Server* replaces the *ODBC application* on the server side.

From **Figure 5 on page 27** you can see that:

- to the application the OOB Client is a standard ODBC driver
- on the server the OOB Server is a standard ODBC application

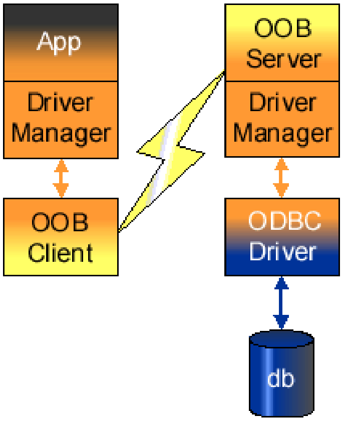


Figure 5: Bridging the network with the Easysoft ODBC-ODBC Bridge

REF

The definitive SQL CLI document is the **Open Group CAE Specification C451, ISBN 1-85912-081-4** (**http://www.opengroup.org/pubs/catalog/c451.htm**).

The **Microsoft ODBC 3.0 Programmer's Reference, ISBN 1-57231-516-4** explains ODBC usage in some detail.

CHAPTER 2 INSTALLATION

Installing the Easysoft ODBC-ODBC Bridge

This section explains how to install, license and remove the Easysoft ODBC-ODBC Bridge (OOB) on supported Windows and Unix platforms.

The Windows installation can be carried out by anyone with local administrator privileges for the target machine.

The Unix installation assumes you are, or have available for consultation, a system administrator.

Chapter Guide

- **Obtaining the Easysoft ODBC-ODBC Bridge**
- **What to install**
- **Installing on Windows**
- **Uninstalling on Windows**
- **Installing on Unix**
- **Uninstalling on Unix**

Obtaining the Easysoft ODBC-ODBC Bridge

There are three ways to obtain the Easysoft ODBC-ODBC Bridge:

- 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-ODBC Bridge section of the website and then choose the platform release that you require.

  First time visitors must complete the new user form and click **Register**. Note that your personal Internet options may require you to login and click **Continue** if you have previously registered.

- 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/odbc-odbc-bridge` directory and then choose the platform release that you require.

- You can order Easysoft software on CD by email, telephone or post (see **Contact Details**).

What to install

The Easysoft ODBC-ODBC Bridge consists of the OOB Client and the OOB Server.

There are shared files which are needed to install either of these components and there are documentation and example files which you are recommended to keep for future reference.

Unix distributions also include the third-party unixODBC driver manager (see **"unixODBC" on page 211**).

As the OOB Client and the OOB Server will typically be installed on different machines with different operating systems, you will need to perform the basic installation procedure twice, probably in different environments. You should first decide which part of the Easysoft ODBC-ODBC Bridge you are installing on which machine.

It is also important to consider that it may be necessary for Easysoft to make a change to the protocol used between the client and server. The following table shows which versions are guaranteed compatible with a ’Y’ and those versions which may not work together with a ’N’. The ’n’ is the major number and the ’m’ is the minor number as in an OOB release n.m.b (b, build numbers are not relevant) .

	Server version		
Client version	n.m	n.m+1	n+1.m
n.m	Y	Y	N
n.m+1	N	Y	N
n+1.m	N	N	Y

Figure 6: Client Server version compatability.

For example:

- Client version 1.1 will work with Server version 1.1, 1.2, 1.3 ...
- Client version 1.1 might not work with Server version 2.0.
- Client version 1.4 might not work with Server version 1.3.

Only the first two fields constitute the version number. The last digit is known as ‘build number’ and does not affect compatibility.

NB	If you connect a Client and Server with mismatched protocols, the Easysoft ODBC-ODBC Bridge will detect and report this immediately.

The name of the Easysoft ODBC-ODBC Bridge install file varies from platform to platform, but is of the form:

- `odbc-odbc-bridge-`*`x_y_z-platform`*`.exe` (Windows)

– OR –

- `odbc-odbc-bridge-`*`x.y.z-platform`*`.tar.gz` (Unix)

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

"*platform*" will vary depending on the operating system distribution you require and you may come across files of the form:

- `odbc-odbc-bridge-`*`x.y.z-platform-variation`*`.tar`

within specific Unix platforms, where "*platform-variation*" refers to alternative versions available for a single platform (e.g. *-mt* for thread-safe versions).

NB Select the highest release available for your platform within your licensed major version number (installing software of a different major version number requires a new Easysoft license).

Unix filenames may also be suffixed with `.gz` for a "`gzipped`" archive, `.bz2` for a "`bzip2ed`" archive, or `.Z` for a "`compressed`" archive.

NB If you download a Unix file using Windows, the browser may change the filename. For example, if you download a `.gz` file and Windows corrupts the filename, it may not be obvious that the file is "`gzipped`". Use "`file` *filename*" to find out the file type of the downloaded file.

You can now download a file and begin the installation process.

Caution!

As long as you stop all running applications (under Windows), or any software either from Easysoft or using Easysoft drivers (under Unix), it is safe to reinstall or upgrade the Easysoft ODBC-ODBC Bridge without uninstalling.

If you do uninstall, you should first back up any configuration data that you still need, as uninstalling some Easysoft products will result in this information being deleted (license details remain in place).

Refer to the section relevant to your platform to continue:

- **"Installing on Windows" on page 35**
- **"Uninstalling on Windows" on page 56**
- **"Installing on Unix" on page 58**
- **"Uninstalling on Unix" on page 78**

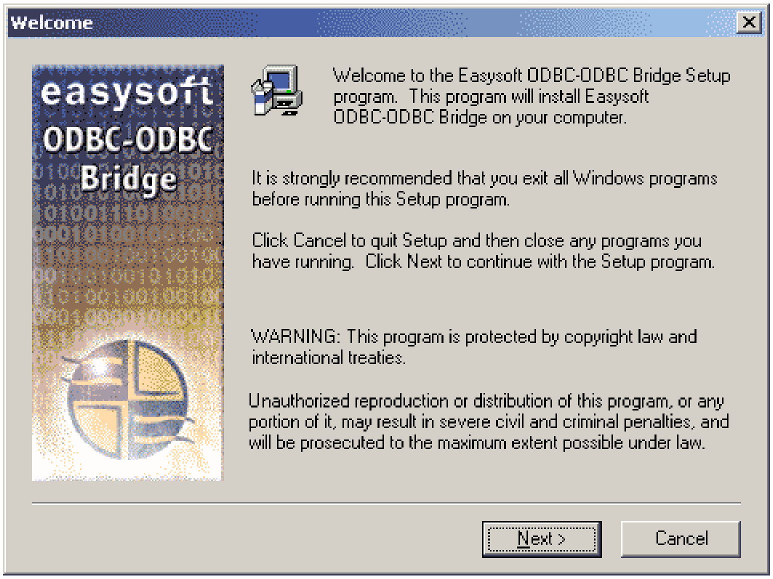
Win 9x	To run the Easysoft ODBC-ODBC Bridge on Windows 9x, you need WinSock2, which can be downloaded from **http://www.microsoft.com/windows95/downloads/contents/wuadmintools/s_wunetworkingtools/w95sockets2/**.

Installing on Windows

1. Execute the file distribution that you downloaded in **"Obtaining the Easysoft ODBC-ODBC Bridge" on page 30**.

Caution! Please shut down other Windows programs before installing. In particular, Microsoft Outlook can cause the installation routine to pause for several minutes when you start it.

There will be a short delay while setup prepares the wizard to guide you through the rest of the install procedure before the **Welcome** dialog box is displayed:



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

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

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

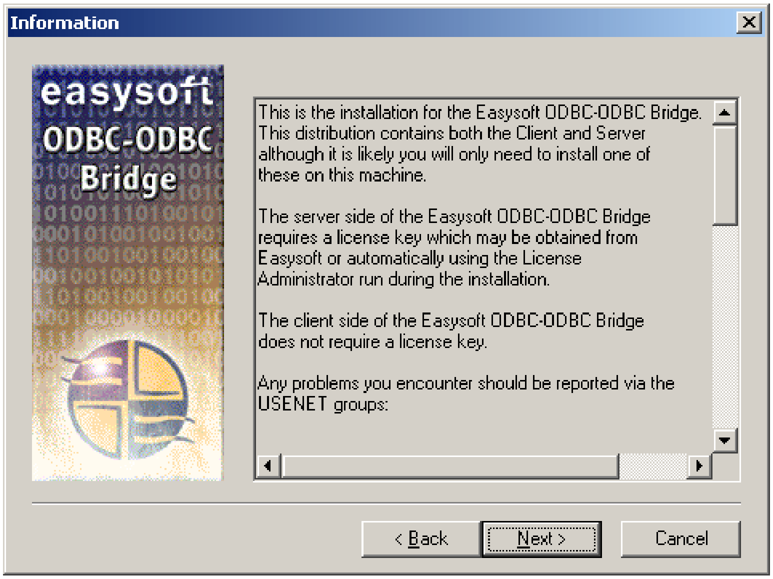
WARNING: This program is protected by copyright law and international treaties.

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

Figure 7: The Welcome dialog box

2. Click **Next** to continue.

The **Information** dialog box displays details about licensing and the components which are installed with the Easysoft ODBC-ODBC Bridge:



This is the installation for the Easysoft ODBC-ODBC Bridge. This distribution contains both the Client and Server although it is likely you will only need to install one of these on this machine.

The server side of the Easysoft ODBC-ODBC Bridge requires a license key which may be obtained from Easysoft or automatically using the License Administrator run during the installation.

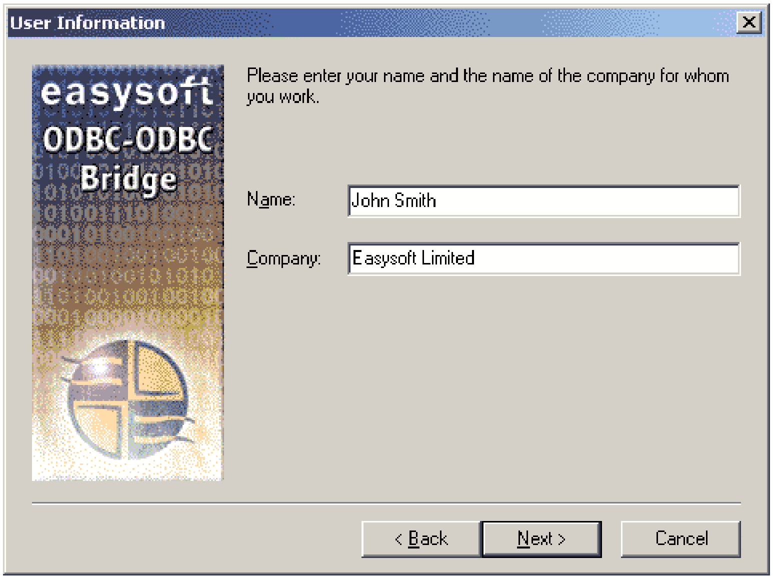
The client side of the Easysoft ODBC-ODBC Bridge does not require a license key.

Any problems you encounter should be reported via the USENET groups:

Figure 8: The Information dialog box

3. Click **Next** to continue.

The **User Information** dialog box is displayed:



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

Name: John Smith

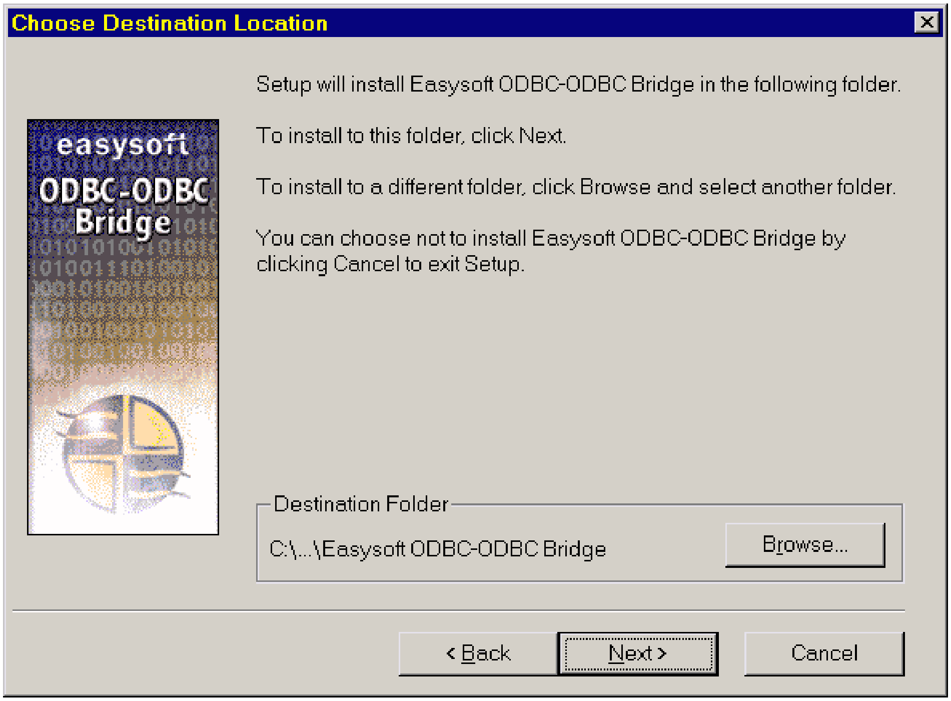
Company: Easysoft Limited

Figure 9: The User Information dialog box

4. Enter your name and the name of your company. Then click **Next** to continue.

NB	The name and company that you enter here will become the defaults in the License Manager later on.

The **Choose Destination Location** dialog box is displayed, where you are asked to choose where to install the Easysoft ODBC-ODBC Bridge files:



Choose Destination Location

Setup will install Easysoft ODBC-ODBC Bridge in the following folder.

To install to this folder, click Next.

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

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

Destination Folder

C:\...\Easysoft ODBC-ODBC Bridge

Figure 10: The Choose Destination Location dialog box

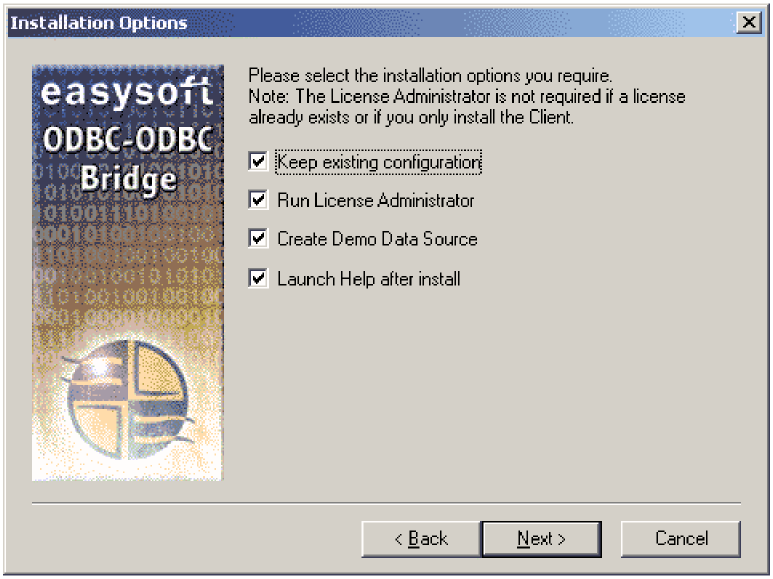
5. Click **Next** to install the application into the default directory location:

   `C:\Program Files\Easysoft\Easysoft ODBC-ODBC Bridge`.

   – OR –

   Click **Browse** and select an alternative directory location for the installation, before clicking **Next**.

The **Installation Options** dialog box is displayed:



Please select the installation options you require.
Note: The License Administrator is not required if a license already exists or if you only install the Client.

☑ Keep existing configuration

☑ Run License Administrator

☑ Create Demo Data Source

☑ Launch Help after install

Figure 11: The Installation Options dialog box

Choose whether you want to:

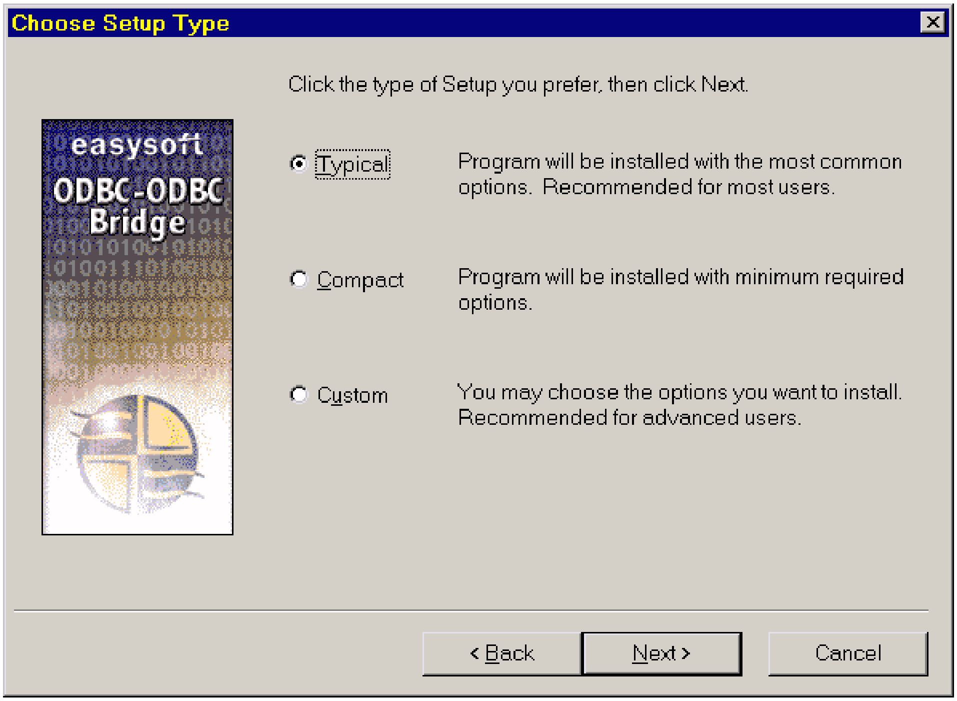
- keep your existing configuration settings if you are upgrading (this option is only visible if the installation program detects that you are upgrading). Some of the documented installation dialog boxes are omitted if you select this option.
- run the License Manager to obtain a license. You do not need to obtain a license if you are upgrading from an already licensed version or if you are installing the OOB Client only.
- create the demonstration data source which connects to data on a server machine at Easysoft (this is only relevant if you are installing the OOB Client).

- launch the Windows Help version of this manual.

These options are ticked when they are selected.

6. Click **Next**.

   The **Choose Setup Type** dialog box is displayed:



Click the type of Setup you prefer, then click Next.

● Typical — Program will be installed with the most common options. Recommended for most users.

○ Compact — Program will be installed with minimum required options.

○ Custom — You may choose the options you want to install. Recommended for advanced users.

Figure 12: The Choose Setup Type dialog box

7. Select **Typical** to install all components, documentation and examples (recommended for developers and administrators).

   – OR –

   Select **Compact** to install both client and server components, but no documentation or examples (recommended for user machines if your administrator already has a **Typical** installation elsewhere).

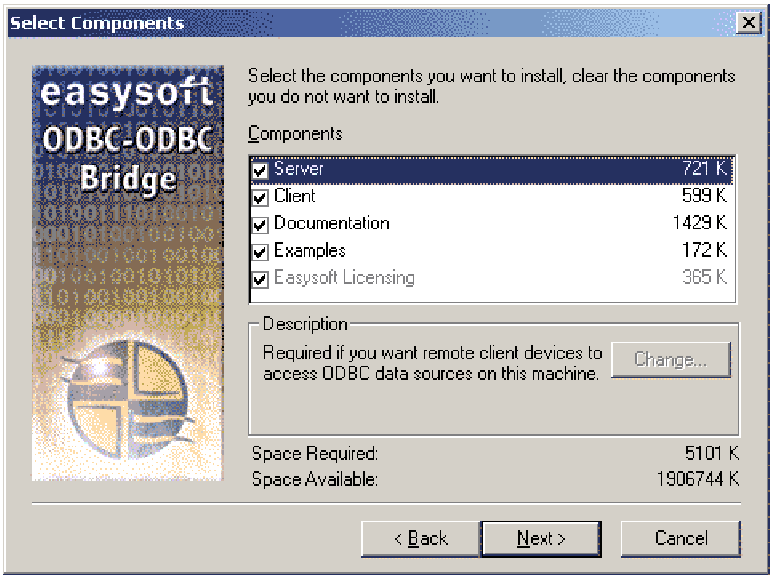
   – OR –

   Select **Custom** to choose which components to install (useful if disk space is low or for making a machine client or a server only).

8. Click **Next** to continue.

   If you selected **Typical** or **Compact**, go to **step 10 on page 42**.

   If you selected **Custom**, a checklist of items to install is displayed:



Select Components

Select the components you want to install, clear the components you do not want to install.

Components

☑ Server	721 K
☑ Client	599 K
☑ Documentation	1429 K
☑ Examples	172 K
☑ Easysoft Licensing	365 K

Description

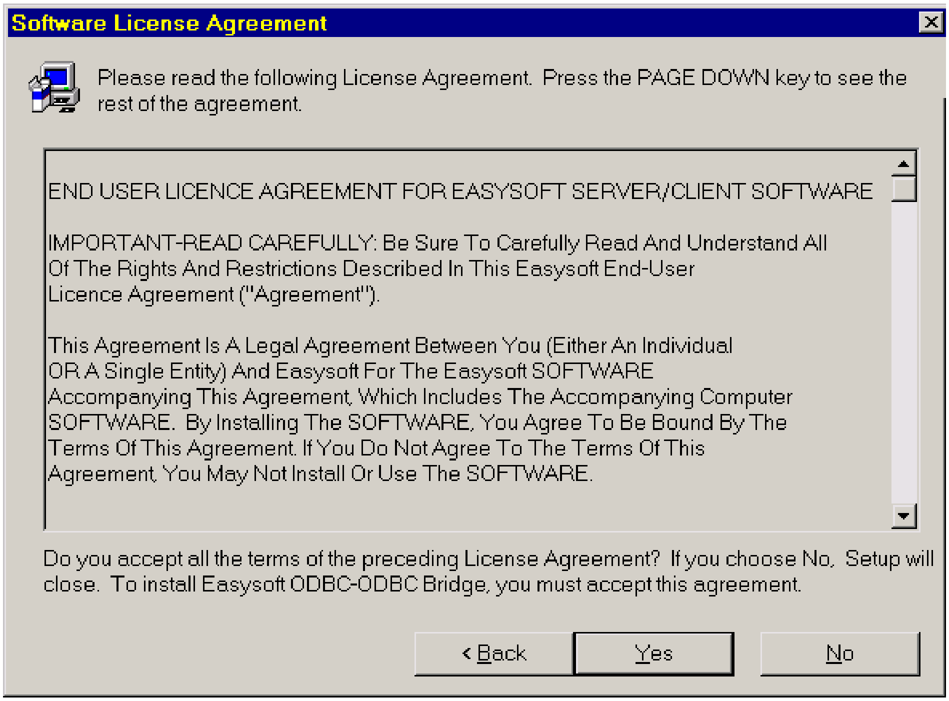
Required if you want remote client devices to access ODBC data sources on this machine.

Space Required: 5101 K

Space Available: 1906744 K

Figure 13: The Select Components dialog box

9. Select the items to install and click **Next** to proceed.
10. The **Software License Agreement** dialog box is displayed:



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

END USER LICENCE AGREEMENT FOR EASYSOFT SERVER/CLIENT SOFTWARE

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

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

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

Figure 14: The Software License Agreement dialog box

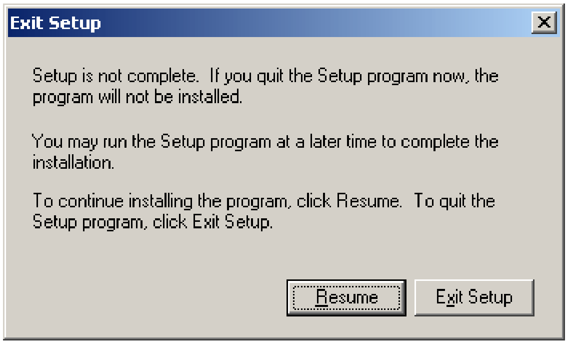
The license agreement varies depending on whether or not you chose to install the OOB Server:

11. Click **Yes** to continue with the installation if you accept the License Agreement.

    – OR –

    If you do not agree with the Easysoft License Agreement then click **No**.

    The **Exit Setup** dialog box is displayed and you may choose to resume or exit the installation.



Exit Setup

Setup is not complete. If you quit the Setup program now, the program will not be installed.

You may run the Setup program at a later time to complete the installation.

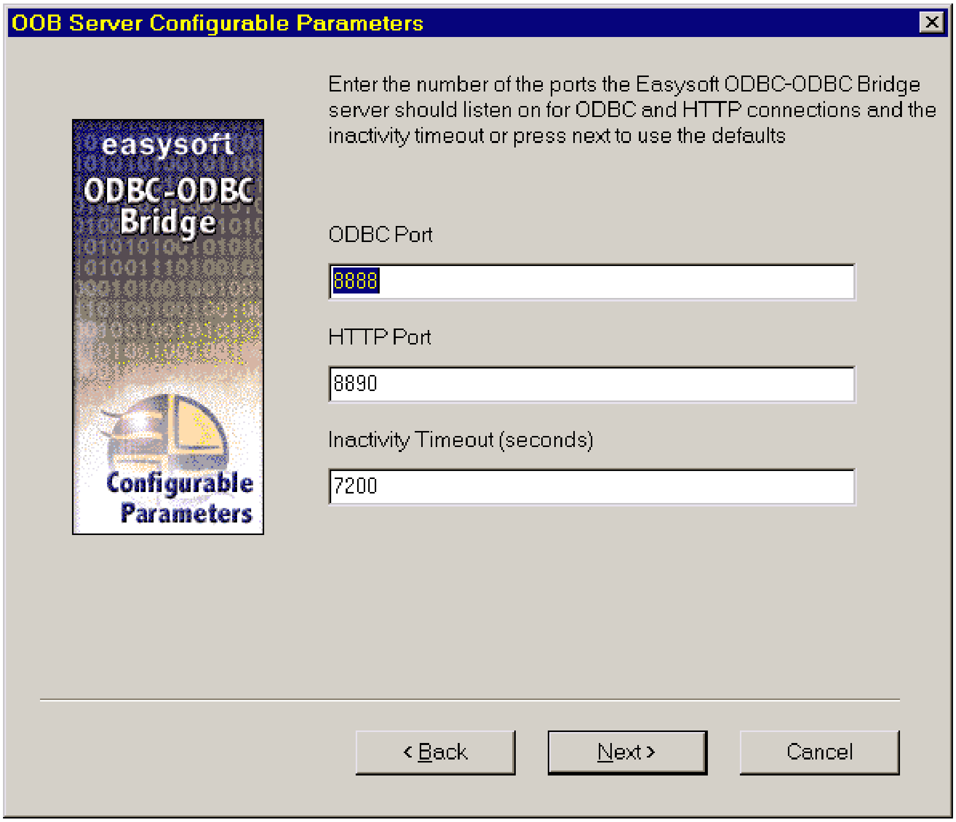
To continue installing the program, click Resume. To quit the Setup program, click Exit Setup.

Figure 15: The Exit Setup dialog box

12. If you chose not to install the OOB Server, go to **step 16 on page 47**.

    – OR –

    If you did not check the **Keep existing configuration** option on the **Installation Options** dialog box or if this is a new installation, the **OOB Server Configurable Parameters** dialog box is displayed:



OOB Server Configurable Parameters

Enter the number of the ports the Easysoft ODBC-ODBC Bridge server should listen on for ODBC and HTTP connections and the inactivity timeout or press next to use the defaults

ODBC Port: 8888

HTTP Port: 8890

Inactivity Timeout (seconds): 7200

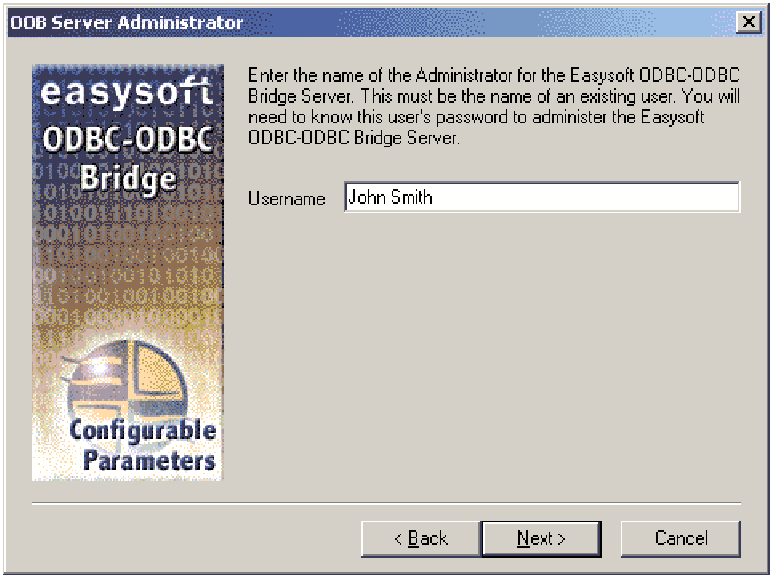
Figure 16: The OOB Server Configurable Parameters dialog box

The **OOB Server Configurable Parameters** dialog box contains three settings:

- the port where the OOB Server will listen for OOB Clients.
- the port at which the OOB Server Web Administrator will listen.
- the inactivity timeout (the number of seconds after which the OOB Server will disconnect inactive OOB Clients).

Accept the default values unless you have a port conflict or have some reason to change the timeout (see **"Choosing another Port or Service Name" on page 190**) and click **Next**.

The **OOB Server Administrator** dialog box is displayed:



Enter the name of the Administrator for the Easysoft ODBC-ODBC Bridge Server. This must be the name of an existing user. You will need to know this user's password to administer the Easysoft ODBC-ODBC Bridge Server.

Username: John Smith

Figure 17: The OOB Server Administrator dialog box

13. Enter the login name of an existing Windows user account as the Server Administrator user name.

    You also need to know the password for this user, because it will be required when you run the Web Administrator (see **"The Web Administrator" on page 158**).

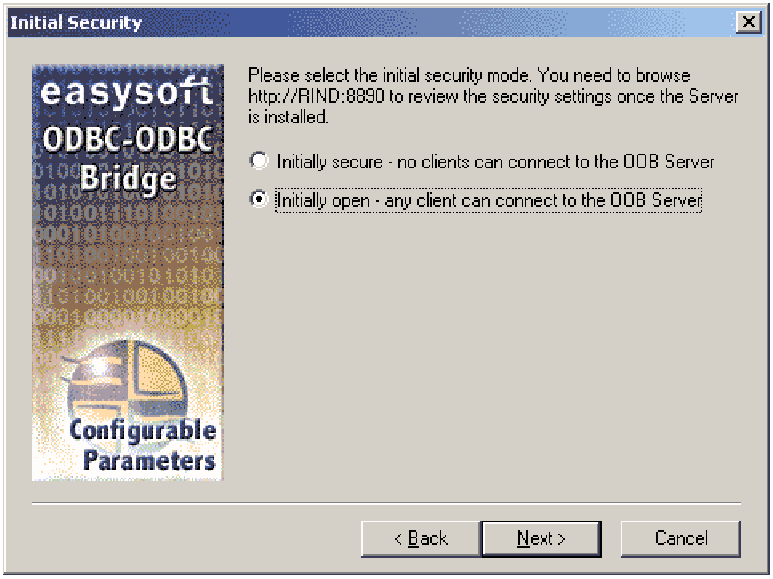
    To grant group access to the Web Administrator, you may wish to create a specific "`Administrator`" user.

To allow anyone to have access to the Web Administrator, leave the **Username** field blank to remove the requirement to log in.

Caution! You should consider carefully whether you wish to allow *anyone* on your network to have access to the Web Administrator in order to change Easysoft ODBC-ODBC Bridge settings.

14. Enter your chosen administrator user name and click **Next**.

The **Initial Security** dialog box is displayed:



Initial Security

Please select the initial security mode. You need to browse http://RIND:8890 to review the security settings once the Server is installed.

○ Initially secure - no clients can connect to the OOB Server

● Initially open - any client can connect to the OOB Server

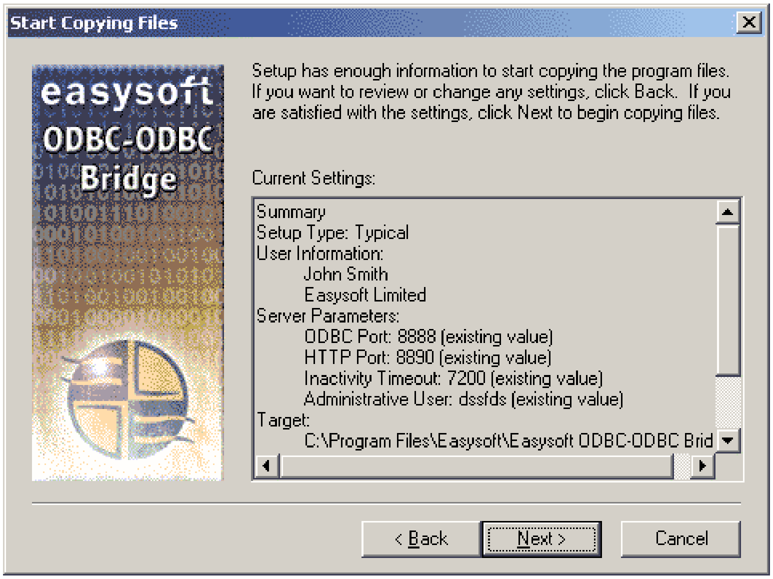
Figure 18: The Initial Security dialog box

The default value will allow any user to access the OOB Server.

The "`Initially secure`" option places an asterisk on the **Denied Access** section of the Web Administrator **Security** page, which will cause all OOB Client connections to be rejected.

This should be removed after installation and the clients that require access to the OOB Server specified individually (see **"The Security Screen" on page 177**).

15. Click **Next.**
16. The **Start Copying Files** dialog box is displayed, containing a summary of the settings you have chosen:



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

Current Settings:

```
Summary
Setup Type: Typical
User Information:
        John Smith
        Easysoft Limited
Server Parameters:
        ODBC Port: 8888 (existing value)
        HTTP Port: 8890 (existing value)
        Inactivity Timeout: 7200 (existing value)
        Administrative User: dssfds (existing value)
Target:
        C:\Program Files\Easysoft\Easysoft ODBC-ODBC Brid
```

Figure 19: The Start Copying Files dialog box

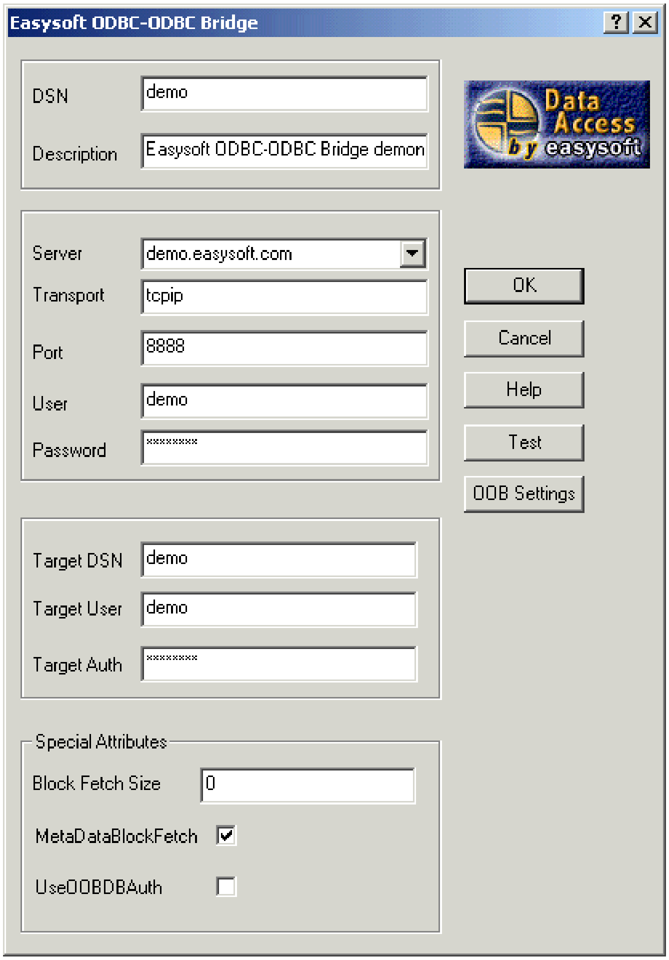
17. Click **Next**.

    There is now a short wait while the relevant Easysoft ODBC-ODBC Bridge components are copied and configured.

18. If you chose not to install the OOB Client, go to **"Licensing under Windows" on page 49**.

    – OR –

If you are installing the OOB Client and you checked the **Create Demo Data Source** option on the **Installation Options** dialog box, the **Easysoft ODBC-ODBC Bridge** dialog box now displays a configured DSN for an OOB Server on a machine at Easysoft.



Easysoft ODBC-ODBC Bridge

DSN: demo
Description: Easysoft ODBC-ODBC Bridge demon

Server: demo.easysoft.com
Transport: tcpip
Port: 8888
User: demo
Password: ********

Target DSN: demo
Target User: demo
Target Auth: ********

Special Attributes
Block Fetch Size: 0
MetaDataBlockFetch ☑
UseOOBDBAuth □

Figure 20: The Easysoft ODBC-ODBC Bridge DSN dialog box

If your firewall permits it and your server is connected to the internet, this allows you to verify that the OOB Client is functioning correctly.

19. Click **Test** to attempt a connection across the internet to `demo.easysoft.com`.

    If you already have an OOB Server running elsewhere you can refer to **"Connection" on page 85** and fill in the attributes for your own data source.

20. Click **OK** to close the **Easysoft ODBC-ODBC Bridge** dialog box.

Licensing under Windows

If you selected the **Custom** installation and elected not to install the OOB Server, then you do not need a license and the installation is complete.

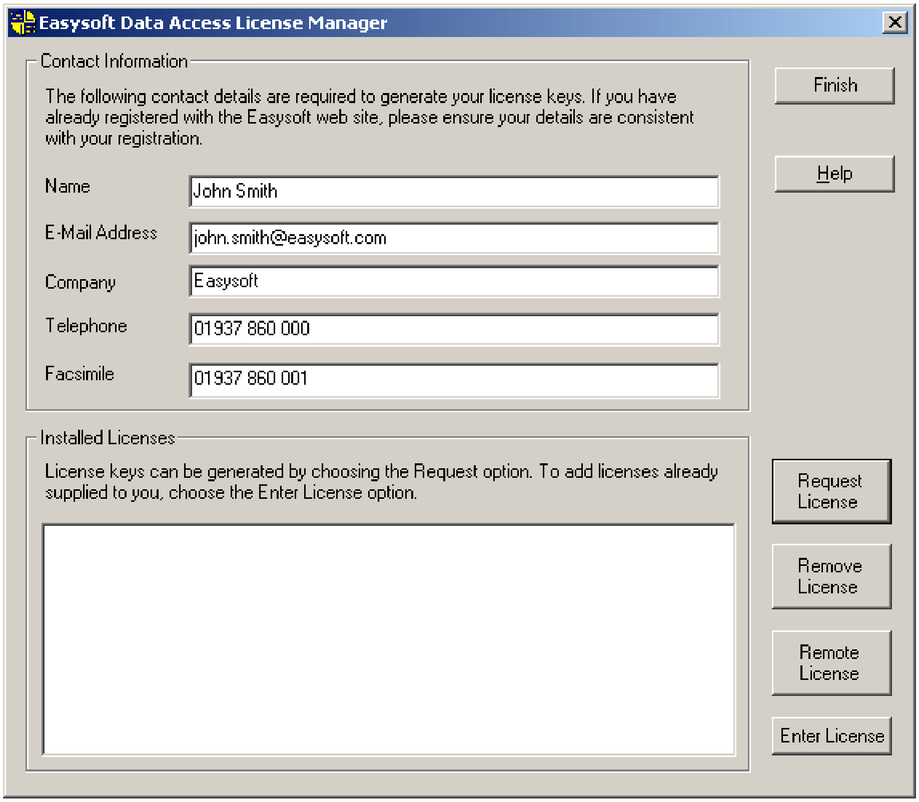
If you are upgrading and have licensed a previous version of the software, you do not need to repeat the licensing procedure.

– OR –

The install program starts the Easysoft License Manager (explained fully in the **Licensing Guide**), because you cannot use the OOB Server until a license is obtained.

The following types of license are available:

- a *free time-limited trial license* which gives you free and unrestricted use of the product for a limited period (usually 30 days).
- a *full license* if you have purchased the product. On purchasing the product you are given an authorization code which you should have to hand.



Contact Information

The following contact details are required to generate your license keys. If you have already registered with the Easysoft web site, please ensure your details are consistent with your registration.

Name: John Smith

E-Mail Address: john.smith@easysoft.com

Company: Easysoft

Telephone: 01937 860 000

Facsimile: 01937 860 001

Installed Licenses

License keys can be generated by choosing the Request option. To add licenses already supplied to you, choose the Enter License option.

Figure 21: The License Manager window

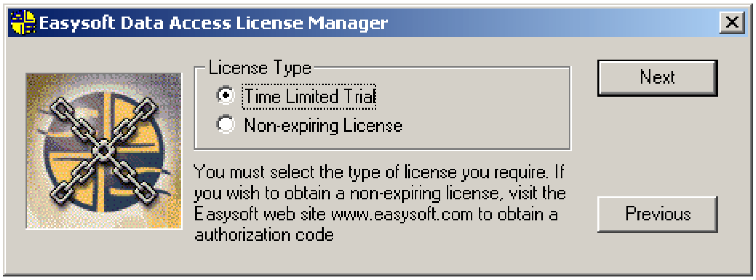
21. Enter your contact details.

    You MUST enter the **Name**, **E-Mail Address** and **Company** fields.

    The **Telephone** and **Facsimile** fields are important if you require Easysoft to contact you by those methods.

22. Click **Request License**.

You are asked for a license type:



License Type

● Time Limited Trial

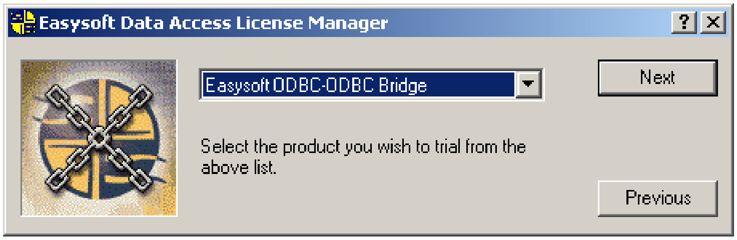
○ Non-expiring License

You must select the type of license you require. If you wish to obtain a non-expiring license, visit the Easysoft web site www.easysoft.com to obtain a authorization code

Figure 22: The License Type dialog box

23. For a trial license click **Time Limited Trial** and then click **Next**.

    The License Manager asks what software you are licensing:



Easysoft ODBC-ODBC Bridge

Select the product you wish to trial from the above list.

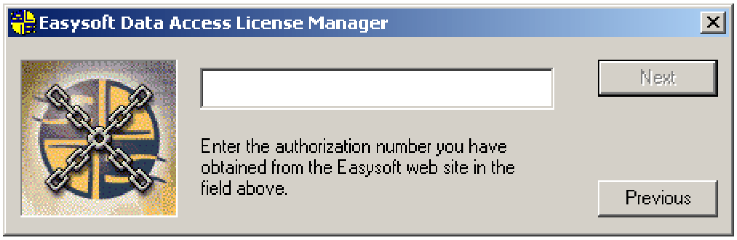
Figure 23: Select the product you are licensing

Select your required version of the Easysoft ODBC-ODBC Bridge (Standard or Enterprise, for example) from the drop-down list and then click **Next**.

– OR –

If you have obtained an authorization code for a purchased license, select **Non-expiring License** and then click **Next**.

The License Manager requests your authorization code:

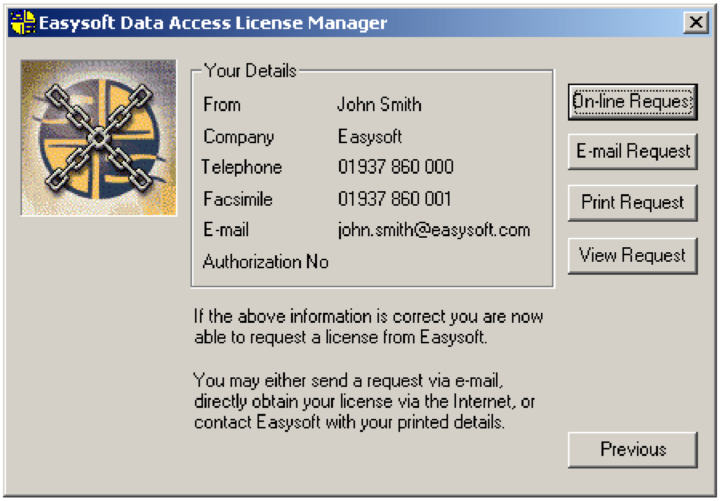


Enter the authorization number you have obtained from the Easysoft web site in the field above.

Figure 24: The Authorization Number dialog box

Enter the authorization code and then click **Next**.

24. The License Manager displays a summary of the information you entered and allows you to choose how to apply for your license:



Your Details

From	John Smith
Company	Easysoft
Telephone	01937 860 000
Facsimile	01937 860 001
E-mail	john.smith@easysoft.com
Authorization No	

If the above information is correct you are now able to request a license from Easysoft.

You may either send a request via e-mail, directly obtain your license via the Internet, or contact Easysoft with your printed details.

Figure 25: The License Application dialog box

Choose **On-line Request** if your machine has an internet connection.

The License Manager then sends a request to the Easysoft license server to activate your license key automatically. This is the quickest method and results in your details being entered immediately into our support database. You can now go to **step 25 on page 54**.

NB Only your license request identifier and contact details as they are displayed in the main License Manager screen are sent to Easysoft.

The remaining three options (**Email Request**, **Print Request** and **View Request**) are all ways to obtain a license if your machine is offline (i.e. does not have a connection to the internet).

Each of these methods involves providing Easysoft with information including your site number (a number unique to your machine) and then waiting to receive your license key.

Instead of emailing, faxing or telephoning your details to Easysoft, you can enter them directly onto the Easysoft web site and your license key will be emailed to you automatically.

To use this method, click **View Request** to display your site number and then go to **http://www.easysoft.com/sales/autolicense.phtml** in a web browser. Choose the type of license you require, enter your site number, click **Continue** and your license key will be emailed to you.

NB You can copy your site number from the **View Request** dialog box using `CTRL-C` and then paste it into the License Generator by using `CTRL-V`.

When you receive the license key, you can activate it either by double-clicking the email attachment or by clicking **Enter License** on the License Manager main screen and pasting the license key into the dialog box.

25. A message tells you how many licenses have been added.

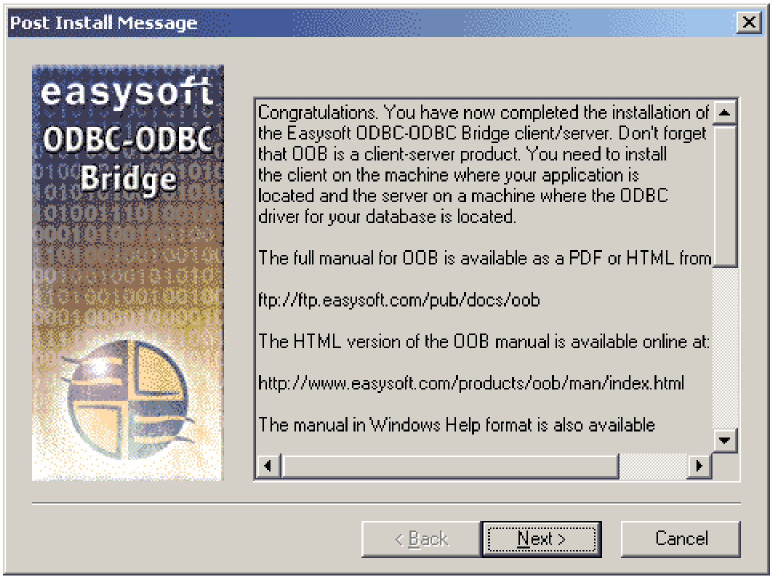
NB	If you use the **Email Request** option, the license key is emailed to the email address as displayed on the License Manager screen, not the `from:` address of your email.

For more information about the licensing procedure refer to the **Licensing Guide**.

NB	If you add a new license you will need to restart the OOB Server service in order to access the new details.

26. Click **Finish** in the License Manager to return to the install program.

The **Post Install Message** dialog box explains where you can find the Easysoft ODBC-ODBC Bridge documentation:



Congratulations. You have now completed the installation of the Easysoft ODBC-ODBC Bridge client/server. Don't forget that OOB is a client-server product. You need to install the client on the machine where your application is located and the server on a machine where the ODBC driver for your database is located.

The full manual for OOB is available as a PDF or HTML from

ftp://ftp.easysoft.com/pub/docs/oob

The HTML version of the OOB manual is available online at:

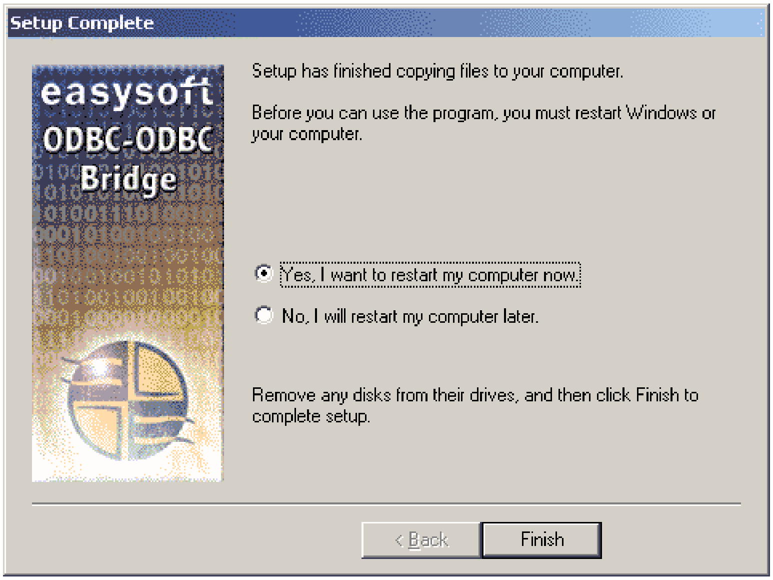
http://www.easysoft.com/products/oob/man/index.html

The manual in Windows Help format is also available

Figure 26: The Post Install Message dialog box

27. Click **Next** when you have read the **Post Install Message**.

    The **Setup Complete** dialog box is displayed:



Setup Complete

Setup has finished copying files to your computer.

Before you can use the program, you must restart Windows or your computer.

● Yes, I want to restart my computer now.

○ No, I will restart my computer later.

Remove any disks from their drives, and then click Finish to complete setup.

Figure 27: The Setup Complete dialog box

28. Click **Finish**.

    The installation is complete.

    Easysoft provide a small demonstration client program which can demonstrate the OOB Client in action by connecting across the internet to the server `demo.easysoft.com`. Refer to **"The Demo.exe Client" on page 139** for more details.

    After restarting your machine you should have an Easysoft program group with links to additional documentation, the product newsgroup and the Web Administrator.

Uninstalling on Windows

This section explains how to remove the Easysoft ODBC-ODBC Bridge from your system.

Caution! If you decide to remove a previous installation before upgrading to a new version a large amount of existing OOB Server configuration data may be deleted (machine access control details, for example).

1. Select **Start > Settings > Control Panel > Add/Remove Programs**.

   A list is displayed of applications that can be automatically removed.

2. Select **Easysoft ODBC-ODBC Bridge** and click **Add/Remove**.

3. Click **Yes** to confirm that you wish to remove the Easysoft ODBC-ODBC Bridge and all its components.

   The system begins to remove all the components. If shared components seem not to be required, you will be prompted to decide whether or not to delete them.

   NB The Windows install/uninstall procedure incorporates a mechanism in the registry to determine whether or not shared files are still required by other programs. Sometimes this database can become out-of-date, for instance if the user deleted an application directly, without using **Add/Remove Programs**, or the registry was ‘repaired’ after a system crash.

4. If you feel confident with the registry (i.e. your system has had relatively few programs installed and removed) you should click **Yes** or **Yes to All** to continue.

   – OR –

If you have any doubts (e.g. uninstall procedures have failed in the past) you should click **No** or **No to All**.

The uninstall process removes the Easysoft ODBC-ODBC Bridge components from your system.

NB	If files have been created in any of the installation directories then these directories will not be removed. In this case, the uninstall program will issue a warning and you can click **Details** to find out what directories remain.

5. On completion, click **OK** to go back to the Control Panel **Install/Uninstall** window.
6. The uninstall process is complete.

   Any licenses you obtained for the Easysoft ODBC-ODBC Bridge and other Easysoft products are held in the Windows registry.

   When you uninstall, your licenses are not removed so you do not need to relicense the product if you reinstall or upgrade.

Installing on Unix

Although this section covers a range of platforms and the precise output may vary from system to system, the installation process is essentially the same.

Installation requirements

The Unix installation routine has the following requirements:

- The Bourne shell (or BASH) must either be named (or linked to) `/bin/sh` or the first line of the `install` file updated to the correct location.
- Various common Unix commands such as `grep`, `awk`, `cut`, `ps`, `sed`, `cat`, `wc`, `uname`, `tr` and `find`.

  If any of these commands are missing they can be obtained from the Free Software Foundation (**http://www.fsf.org**).
- Depending on the platform, you will need up to 10Mb of free disk space for the installed programs and up to 10Mb temporary space for the installation files themselves.

Preparation

1. Log on to your Unix machine as the `root` user.
2. Download the Easysoft ODBC-ODBC Bridge (see **"Obtaining the Easysoft ODBC-ODBC Bridge" on page 30**).
3. Place the distribution file in a temporary directory on your Unix machine.

Extracting the installation files

4. Change to the directory in which the distribution file resides.

5. Extract the installation files from the distribution file:

   If the distribution file has been `gzipped` (i.e. the filename ends in `.gz`), then use:

   ```
   gunzip odbc-odbc-bridge-x.y.z-platform.tar.gz
   ```

   – OR –

   If the distribution file has been `bzip2ed` (i.e. the filename ends in `.bz2`), then use:

   ```
   bunzip2 odbc-odbc-bridge-x.y.z-platform.tar.bz2
   ```

   – OR –

   If the distribution file has been `compressed` (i.e. the filename ends in `.Z`), then use:

   ```
   uncompress odbc-odbc-bridge-x.y.z-platform.tar.Z
   ```

6. If the distribution file has not been compacted at all (i.e. the filename ends in `.tar`), then it is ready for extraction:

   ```
   tar -xvf odbc-odbc-bridge-x.y.z-platform.tar
   ```

   The `tar` program creates a directory with the same name as the `tar` file (without the final `.tar`) containing further archives, checksum files, a script called `install` and a text file called `INSTALL.txt`.

It also contains a versioned directory to ensure that any shared components already installed by other Easysoft products are only overwritten if those included in this distribution are newer.

NB If you do not wish to keep the original downloaded distribution file you can now delete it safely.

7. Change into the newly-created `odbc-odbc-bridge-`*`x.y.z-platform`* directory.

Caution! Check through the `INSTALL.txt` file before continuing. It gives full installation instructions for the Unix-literate, and if you are confident in the use and administration of your system, you can follow the instructions in the `INSTALL.txt` file instead of working through the remainder of this section.

8. There are two license agreement files provided in the archive: one that applies if you are installing just the OOB Client, and another that applies if you are installing the OOB Server or both the OOB Client and the OOB Server.

Caution! You must read and accept the terms of the applicable license to use the software. The license texts can be found in the files `Client-License.txt` and `Server-Client-License.txt`, respectively. Determine which applies to you, and be sure to understand the terms before continuing.

Beginning the Installation

9. Type:

```
./install
```

NB	During the installation, you are asked to answer some questions. The default reponse is displayed in square brackets [], which you can press *<Enter>* to accept or you can choose any of the alternative responses shown in round brackets () by typing the required response and then pressing *<Enter>*. Occasionally, the install program pauses to give you time to read the information displayed on screen. Press *<Enter>* to continue when you have read the current screen of information.

10. If you have read and agree to the **Easysoft License Agreement**, type `yes` and then press *<Enter>* to continue.

NB	You must type `yes`, not y, to continue.

11. The script pauses to allow you to read its output so far. Up to this point it has checked the following:

    - that you have the minimum set of Unix programs it requires
    - the platform you are running
    - any platform-specific components, such as the version of the C runtime library

12. Press *<Enter>* to continue.

    The script checks the archive package, with three possible outcomes:

    - The files are checked and they pass

- The files are checked and they fail
- The files are not checked because some component required for the check is not found

13. If the check failed because of missing components, enter y to continue regardless or n to quit and investigate the missing components.

    – OR –

    If the check was carried out and the files failed then the files have been damaged.

    Either:

    - return to **"Obtaining the Easysoft ODBC-ODBC Bridge" on page 30** and download the install archive again

    – OR –

    - call the Easysoft support team (see **Contact Details** for more information)

    If the files passed the check then you are asked for a directory in which to place the `easysoft` installation directory tree.

    The default is `/usr/local`, which would be the normal location to install software for system-wide use.

    If you wish to install the software in another directory then specify a directory name here.

    If you have any other Easysoft products already installed, then choose the same directory that you chose for the other Easysoft product(s).

The installation script will create a directory called `easysoft` under the directory you specify here, into which all Easysoft ODBC-ODBC Bridge files will be placed.

> **NB** The installation script can accept a path to a directory that does not exist, provided a parent directory exists. For example, if the directory `/usr/local` exists on your machine, the script can accept the non-existent directory `/usr/local/odbc` and will create it, but will not create `/usr/local/odbc/oob`. The script always creates an `easysoft` directory beneath the specified path.

14. If you have `root` permission and want a typical system-wide installation, press *`<Enter>`* to place the `easysoft` directory in `/usr/local/`.

    – OR –

    If you do not have `root` privileges, or wish to select a custom installation directory, type the desired directory and press *`<Enter>`*.

    The script now creates the `easysoft` directory under the directory you specified. If your chosen directory does not exist, you will be asked whether or not the script should create it. Enter `y` or `n`.

    If the chosen directory already contains an `easysoft` directory, the script warns that you may be installing over a previous Easysoft ODBC-ODBC Bridge installation (this conflict can also arise if you have another Easysoft product installed).

    Normally you should continue with the installation to install into the existing `easysoft` directory. Enter `y` or `n`.

If you chose a directory other than `/usr/local` the script creates a symbolic link `/usr/local/easysoft`, pointing to the `easysoft` directory in the directory you specified, in order to ensure that licensing will work.

15. Press *`<Enter>`* to continue.

Installing unixODBC and Common Files

Installing unixODBC permits ODBC applications on the local machine to choose a data source at runtime, as the driver manager loads the required driver.

If you do not have a driver manager on your system then the OOB Server will be restricted to only the data sources visible through a particular driver.

NB	unixODBC is an open source project sponsored by Easysoft and other industry members which is rapidly becoming the standard driver manager across the Unix data access community. Easysoft recommend it for use in all Unix ODBC installations (see **"unixODBC" on page 211**).

You should install the unixODBC driver manager if:

- the application(s) you will be using with ODBC specifically support unixODBC (e.g. Perl, PHP, Applixware)
- you intend using multiple drivers under the OOB Server
- you already have other ODBC drivers installed on this machine
- you intend using other Easysoft products in combination with the Easysoft ODBC-ODBC Bridge
- you want to use the unixODBC GUI configuration tools to define DSNs

For an OOB Client installation, unixODBC is particularly useful if you want to run ODBC applications with local ODBC data sources alongside the Easysoft ODBC-ODBC Bridge.

In general, Easysoft recommend installing unixODBC, as there is little to be gained by not doing so.

The installation will examine your system in an attempt to locate an already installed unixODBC. If unixODBC is already installed, the installation script can use your existing unixODBC, but you should bare in mind the following possible complications of doing this:

- If there are multiple copies of unixODBC on your system already you will have to choose one of them. If you pick the wrong one (i.e. not the one your applications are linked with, or not the one your run-time linker uses) the Easysoft ODBC-ODBC Bridge ODBC driver will not be visible to your applications until this is corrected. This situation is worse for the OOB Server.
- If you have built unixODBC yourself from sources you must make sure it has been configured correctly for use with your applications. For example, the unixODBC configuration script defaults to building a version linked with `pthreads` which is incompatible with Perl (on some platforms) and Apache/PHP (prior to Apache 2).
- You must make sure the version of the OOB Client is compatible with your existing unixODBC build. For example, you must use the `mt` (thread-safe) Easysoft ODBC-ODBC Bridge distribution with unixODBC built with `--enable-threads` (the default).

Easysoft ODBC-ODBC Bridge distributions contain matched builds of unixODBC and the Easysoft ODBC-ODBC Bridge, so installing the incorporated unixODBC build is often the safest policy.

16. Press *<Enter>* to continue and then choose whether to install unixODBC.
17. If you do not wish to install unixODBC, enter n and skip to **"Installing the OOB Client" on page 66**.

    – OR –

    If you wish to install unixODBC, enter y at the prompt.
18. The script pauses at this point. Press *<Enter>* to continue.

    The script extracts the unixODBC files.
19. The script pauses again. Press *<Enter>* to continue.

    The parts of the Easysoft ODBC-ODBC Bridge installation common to both the OOB Client and the OOB Server are extracted.

Installing the OOB Client

The script asks you whether you want to install the client component of the Easysoft ODBC-ODBC Bridge. This is required if you want to connect an ODBC application or script on the local machine to a remote data source.

20. Enter n to skip installing the OOB Client and go to **"Installing the OOB Server" on page 70**.

    – OR –

    Enter y to install the OOB Client.
21. The script pauses at this point. Press *<Enter>* to continue.

    The script extracts the OOB Client files.
22. Press *<Enter>* to continue.

The script displays important information regarding the dynamic library locations at this point.

Unix	Note these directories. Before you can use the OOB Client you will need to set up and export LD_LIBRARY_PATH, SHLIB_PATH, DYLD_LIBRARY_PATH or LIBPATH (depending on your operating system and run-time linker) to include these paths.
Linux	The script sets up the dynamic linker with the paths to the client shared objects. If you are not the `root` user then you will need to configure the dynamic linker manually. Make a note of any paths you are given because you will need to get a `root` user to set it up for you. If you have the file `/etc/ld.so.conf`, then you need to insert the paths given here, then run `/sbin/ldconfig` to re-read the changes.
Free BSD	The script sets up the dynamic linker with the paths to the client shared objects. If you are not the `root` user then you will need to configure the dynamic linker manually. Make a note of any paths you are given because you will need to get a `root` user to set it up for you. If you have the file `/etc/defaults/rc.conf`, then you need to insert the paths given here and then run `/sbin/ldconfig -m <paths>` to re-read the changes.

23. Press *`<Enter>`* to continue.

Configuring the OOB Client in unixODBC

If unixODBC was installed as part of this installation, or unixODBC was found to be already installed, the script will offer to set up the OOB Client under unixODBC.

This allows any ODBC application which is using (or can use) the unixODBC driver manager to connect to remote data sources through the Easysoft ODBC-ODBC Bridge.

24. If you do not want to set up the Easysoft ODBC-ODBC Bridge under unixODBC or you do not have unixODBC, enter n and go to **"Installing the OOB Server" on page 70**

    – OR –

    If you have unixODBC installed on your system, and wish to use it with your OOB Client, enter y.

    The script runs `odbcinst`, the command for installing data sources under unixODBC.

25. If the `odbcinst` program is not in its expected location, you are asked to provide the directory into which unixODBC was installed.

    If you do not know the answer, enter q to abandon the attempt to configure unixODBC and skip to **"Installing the OOB Server" on page 70**.

    – OR –

    If known, supply the unixODBC install path.

    The script checks whether or not the Easysoft ODBC-ODBC Bridge has already been set up under unixODBC.

26. If the unixODBC driver manager has already been configured for the Easysoft ODBC-ODBC Bridge then a warning is given.

Caution!

If unixODBC has been previously configured with the Easysoft ODBC-ODBC Bridge then it will continue to access Easysoft ODBC-ODBC Bridge software from the directory in which it was originally installed.

If the current install directory is different then unixODBC will pick up the old version of the Easysoft ODBC-ODBC Bridge, resulting in incorrect behavior.

The script now calls `odbcinst` with the relevant details for the OOB Client.

NB

The messages `"Target directory is /usr/lib"` and `"added to ODBC.INI"` displayed whilst the `odbcinst` script is running are inaccurate and should be ignored.

If this part of the installation fails, refer to the unixODBC documentation for details of how to manually reconfigure the driver manager.

REF

Comprehensive unixODBC documentation can be found at **http://www.unixodbc.org**.

The settings that the install script attempted to set can be found in the file `unixODBC.template`.

A server at Easysoft (`demo.easysoft.com`) runs the OOB Server continuously so that users can connect to verify their OOB Client is working correctly.

The install script now offers to create a demo data source under unixODBC that refers to this demo Easysoft OOB Server.

27. If you do not want to create a demo data source, enter n at the prompt and go to **"Installing the OOB Server" on page 70**.

    – OR –

    If you want to create a demo data source for verifying the OOB Client installation, then enter y and continue.

    The script calls `odbcinst` again, this time with the data source details.

    Once the installation is complete, you should be able to connect to `demo.easysoft.com` through the Easysoft ODBC-ODBC Bridge data source `demo`.

INSTALLING THE OOB SERVER

The script asks whether you wish to install the OOB Server, which allows you to make data sources on the local machine available to remote ODBC applications using the Easysoft ODBC-ODBC Bridge driver.

28. If you wish to skip the OOB Server install, enter n and go to **"Completing the installation" on page 75**, which displays a final checklist.

    – OR –

    Enter y to continue.

29. The script pauses at this point. Press *<Enter>* to continue.

    The files necessary for the OOB Server are extracted.

30. Press *<Enter>* to continue.

Licensing the Easysoft ODBC-ODBC Bridge

31. The script prepares to license the product and then asks if you would like to run the License Manager.

    If you intend to license the software after finishing the installation, enter `n` and go to **step 40 on page 73**.

    – OR –

    If you want unrestricted use of the software, either via a free time-limited trial license or with a full (purchased) license, enter `y` and continue.

    The License Manager displays a menu of options.

32. Enter the number corresponding to the Easysoft ODBC-ODBC Bridge.

    The License Manager requests some contact information.

33. Enter your **Name**.
34. Enter your **Company Name**.
35. Enter at least one of **Email**, **Phone** and **Fax** (preferably all three).
36. At the **Ref** prompt, do either of the following:

    - If you want a trial license, leave it blank and press *<Enter>*
    - If you want a full license, enter your authorization code and then press *<Enter>*

37. The License Manager displays a menu of options for acquiring your license.

    If you have an internet connection you should select `1, Automatic`.

This is the quickest and easiest method, unless your firewall or other network obstacles prevent the message getting through.

NB	Opening up port `8884` via `sysadmin` will allow you to get around this if you wish.

If you do not have an internet connection, or the `Automatic` option fails, select option 2 to write the required information to a text file in the current working directory called `license_request.txt`, which you will then need to view to obtain a license.

38. The License Manager displays the menu again. Select `0` to exit.

NB	The **View Existing Licenses** option will not display any license you obtained during this License Manager session. You must quit the License Manager and allow the install script to finish applying the license.

If you chose the automatic licensing method and licenses are retrieved, the licensing script outputs them to a file called `licenses.out`. If the install script detects possible new licenses in the file `licenses.out` it asks whether you want to add them to the license database.

39. If you obtained licenses and you want to add them now, enter y.

– OR –

If you did not obtain licenses, enter n.

– OR –

If you do not see this message, then read on. You will need to complete the offline licensing procedure as explained in **"Completing the offline licensing procedure" on page 75**.

40. If you are not `root` then the installation terminates. Go to **"Completing the installation" on page 75**.

    – OR –

    The script now begins to configure the OOB Server under the `{x}inetd` “super-server” and asks whether you would like to install the `services` and `inetd/xinetd` entries.

NB	Some new Linux distributions (in particular) use `xinetd`, rather than `inetd`. The installation recognizes and correctly handles the configuration of either. For more information about `xinetd` visit the web site at **http://www.xinetd.org**.

41. If you want to skip these changes, enter n and the installation terminates. Go to **"Completing the installation" on page 75**.

    – OR –

    Press *`<Enter>`* to continue.

    The script now amends the `/etc/services` and `{x}inetd` configuration files.

NB	If you are installing the Easysoft ODBC-ODBC Bridge for use with other Easysoft products then the `services` and `{x}inetd` entries MUST be configured.

42. If either of these files is missing or protected against writing, then you must supply their locations as prompted.

NB	You must use full path and file names in order to specify the relevant configuration file, and not just the file names themselves.

The script looks for an existing entry in the `services` file for the default service name of the OOB Server. If one already exists then you are prompted for a decision.

43. If there is no service name conflict, then go to **"Completing the installation" on page 75**. If there is a service name conflict, read on.

    If you are reinstalling the Easysoft ODBC-ODBC Bridge (to replace an older installation), enter `r` to replace the existing entry.

    – OR –

    If you want to have two different Easysoft ODBC-ODBC Bridge services running beside one another and you wish to define a new service name for the OOB Server, then type "*`d`*", press *`<Enter>`*, then enter your chosen new name for the OOB Server service.

    The script creates backups of both the `services` (to `services.pre_OOB`) and `{x}inetd` (to `/etc/inetd.conf.pre_OOB` or `/etc/xinetd.conf.pre_OOB`) configuration files.

44. The script looks for an existing entry in the `services` file for the OOB Server’s default port. If an entry already exists then you are asked to choose another port. Enter another port number and the script will check whether the new value conflicts with any other services.

45. If the script cannot determine what shell to use for the `{x}inetd` configuration file you must enter one at the prompt. It will now be read again so that any changes take effect, and the OOB Server startup script is installed.

Completing the installation

46. Press *<Enter>* if necessary to return to the shell prompt.

    Don't forget:

    - If, for some reason, you were not logged in as the `root` user, then now is the time to arrange for the dynamic linker to be set up with those paths
    - If the install failed to register the oob client driver with unixODBC then you should do it manually now.

47. You can remove the installation files. Unless you specified the installation directory (at **step 14 on page 63**) to be within the temporary directory, then you can safely remove the temporary directory and all its contents.

NB	Any warnings generated during the installation are appended to the `./warnings` file and a message output containing a list of all the warnings. You should review this file and satisfy yourself that none of the warnings have adversely affected your installation. You should mail this file to Easysoft support if you are unsure. In particular, one warning you may see is caused by the installation attempting to untar a file which is in use. If you see this warning it will be necessary to make sure all applications using the file in question are stopped and the installation is rerun.

Completing the offline licensing procedure

If at **step 37 on page 71** you chose to write the licensing information to a file, you still need to complete the licensing procedure before the OOB Server can be activated.

The licensing information is written to the `license_request.txt` file. This file contains information including a site number (a number unique to your machine) which Easysoft require before a license key can be issued.

1. Do any one of the following:
   - Display the `license_request.txt` file (e.g. using `cat license_request.txt`) and note the site number. Now run a web browser and go to **http://www.easysoft.com/sales/autolicense.phtml**. Log in to the Easysoft web site. On the License Generator screen, choose the type of license you want, then enter your site number and click **Continue**. You can now close the web browser. You will shortly receive your license key(s) via email.
   - Email the file `license_request.txt` to **autolicense@easysoft.com**. Your license key(s) will be emailed to you automatically.
   - Email the file `license_request.txt` to **license@easysoft.com**. A member of the Licensing Department will email the license keys(s) to you.
2. When you receive your license key(s), append them to the file `/usr/local/easysoft/license/licenses`, removing any `LIC:` prefixes.

NB	The Easysoft license responder puts an attachment in its outgoing emails that allows Windows users to activate their licenses with a double-click. If you read your email in Windows, this attachment will be visible but it will not work for licensing Unix versions of the software.

The Easysoft ODBC-ODBC Bridge is now licensed and you can begin using it.

If you need further information about licensing, please refer to the **Licensing Guide**.

Using the OOB client on a non-Linux platform

If you are installing to a non-Linux platform, before you can use the OOB Client you must set and export the shell environment variable LD_LIBRARY_PATH, LD_RUN_PATH, DYLD_LIBRARY_PATH or SHLIB_PATH to

```
<InstallDir>/easysoft/oob/client:<InstallDir>/
easysoft/lib
```

where *<InstallDir>* is the directory in which you chose to install the Easysoft ODBC-ODBC Bridge.

If you accepted the default install directory you might use:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/easyso
ft/oob/client:/usr/local/easysoft/lib
export LD_LIBRARY_PATH
```

NB

The exact command you need to use to set and export an environment variable depends on your shell and the variable that needs to be set depends on the run-time linker used on your Unix platform. Refer to the platform documentation for your current shell and `ld(1)`, `dlopen` or `ld.so(8)`.

If unixODBC has been installed then you will also need to add `<InstallDir>/easysoft/unixODBC/lib` to your LD_LIBRARY_PATH, LD_RUN_PATH, DYLD_LIBRARY_PATH or SHLIB_PATH shell environment variable.

Uninstalling on Unix

To uninstall the Easysoft ODBC-ODBC Bridge under Unix:

- If unixODBC is installed, the Easysoft ODBC-ODBC Bridge driver must be removed from its database.
- The entries for the Easysoft ODBC-ODBC Bridge must be manually removed from the Unix `services` and `{x}inetd` configuration files (this requires `root` access).

Caution!

This is only required if the Easysoft ODBC-ODBC Bridge is not going to be upgraded, or is to be upgraded using a different configuration.

If the Easysoft ODBC-ODBC Bridge is going to be upgraded with the existing configuration then do NOT make these edits.

- If the system has a dynamic linker (such as `ld.so` on Linux), the Easysoft ODBC-ODBC Bridge directories must be removed from the dynamic linker search path (this may require `root` access, depending on the mechanism used by the platform).
- The Easysoft ODBC-ODBC Bridge install directory tree must be removed (this requires the same privileges as the user who performed the installation, which is normally `root`).

A step-by-step guide follows:

1. Close down all client programs attached to your service.
2. Log in as `root`.

Removing from unixODBC

3. Check whether the Easysoft ODBC-ODBC Bridge is configured under unixODBC by typing:

   ```
   odbcinst -q -d
   ```

4. If "OOB" is returned in the output then remove the Easysoft ODBC-ODBC Bridge by typing:

   ```
   odbcinst -u -d -n OOB
   ```

   If a message is displayed about a reduced usage count, repeat this step until `odbcinst` states that the Easysoft ODBC-ODBC Bridge has been removed.

Removing the service entries

5. Make a backup of the `/etc/services` configuration file.
6. Open `/etc/services` and look at the end for a line of the format:

   ```
   esoobserver 8888/tcp   # Easysoft ODBC-ODBC Bridge
   ```

   If more than one Easysoft ODBC-ODBC Bridge service has been created then there will be more than one line in the `services` file.

   Each such line should have a comment referencing the Easysoft ODBC-ODBC Bridge.

   `esoobserver` is the default name for an Easysoft ODBC-ODBC Bridge service and any additional Easysoft ODBC-ODBC Bridge services will display the alternative names given to them.

   Note down the names of the services you remove at this stage, so that if there are problems you can look them up in your `services` backup file and re-introduce them.

NB	You may wish to compare `/etc/services` with `/etc/services.pre_OOB` (which is installed with the Easysoft ODBC-ODBC Bridge) for details of entries that require removal.

7. You must remove all services that were configured for use with the Easysoft ODBC-ODBC Bridge. Delete all lines pertaining to all OOB Servers and save the file.

Removing from inetd

8. Make a backup of the `/etc/inetd.conf` configuration file.
9. Open `/etc/inetd.conf` in your editor. Look for a line in the file similar to the following:

```
 esoobserver stream tcp nowait root /bin/sh
/bin/sh /usr/local/easysoft/oob/server/SERVER
```

where `esoobserver` is the name as specified in the `services` file, so there should be one entry in `inetd.conf` for every entry in the `services` file.

NB	You may wish to compare `/etc/inetd.conf` with `/etc/inetd.conf.pre_OOB` (which is installed with the Easysoft ODBC-ODBC Bridge) for details of entries that require removal.

10. Remove the lines pertaining to all OOB Servers and save the file.
11. Use `ps` to find the Process ID (PID) of the `inetd` process and send it a Hangup signal:

`kill -HUP` *`pid`*

REMOVING FROM XINETD

`xinetd` uses a configuration file of `/etc/xinetd.conf` (by default), which can either contain the names of the services and what to do with them, OR (on some Red Hat machines, for example) an `includedir` setting which points to a directory where those services are defined (one per file).

You should have been made aware of which method your machine uses from the original installation procedure (see **"Installing the OOB Server" on page 70**).

In the former case the Easysoft ODBC-ODBC Bridge install adds a service entry to `/etc/xinetd.conf` and in the latter case it creates a new service file called `esoobserver` containing the Easysoft ODBC-ODBC Bridge service settings.

12. There are therefore two ways to delete the Easysoft ODBC-ODBC Bridge service from `xinetd`.

- make a backup of the `/etc/xinetd.conf` configuration file.
- Open `/etc/xinetd.conf` in your editor. Look for a line in the file similar to the following:

  ```
  service esoobserver
  ```

  where `esoobserver` is the name as specified in the `services` file, so there should be one entry in `xinetd.conf` for every entry in the `services` file.

– OR –

- Delete the `esoobserver` file in the directory to which the `includedir` setting in `/etc/xinetd.conf` points.
- Remove all the lines referring to the OOB Server and save the file.

13. Use `ps` to find the Process ID (PID) of the `xinetd` process and send it a Hangup signal:

    ```
    kill -USR2 pid
    ```

    – OR –

    ```
    kill -USR1 pid
    ```

Removing from the dynamic linker

Notify the dynamic linker that the shared objects are no longer available.

NB	This information only applies to systems with the `ld.so` dynamic linker (normally only Linux).

14. If you have the file `/etc/ld.so.conf` file, make a backup copy, e.g.

    ```
    cp /etc/ld.so.conf /etc/ld.so.conf.safe
    ```

15. Open `/etc/ld.so.conf` and manually remove the path to the Easysoft ODBC-ODBC Bridge client shared objects. The line is of the form:

    ```
    <InstallDir>/easysoft/oob/client
    ```

16. If you are not using any other Easysoft software then you may remove the path to the common Easysoft shared objects:

    ```
    <InstallDir>/easysoft/lib
    ```

17. If you are no longer using unixODBC then you can also remove the reference:

    ```
    <InstallDir>/easysoft/unixODBC
    ```

18. Run `/sbin/ldconfig` so that the dynamic linker re-reads the file and will no longer search the removed paths.

Deleting the Software

Finally, remove the software from your system's hard drive.

19. Change directory to:

```
<InstallDir>/easysoft/
```

```
pwd
```

The system displays the current directory. Double-check that this is the directory under which you installed the Easysoft ODBC-ODBC Bridge.

Caution! Be very careful issuing the `rm -r` command as `root`. Normally `rmdir` will not remove directories that contain files, but `rm -r` will remove all subdirectories along with their contents. It is possible to effectively destroy your system and/or lose all user files by removing the wrong directory.

20. Remove the Easysoft ODBC-ODBC Bridge installation directory:

```
ls
```

Check that you are in the right directory.

```
rm -r oob
```

The system may ask you to confirm deletion for some files. You can confirm these as long as you are sure you are in the correct directory.

```
ls
```

21. If you have no other Easysoft products on your system and you are not using any copy of unixODBC that may be in this directory, then you can delete the `easysoft` directory too.

    ```
    cd ..
    ```

    ```
    rm -r easysoft
    ```

    – OR –

    If there are other files in the directory tree (i.e. you have other Easysoft products installed) then you must not remove the `easysoft` directory, because it will contain your license keys and other important files.

22. If you left the the Easysoft ODBC-ODBC Bridge distribution file on your system then you may wish to remove it at this point.

    The uninstall process is complete.

    Any licenses you obtain for the Easysoft ODBC-ODBC Bridge and other Easysoft products are stored in the `<`*`InstallDir`*`>/easysoft/license/licenses` file.

    After uninstalling the Easysoft ODBC-ODBC Bridge, unless you have deleted this file, you will not need to relicense the product when you reinstall or upgrade.

    However, for security purposes you may want to make a copy of `<`*`InstallDir`*`>/easysoft/license/licenses` before uninstalling.

CHAPTER 3 CONNECTION

Connecting via the Easysoft ODBC-ODBC Bridge

An ODBC-compliant application is connected to a remote data source via the Easysoft ODBC-ODBC Bridge (OOB) as follows:

1. On the server, create a system data source for the target database.
2. On the client, configure a data source pointing to the server machine and the server data source created in **step 1 on page 85**.
3. From your application on the client machine, connect to the OOB Client data source created in **step 2 on page 85**.

If you have an internet connection you can omit the server sections and set up a client connection to the `demo.easysoft.com` server by using the recommended settings in the relevant client subsection.

Chapter Guide

- **The connection process**
- **Setting up the OOB Server**
- **Windows server setup**
- **Unix server setup**
- **Testing the OOB Server**
- **Setting up the OOB Client**
- **Windows client setup**
- **Unix client setup**

The connection process

This section explains what happens when an ODBC application connects to a data source, and what happens when connecting through the Easysoft ODBC-ODBC Bridge.

Understanding the connection process will give an insight into why the connection might fail and what is required to develop a bespoke ODBC application to connect to remote data sources.

To skip this explanatory section, proceed to **"Setting up the OOB Server" on page 96**.

In ODBC an application connects to a database by means of a data source description, which depends on the ODBC driver used to access the database and consists of a set of attribute and value pairs.

Usually, the application links with a driver manager that looks at the data source description in the connection string, loads in the required ODBC driver and then passes the connection string to the ODBC driver.

At its simplest the application passes a connection string which defines a data source name (DSN) to the ODBC driver (or driver manager), such as:

```
DSN=test_datasource;
```

In this case the driver manager looks at the **Driver** attribute in the data source to decide which driver to use, loads the driver and then the driver looks up the data source to retrieve all the other required attributes.

This information is found in the registry under Windows and in the user and system `odbc.ini` files under Unix.

It therefore follows that before an application can connect to an ODBC driver you have to create a data source containing all the attributes the driver requires to describe the database (or alternatively, the application can pass all the attributes in the connection string).

Two data sources are required for the Easysoft ODBC-ODBC Bridge:

- a server data source on the machine where the data is located, describing the database in whatever terms the ODBC driver requires (e.g. in Windows a dialog box is provided by the ODBC driver and accessed from the ODBC Administrator).
- a client data source on the machine where the application is running, which is an Easysoft ODBC-ODBC Bridge data source describing how to connect to the OOB Server (e.g. the network transport to use, the address of the server), a user identity with which to access the server machine and a description of the server data source.

For example (omitting the driver manager differences and using TCPIP as the network transport):

Assume there is a Microsoft Access database and a Microsoft Access ODBC driver on a Windows NT machine called "`ntbox.easysoft.com`".

There is also an application on a Linux box called "`linuxbox.easysoft.com`", which wants to read data from the Microsoft Access database.

It would be necessary to install:

- the OOB Server on `ntbox`, and then create a system data source using the Microsoft Access ODBC driver to make the Microsoft Access database available to the OOB Server.
- the OOB Client on `linuxbox`, and then create a data source using the Easysoft ODBC-ODBC Bridge driver, so that the application on this machine can access the database on `ntbox`.

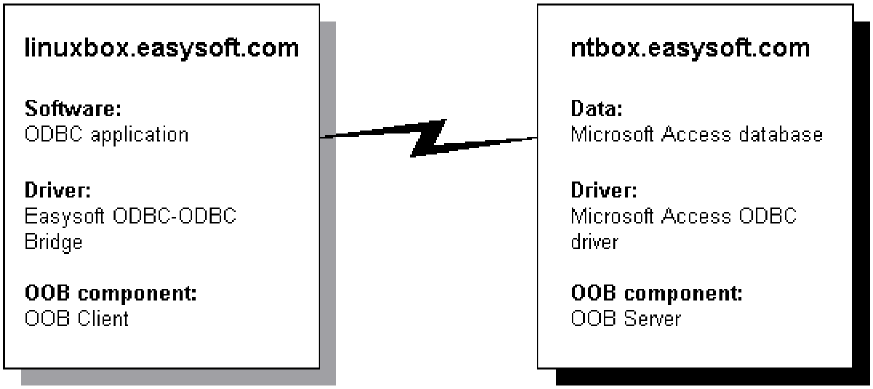


Figure 28: Linking a Linux client to an NT server

On the `ntbox` server machine the data source description for the Microsoft Access database might consist of:

```
[MyDB]

Description = my Access database

mdb file = c:\databases\msaccess\mydb.mdb
```

`MyDB` is the data source name and Microsoft Access needs the location of `mdb file` to define the database.

On the linuxbox client machine the Easysoft ODBC-ODBC Bridge data source might consist of:

```
[ntbox_mydb]

Description = Access mydb on ntbox

Server = ntbox.easysoft.com

Transport = TCPIP

Port = 8888

LogonUser = my_ntusername

LogonAuth = my_ntpassword

TargetDSN = mydb
```

- ntbox_mydb is the data source name which you can refer to in the connection string passed to the OOB Client.
- Server is the name of the server machine where the OOB Server is running.
- Port is the port where the OOB Server is listening for connections.
- LogonUser and LogonAuth describe the user details under which the OOB Client will log on to the server.
- TargetDSN describes the data source on the server machine to which the OOB Server should connect.

The application on linuxbox can now call SQLDriverConnect or SQLConnect passing a connection string of the form "DSN=ntbox_mydb;".

The OOB Client looks up the ntbox_mydb data source and retrieves all the other necessary attributes to connect to the server.

The OOB Client connects to the OOB Server, passing `my_ntusername` and `my_ntpassword` which are verified and then the OOB Server becomes the user `my_ntusername`.

At this stage the OOB Client produces a new connection string of the form "`DSN=mydb;`" in a call to `SQLDriverConnect` on the OOB Server and the process starts again in the server ODBC driver.

Note that the application is capable of passing any or all the attributes as part of the connection string (for example, the user name and password attributes can be included to avoid storing them on the server machine.

The full connection string for this example would be:

```
Server=ntbox;

Transport=tcpip;

Port=8888;

LogonUser=my_ntusername;

LogonAuth=my_ntpassword;

TargetDSN=MyDB;
```

This is a simplified example of the connection process, but it does illustrate what actually happens.

When the DBMS requires authentication

When the server database requires authentication the ODBC defined `UID` and `PWD` attributes may be included in the connection string.

Alternatively, if the `TargetUser` and `TargetAuth` attributes are added to the Easysoft ODBC-ODBC Bridge client data source, the OOB Client will pass these to the database as `UID` and `PWD`.

The OOB Client uses some simple rules to determine what to pass to the DBMS:

1. If `UID` and/or `PWD` exist in the connection string and `UseOOBDBAuth` is *not* enabled (i.e. is not selected on the Easysoft ODBC-ODBC Bridge DSN dialog box in Windows or is set to 0 for that data source in `odbc.ini` on Unix platforms), then `UID` and/or `PWD` are passed to the DBMS unchanged. If `UseOOBDBAuth` is enabled, then any `UID` and/or `PWD` values in the connection string are ignored.
2. If not Rule 1, but the DSN contains `TargetUser` and/or `TargetAuth`, these are changed to `UID` and `PWD` and passed to the DBMS.
3. If not Rule 1 or 2, no `UID` or `PWD` will be in the connection string passed to the DBMS (implying that authentication is not required).

This has implications for some applications which require you to specify a DBMS user name and password, such as the isql utility supplied with unixODBC.

Although the DBMS user name and password are optional command line arguments to isql, if you omit them isql passes empty `UID` and `PWD` attributes (this is partly because it uses `SQLConnect` instead of `SQLDriverConnect`).

As a result, if they are omitted from the command line, but the DBMS requires authentication, it will receive `"UID=;PWD=;"` (see Rule 1) and you will fail to connect.

In this case if the data source contains `TargetUser/TargetAuth`, the Easysoft ODBC-ODBC Bridge ignores them because the application specifies `UID` and `PWD` (Rule 1).

When connecting to a data source using isql, you should type:

```
./isql data_source_name username userpassword
```

e.g.

```
./isql pubs demo easysoft
```

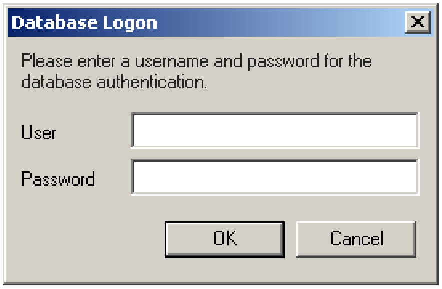
NB If you have problems connecting using isql, then use the `./isql -v` option to obtain ODBC diagnostic messages.

On platforms where the OOB Client has specific support for a GUI environment (currently only Windows) the ODBC driver may prompt you for database user name and password details.

At first the OOB Client ODBC driver will connect to the server and attempt a database connection, as described in **"The connection process" on page 86**.

However, if the ODBC driver at the server end denies authentication and returns the ODBC state `28000`, the OOB Client ODBC driver displays the **Database Logon** dialog box.

This requires a valid database user name and password to be entered, which will prompt the OOB Client ODBC driver to make one further connection attempt with this new authentication:



Please enter a username and password for the database authentication.

Figure 29: The Easysoft ODBC-ODBC Bridge Database Logon dialog box

A **Server Logon** dialog box will be displayed if you have failed to enter values into the **User** and **Password** fields of the OOB Client DSN.

In this case a valid user name and password for the server operating system must be entered:

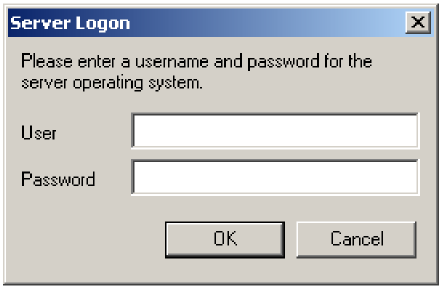

Server Logon
Please enter a username and password for the server operating system.
User
Password
OK
Cancel

Figure 30: The Easysoft ODBC-ODBC Bridge Server Logon dialog box

How ODBC driver managers fit into the connection process

This section explains how ODBC driver managers fit into the connection process when connecting via the Easysoft ODBC-ODBC Bridge.

You should read this section if you intend to integrate the Easysoft ODBC-ODBC Bridge with your own ODBC application.

In general, ODBC applications must be linked with either an ODBC driver or a driver manager.

When a program calls `SQLDriverConnect` (or `SQLConnect`), it passes in a connection string, which contains a list of connection attributes, normally including one of the following:

- a `DSN=` attribute which names the data source. Specifying a DSN (Data Source Name) implicitly tells the driver manager which driver to load because each DSN will contain a `DRIVER` attribute.
- a `DRIVER=` attribute which names the driver. Specifying a driver allows you to choose any database to which the driver has access.
- a `FILE=` attribute which allows the database attributes to be read from a file.

A connection string would look something like:

```
SQLDriverConnect("DSN=pubs;UID=demo;PWD=easysoft;")
```

where `pubs` is the data source name, `demo` is the user name with which to connect to the database, and `easysoft` is the password for the `demo` user.

The driver manager examines the connection attributes and loads the required driver (the driver is either named in the `DRIVER=` attribute or is looked up in a database of DSNs). From then on the driver manager relays ODBC calls to the driver and passes the result back to its caller.

In Windows, a configured Easysoft ODBC-ODBC Bridge driver (on the client side) appears as an ordinary ODBC driver connected through the driver manager like any other.

On non-Windows systems (which do not necessarily have a driver manager installed) the Easysoft ODBC-ODBC Bridge can be configured as a client either with a driver manager or standalone.

At the server end, the OOB Server acts like an ordinary ODBC-compliant application:

- In Windows NT, it takes the form of a network service, configured to start automatically.
- Under Unix, it is called by `inetd` or run as standalone, and is linked to the driver manager of your choice.

When your client application runs on Unix, you can choose whether to connect your application direct to the OOB Client driver or to a data source set up under a driver manager:

- if the OOB Client driver will be the only ODBC driver used on that machine you do not need a driver manager and you can link your application directly with the Easysoft ODBC-ODBC Bridge (e.g. using perl CGI scripts on a Linux web server to access a Windows ODBC driver across the Easysoft ODBC-ODBC Bridge).
- if you already have a driver manager or would like to use more than one ODBC driver on the Unix system it is necessary to create and set up a data source for the Easysoft ODBC-ODBC Bridge in the driver manager. How you do this depends on the driver manager.

The Windows platform has a well-established driver manager (`odbc32.dll`) to which programmers link their code

On Unix, many installations do not have a driver manager, so Easysoft distribute the unixODBC driver manager and recommend that you use it with the Easysoft ODBC-ODBC Bridge on Unix platforms (see **"unixODBC" on page 211**).

REF

The unixODBC driver manager is an open source project sponsored by Easysoft, rather than a commercial Easysoft product, and is fast becoming a standard across the data access community.

Other driver managers are available, but Easysoft believe unixODBC is demonstrably the most flexible and reliable open source solution.

unixODBC driver manager distributions can be found at **http://www.unixodbc.org**.

Setting up the OOB Server

In terms of the Easysoft ODBC-ODBC Bridge, the server is the machine where the ODBC driver for your database is located. The database itself may also be on this machine, although it can be located elsewhere.

To allow remote machines to access your database, you first need to create a data source on the server machine to make the database available to the OOB Server.

Before setting up a data source on your server machine, you must have successfully installed and licensed the OOB Server on this machine (see **"Installation" on page 29**).

Once the OOB Server is successfully installed, go to the section appropriate to your server platform:

- **"Windows server setup" on page 97**
- **"Unix server setup" on page 105**

Windows server setup

The OOB Server for Windows can connect to any system data source configured on your machine, given the necessary information.

When creating the data source on your server, you should use the ODBC driver suitable for your database (for example, to connect to a SQL Server database, use the SQL Server ODBC driver to create the data source).

The instructions in this section describe how to create a data source for the Microsoft Northwind database, which is shipped with Microsoft Access.

You should follow the same procedure to connect to your own database on your server machine.

To follow this example, you should have on your computer:

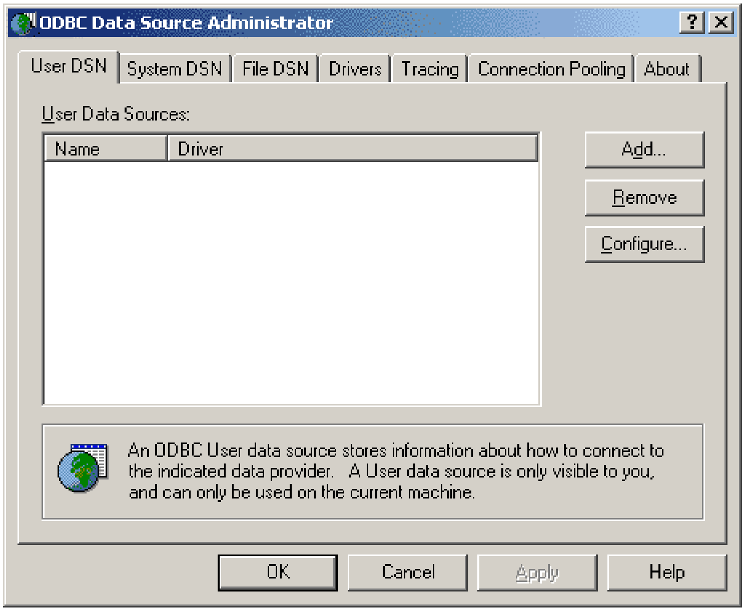
- Microsoft Access
- the ODBC driver for Microsoft Access (almost all Microsoft Access installations have this)
- a Microsoft Access database to connect to, such as `northwind.mdb`.

The first step is to open the Microsoft ODBC Data Source Administrator:

1. Select **Start > Settings > Control Panel**, double-click **Administrative Tools** and then **Data Sources (ODBC)**.

NT	Select **Start > Settings > Control Panel** and double-click **Data Sources (ODBC)**.

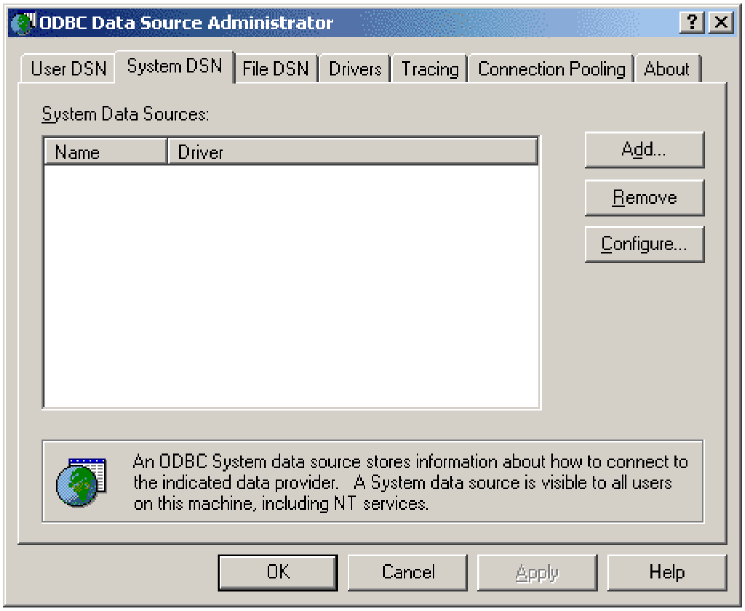
The **ODBC Data Source Administrator** dialog box is displayed.



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

Figure 31: The ODBC Data Source Administrator User DSN tab

2. Select the **System DSN** tab:



An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

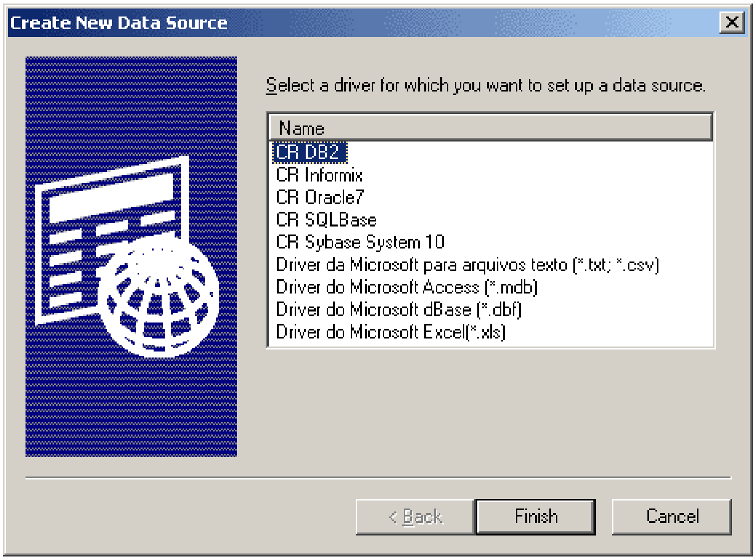
Figure 32: The ODBC Data Source Administrator System DSN tab

It is important to create a system DSN rather than a user DSN, which is only visible to the desktop user who created it.

Since the OOB Server runs as a service, user DSNs are not available to it.

3. Click **Add...** to add a new data source.

The **Create New Data Source** dialog box displays a list of drivers:



Create New Data Source

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

Name
CR DB2
CR Informix
CR Oracle7
CR SQLBase
CR Sybase System 10
Driver da Microsoft para arquivos texto (*.txt; *.csv)
Driver do Microsoft Access (*.mdb)
Driver do Microsoft dBase (*.dbf)
Driver do Microsoft Excel(*.xls)

Figure 33: The Create New Data Source dialog box

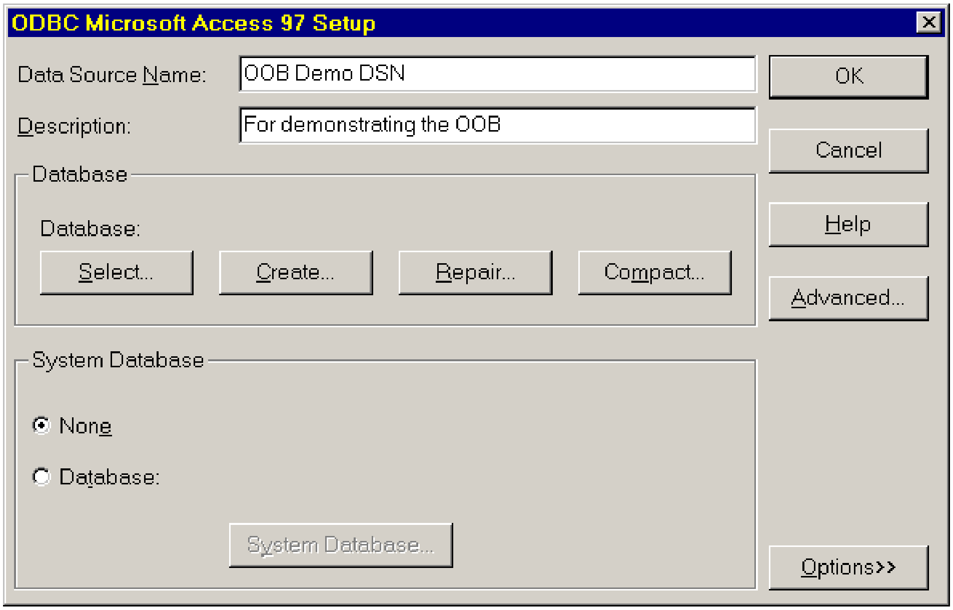
4. Select `Microsoft Access Driver` and click **Finish**.

   The ODBC driver for Microsoft Access displays a dialog box for configuring the data source (this dialog box and the attributes you need to specify vary depending on the ODBC driver you are using).

NB	The Microsoft Access ODBC driver is NOT thread-safe unless run with Jet version 4. Previous versions require the OOB Server to be configured to run in multi-process mode, rather than the default multi-threaded mode (see the **Easysoft ODBC-ODBC Bridge FAQ**).

5. Enter your chosen name for this data source in the **Data Source Name** box (e.g. `"OOB Demo DSN"`).

6. In the **Description** field, enter something that would help a user faced with a choice of data sources (e.g. "`For demonstrating the OOB`"):



ODBC Microsoft Access 97 Setup

Data Source Name: OOB Demo DSN

Description: For demonstrating the OOB

Database

Database:

System Database

● None

○ Database:

Figure 34: The ODBC Microsoft Access Setup dialog box

7. Click **Select...** to browse for the target database, select a database and click **OK**.

   If the `Microsoft Office\Office\Samples\Northwind.mdb` example database is not accesible, use any available database (preferably a small one).

NB	Note the data source name because it will need to be specified when a data source is created on the client machine.

8. Click **OK** to return to the ODBC Data Source Administrator window.

Note:

- the window now contains a line containing the new data source
- the System DSN tab should be selected (if it is not, then remove the new data source, select the System DSN tab, and return to **step 3 on page 99**).

9. Click **OK**.

   You have now set up a system data source on your machine to a local database, making it visible to the OOB Server.

Starting the OOB Server in Windows

Before an OOB Client can connect to a data source on the server machine, the OOB Server must be running. In Windows NT and Windows 2000 the installation program configures the OOB Server to start automatically as an NT Service. For Windows 9x, the OOB Server must be run manually.

95/98	Choose **Start > Programs > Easysoft > ODBC-ODBC Bridge > Server**.

Checking the OOB Server Service under Windows NT

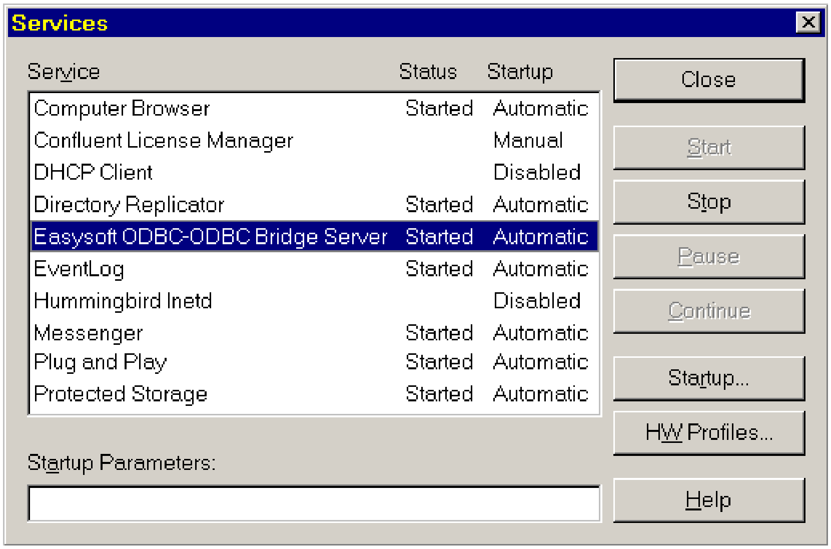
This procedure is not normally necessary, but should be followed if you are having problems connecting with the Easysoft ODBC-ODBC Bridge.

1. Choose the `Services` icon from **Start > Settings > Control Panel**.

2000/ XP	Choose **Start > Settings > Control Panel** and open the `Administrative Tools` icon followed by the `Services` icon.

A list of your NT system's registered services is displayed.

2000/ XP	The `Services` dialog box looks different in Windows 2000 and Windows XP, but the principles and the functionality are the same.



Service	Status	Startup
Computer Browser	Started	Automatic
Confluent License Manager		Manual
DHCP Client		Disabled
Directory Replicator	Started	Automatic
Easysoft ODBC-ODBC Bridge Server	Started	Automatic
EventLog	Started	Automatic
Hummingbird Inetd		Disabled
Messenger	Started	Automatic
Plug and Play	Started	Automatic
Protected Storage	Started	Automatic

Startup Parameters:

Figure 35: The Easysoft ODBC-ODBC Bridge Server Services entry

2. Find the entry for `Easysoft ODBC-ODBC Bridge Server.`
3. If the `Startup` field says `Disabled` then click **Startup...**.
4. In the resulting dialog box, select **Automatic and** then click **OK**.
5. If the `Status` field does NOT say `Started` then click **Start** to bring the Server on-line.
6. Click **Close**.
7. Close the **Control Panel**.

Testing the Data Source

You now have the OOB Server running on your Windows machine and a data source connecting to the database on the server.

Before setting up the OOB client, test that this data source is working, so that you can verify that the server side is functioning correctly, by running any other ODBC application on your Windows machine, linking to this data source and accessing its data.

Refer to the documentation supplied with your ODBC application if you are unsure how to link to a data source.

NB

If the database on your server machine is a Microsoft Access database then you cannot test the data source by linking to it from the Microsoft Access application.

You must connect to it via another ODBC application on your Windows machine, because although Microsoft Access is a multi-threaded application, the Microsoft Access ODBC driver is NOT thread-safe.

When using the Easysoft ODBC-ODBC Bridge and the Microsoft Access ODBC driver together you must ensure that the OOB Server is running in multi-process mode (see **"MultiProcess" on page 171**).

Once you have established that the data source on your server machine is accessing data correctly, you can proceed to **"Testing the OOB Server" on page 111**.

Unix server setup

The OOB Server for Unix can connect to any system data source configured on the Unix machine, given the necessary information.

Easysoft recommend that you use the unixODBC driver manager, supplied with the Easysoft ODBC-ODBC Bridge, for setting up data sources on Unix.

NB	This section explains how to set up data sources using unixODBC as installed by the Easysoft ODBC-ODBC Bridge. If you choose to use a different driver manager, you should refer to the documentation with that driver manager for details of setting up data sources on Unix. For further information about using unixODBC, see **http://www.unixodbc.org**.

With unixODBC, you can create a data source by doing either of the following:

- directly adding the data source and its attributes into a configuration file (`odbc.ini`)
- if you are running an X server you can create a data source using the graphical ODBC Data Source Administrator (`ODBCConfig`).

Creating a DSN by editing a configuration file

With unixODBC, data sources are stored in a configuration file called `odbc.ini`.

If you accepted the default Easysoft ODBC-ODBC Bridge installation, system data sources will be stored in `/etc/odbc.ini.`

However, if you have built unixODBC yourself, then it will be whatever path you specified in the `sysconfdir=`*directory* configure option.

If `sysconfdir` has not been specified then the path will default to `/usr/local/etc`).

NB	By default, you must be logged in as `root` to edit a system data source defined in `/etc/odbc.ini`, but user data sources created in an `.odbc.ini` file in a home directory are visible to an individual user only (i.e. the `Logon User` passed from the client to the server).

Each section starts with a data source name in square brackets [ ], followed by a number of *attribute=value* pairs.

The attributes that you need to specify vary depending on which ODBC driver you are using to connect to the local database.

A sample data source using the PostgreSQL driver is of the format:

```
[MAIN]

Description = Main data on Admin box

Driver = PostgreSQL

Database = main

Servername = localhost

UserName =

Password =

Port = 5432

Protocol = 6.4

ReadOnly = No
```

```
RowVersioning = No

ShowSystemTables = No

ShowOidColumn = No

FakeOidIndex = No

ConnSettings =
```

NB unixODBC uses the `Driver` attribute to look up the driver in the `odbcinst.ini` file and locate the shared object to use as the ODBC driver.

Refer to the documentation with your ODBC driver for full details of the attributes required to define a data source.

Creating a DSN using the ODBC Data Source Administrator

To create a data source using the graphical ODBC Data Source Administrator supplied with unixODBC:

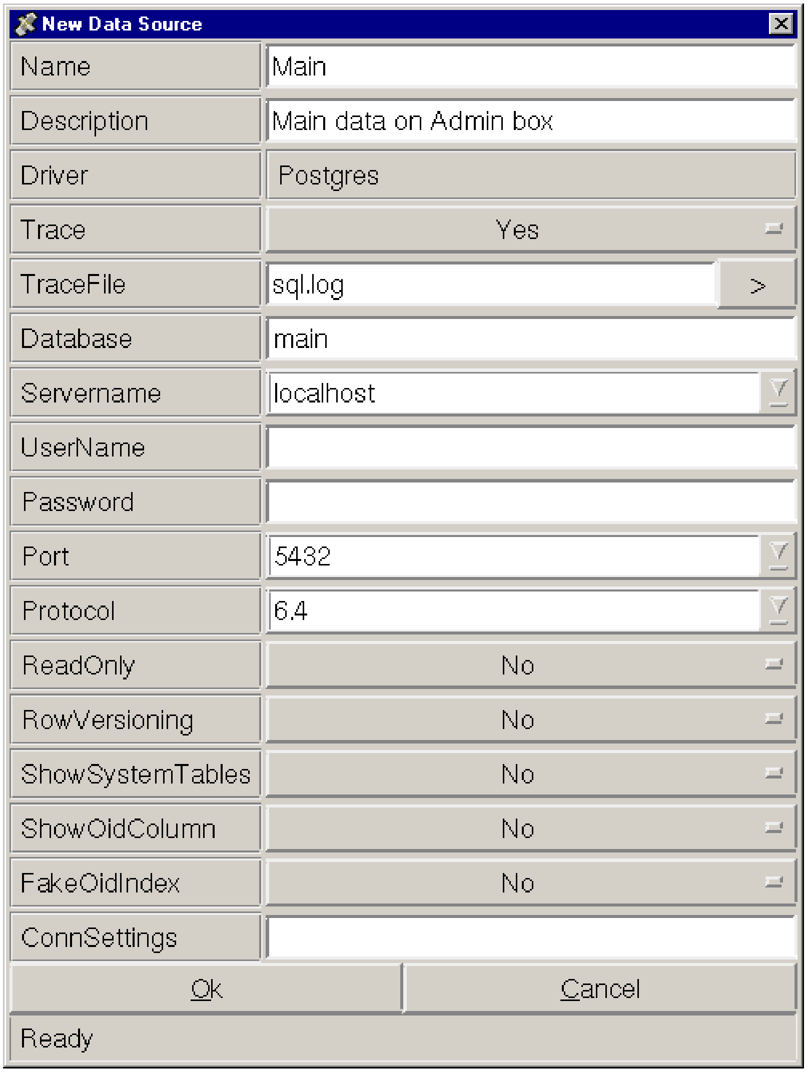
1. Run an X session connecting to your Unix machine, ensuring that you log in as `root`.
2. Change into the `<`*`InstallDir`*`>easysoft/unixODBC/bin` directory.
3. Type `./ODBCConfig` *<Enter>*.

   The ODBC Data Source Administrator opens.
4. Click the **System DSN** tab to create a data source which is available to any user or service that logs into this machine.
5. Click **Add** to create a new data source.

   The **Adding a New Data Source** dialog box displays the available drivers.

6. Select the driver required to connect to the database and click **OK**.

   A configuration dialog box specific to that driver is now displayed, such as the PostgreSQL data source:



New Data Source

Name	Main
Description	Main data on Admin box
Driver	Postgres
Trace	Yes
TraceFile	sql.log
Database	main
Servername	localhost
UserName	
Password	
Port	5432
Protocol	6.4
ReadOnly	No
RowVersioning	No
ShowSystemTables	No
ShowOidColumn	No
FakeOidIndex	No
ConnSettings	

Ready

Figure 36: The Configuration dialog box for a PostgreSQL data source

Refer to the documentation with your ODBC driver for full details of the attributes you need to specify on this dialog box.

7. Click **OK** when you have specified the data source attributes you require and then close the ODBC Data Source Administrator.

Testing the Data Source

Before setting up the OOB Client, the new data source and the OOB Server setup should be verified with either an ODBC application available on your Unix machine or the unixODBC `isql` utility.

To use `isql` to test the data source:

1. Change into the `<`*`InstallDir`*`>easysoft/unixODBC/bin` directory.

2. Type:

```
./isql data_source_name
```

For example, to connect to the PostgreSQL data source illustrated earlier in this section, you would type:

```
./isql main
```

NB	If there is a problem connecting using `isql`, then use the `./isql -v` option to obtain ODBC diagnostic messages.

If your server DBMS requires authentication, you should include the DBMS user name and password arguments in the `isql` command (see **"When the DBMS requires authentication" on page 90**).

3. Once connected, either:

    - type an SQL statement to query the data, such as:

      `select * from` *`tablename`*

      where *`tablename`* is a table in that database

    – OR –

    - type `help` to get a list of tables in the database.

4. To leave `isql` and return to the system prompt, press *<Enter>*.

   You can now proceed to **"Testing the OOB Server" on page 111**.

Testing the OOB Server

OOBPING

If you believe that there are problems with the connection between the application and the server then you can use `oobping` to help diagnose and fix any errors before proceeding to **"Setting up the OOB Client" on page 125**.

`oobping` is a small program distributed with versions of the Easysoft ODBC-ODBC Bridge from 1.0.0.35 onwards, and is a valuable tool for checking Easysoft ODBC-ODBC Bridge connectivity and diagnosing connection problems or connection timing issues:

In Windows distributions the `oobping.exe` program is located in the `<installdir>\Easysoft\Easysoft ODBC-ODBC Bridge` directory.

In Unix distributions there are two versions of `oobping` located in the `/usr/local/easysoft/bin` directory.

- `oobpings` (a statically linked version)
- `oobpingd` (a dynamically linked version linked against the `libesoobclient` shared object)

NB	To use `oobpingd` you may need to set and export your `LD_LIBRARY_PATH/LD_RUN_PATH/LIBPATH` to include `/usr/local/easysoft/oob/client` and `/usr/local/easysoft/lib`.

`oobping` has the following command line:

```
oobping [-h host | -d ODBC_connection_string] {-t
port} {-u osuser} {-p ospassword} {-e}
```

where:

`-h host`

The name or IP address of the machine where the OOB Server is located.

`-d ODBC_connection_string`

An ODBC connection string consisting of a list of semi-colon separated `attribute=value` pairs, as defined by ODBC.

e.g. `DSN=test;UID=john;PWD=smith;`

If the `-u` and/or `-p` attributes are also specified as well as `-d` then "`LogonUser=xxx;LogonAuth=yyy;`" (where `xxx` and `yyy` are the values specified for `-u` and `-p`) will be added to the end of the connection string.

`-t port`

The port on which the OOB Server is listening.

`-u osuser`

A valid user name on the "host" operating system.

`-p ospassword`

A password for the user specified with the `-u` attribute.

`-e`

Show the amount of time (in seconds) to complete the requested operation (from Easysoft ODBC-ODBC Bridge version 1.1.0.0 onwards).

WORKED EXAMPLES

The following examples illustrate the ways in which `oobping` can be used to investigate connection issues:

- **Example 1: Check the OOB Server is running on the correct machine and listening on the correct port**
- **Example 2: Check OOB Server authentication**
- **Example 3: Check connectivity to a specified remote data source**
- **Example 4: Checking connection times**

Example 1: Check the OOB Server is running on the correct machine and listening on the correct port

`oobping` connects to port `8888` on the machine `myserver`, where `OOB Server version 1.1.0.00` is reported as running.

```
oobping -h myserver -t 8888


Host: myserver, Port: 8888
Attempting connection...OK
Examining Server...
    OOB Server Version: 1.1.0.00
    OOB Server Name: OOB
```

NB `oobping` defaults to port `8888`, so the `-t` attribute can be omitted.

If the wrong port is specified or some other service is listening on the specified OOB Server port, then you will get a variety of errors, such as when pointing `oobping` at an SMTP server on port 25:

```
    oobping -h myserver -t 25
    Host: myserver, Port: 25

    Failed to receive data
    Packet (size=842149920) too big for buffer
(size=256)
```

If there is nothing listening on the specified port then you will get a connection refused error and you should check that the OOB Server is running and is configured to use the specified port.

e.g.

```
    oobping -h myserver -t 8889
    Host: myserver, Port: 8889

    Connection refused, connect(), after 5 attempts
```

If you have specified access control rules in the OOB Server then you might see an error such as this:

```
    oobping -h myserver -t 8888
    Host: myserver, Port: 8888

    Client denied access due to access control
rule.
```

Here the machine you are running `oobping` on has been denied access to the OOB Server and you should check the access control rules on the security page of the OOB Web Administrator.

Example 2: Check OOB Server authentication

Once you are sure a connection can be made to the OOB Server with `oobping` (as in example 1) you can check OOB Server authentication.

The `-u` and `-p` arguments to `oobping` allow you to specify a valid operating system user name and password.

> **NB**
>
> If you have disabled authentication in the OOB Server then any user name and password should work.
>
> If the user names or passwords you are using contain spaces or characters that the shell might interpret you should enclose them in single quotes (Unix) or double quotes (Windows).
>
> The values specified with `-u` and `-p` are equivalent to the OOB Client DSN attributes `LogonUser` and `LogonAuth`.

e.g.

```
    oobpings -h myserver -t 8888 -u 'A User' -p
'mypassword'

    Host: myserver, Port: 8888

    Attempting connection...OK

    Examining Server...

        OOB Server Version: 1.1.0.00

        OOB Server Name: OOB

    Trying to authenticate...OK
```

If there is something wrong with the user name or password then the output will look something like this:

```
    oobpings -h myserver -t 8888 -u 'A User' -p
'mypassword'

    Host: myserver, Port: 8888

    Attempting connection...OK

    Examining Server...

        OOB Server Version: 1.1.0.00

        OOB Server Name: OOB

    Trying to authenticate...Fail

    Authentication failure (error number 1326)
```

In this case the error returned by the remote user name/password authentication service is 1326 (as the server was on Windows, this is a Windows error code).

All the common Windows error codes can be found in the **Easysoft ODBC-ODBC Bridge FAQ**.

Example 3: Check connectivity to a specified remote data source

Once you have completed examples 1 and 2 you should have:

- the name of a server where an OOB Server is running
- the port it is listening on
- a valid operating system user name and password
- be enabled in the OOB Server access control rules

You can now check connectivity to a specified remote data source (DSN), which can be done in two ways with `oobping`.

The simplest way is to add a local DSN to your `odbc.ini` file and then specify the DSN in the `-d` argument as "`DSN=dsnname`". This method tests you have defined the local OOB Client DSN correctly and put the definition in the right file.

An alternative method is to specify all the connection attributes in the `-d` argument, not just the DSN.

e.g.

Suppose as a result of examples 1 and 2 you have the following information:

```
Server = myserver

Port = 8888

LogonUser = me

LogonAuth = mypassword
```

Now enter the name of the target DSN on `myserver` in the `TargetDSN` attribute.

NB	Note that the target DSN **MUST** be a remote SYSTEM data source, as you cannot access USER data sources.

If the database also needs login information then the database user name and password are specified using the `TargetUser` and `TargetAuth` attributes.

If you want to test a DSN in your `odbc.ini` file then it would look something like:

```
[mydsn]

Server = myserver

Port = 8888

LogonUser = me
```

```
LogonAuth = mypassword

TargetDSN = mysystemdsn

TargetUser = dbusername

TargetAuth = dnpassword
```

NB "`Driver=OOB`" must be included in the DSN in order to use the **unixODBC** driver manager, but this is not needed when using `oobping`.

You can run `oobping` as follows:

```
    oobping -d "DSN=mydsn;"

    Using Connection string :

    DSN=mydsn;

    Connected OK


    01000:1:5701:[NetConn:
032bc620][Microsoft][ODBC SQL Server Driver]

    [SQL Server]Changed database context to 'pubs'.

    01000:2:5703:[NetConn:
032bc620][Microsoft][ODBC SQL Server Driver]

    [SQL Server]Changed language setting to
us_english.

    OutConnectionString:


DSN=mydsn;UID=dbusername;PWD=dbpassword;SERVER=mys
erver;
```

```
PORT=8888;TARGETDSN=mysystemdsn;LOGONUSER=me;LOGON
AUTH=mypassword;
```

```
    Connected to database:  pubs
    DBMS Name:  Microsoft SQL Server
    Driver Name:  esoobclient
    Driver Version:  01.00.0043
    Disconnecting
```

The ODBC connection string "`DSN=mydsn`" was passed via `oobping` to the OOB Client, but the OOB Client did not receive sufficient attributes in the connection string.

In order to define what it should do, the OOB Client looked up the DSN '`mydsn`' in the `odbc.ini` file, where it obtained the additional attributes `Server`, `Port`, `LogonUser`, `LogonAuth`, `TargetDSN`, `TargetUser` and `TargetAuth`.

The OOB Client then connected to the OOB Server on `myserver` via port `8888`, logged you in with `LogonUser/LogonAuth` values and finally connected via ODBC to the remote data source '`mysystemdsn`'.

NB	Although in this example the DSN pointed to Microsoft SQL Server and informational diagnostics reported the language as "`us_english`" and the database as "`pubs`", not all databases will return informational messages on the connection.

The `OutConnectionString` is a string returned by the OOB Client which you can use to connect to this data source again and the final messages show the database name, DBMS name, driver name and driver version.

Instead of specifying just the name of a DSN to `oobping` and defining the other connection attributes in the `odbc.ini` file you can specify all the connection attributes in one go and not use an `odbc.ini` file.

The `OutConnectionString` shows you what connection string you could have passed to the OOB Client to connect without a DSN (if you remove the "`DSN=mydsn`;") to produce the same result.

e.g.

```
    oobping -d
"UID=dbusername;PWD=dbpassword;SERVER=myserver;

PORT=8888;TARGETDSN=mysystemdsn;LOGONUSER=me;LOGON
AUTH=mypassword;"
```

You might be slightly confused as to why `TargetUser/TargetAuth` is specified in the `odbc.ini` file, but `UID/PWD` in the connection string.

Strictly speaking, the ODBC defined attributes are `UID/PWD` and they are passed to the database for authentication.

As far as ODBC connection strings are concerned, `UID/PWD` and `TargetUser/TargetAuth` are synonymous in the Easysoft ODBC-ODBC Bridge, but if specified in the `odbc.ini` file you should always use `TargetUser/TargetAuth`.

Now you know how to use `oobping` with connections string here are some examples of common problems you might see:

```
    oobping -d

"UID=dbuser;PWD=dbauth;TargetDSN=test;port=8888;Lo
gonUser=me;LogonAuth=mypassword"

    Using Connection string :


UID=dbuser;PWD=dbauth;TargetDSN=test;port=8888;Log
onUser=me;LogonAuth=mypassword


    IM002:1:0:[Easysoft ODBC (Client)]

    Data source not found and no default driver
specified

    HY000:2:0:[Easysoft ODBC (Client)]

    general error: Missing attribute(s): SERVER
```

The initial diagnostic, `IM002` is defined by ODBC, but as it is rather vague, the OOB Client added a secondary more helpful diagnostic.

Here the Easysoft ODBC-ODBC Bridge connection attribute '`Server`' was omitted and so the OOB Client did not know which server to connect to.

```
    oobping -d

"Server=myserver;UID=dbuser;PWD=dbauth;TargetDSN=t
est;port=8888;LogonUser=me;Log

onAuth=mypassword"

    Using Connection string :
```

```
Server=myserver;UID=dbuser;PWD=dbauth;TargetDSN=te
st;port=8888;LogonUser=me;LogonAuth=mypassword
```

```
    28000:1:18456:[][Microsoft][ODBC SQL Server
Driver][SQL Server]
    Login failed for user 'dbuser'.
```

This Microsoft SQL Server error suggests the database password '`dbauth`' is not correct for the database user '`dbuser`', and illustrates the importance of identifying which component is reporting an ODBC error.

You should examine the components displayed in `[]`, where the one furthest to the right is the reporting component and as you move left through the components you are moving closer to the OOB Client.

For example, if you specified a `TargetDSN` which did not exist on the server machine then the driver manager on the remote machine would be the component reporting the error and so the diagnostic error message (for Windows) would be:

```
    IM002:1:0:[][Microsoft][ODBC Driver Manager]
    Data source name not found and no default
driver specified
```

The other important thing to note here is that `oobping` outputs any ODBC diagnostics as:

```
    ODBC State : diagnostic sequence : ODBC native
error: text
```

The native error code is specific to the component reporting the error, and can be referenced in the Microsoft SQL Server documentation, which should tell you more accurately in which circumstances this error is reported.

Example 4: Checking connection times

From version 1.1.0.0 of the Easysoft ODBC-ODBC Bridge, `oobping` includes the `-e` option, which times the requested operation and can be invaluable in diagnosing a slow connection and determining which phase the problem is occurring in.

These operations may also use the `-e` switch.

e.g.

```
oobping -e -h myserver -t 8888

 Host: myserver, Port: 8888

 Attempting connection...OK

 Examining Server...

     OOB Server Version: 1.1.0.00

     OOB Server Name: OOB

 Time for execution: 0.16s
```

If this is repeated with the `-u` and `-p` attributes you can work out the extra time required to perform the operating system logon:

```
oobping -e -h myserver -u myosuser -p mypassword

  Host: myserver, Port: 8888

  Attempting connection...OK

  Examining Server...

      OOB Server Version: 1.1.0.00

      OOB Server Name: OOB
```

```
    Trying to authenticate...OK
    Time for execution: 0.52s
```

This example clearly demonstrates the extra time required to authenticate a user.

Setting up the OOB Client

The Easysoft ODBC-ODBC Bridge client is the machine running the ODBC application with which you want to access the data on the server.

To allow an ODBC application on the client machine to access data on the remote server, you need to create a data source on the client.

This data source uses the Easysoft ODBC-ODBC Bridge driver and specifies the attributes required to connect to the data source on the remote server.

Before setting up a data source on your client machine, you must have successfully installed the OOB Client on this machine, and set up the server side of the Easysoft ODBC-ODBC Bridge.

Instructions for installing the OOB Client on Windows and Unix platforms are provided in **"Installation" on page 29**.

If you have not already set up the server side of the Easysoft ODBC-ODBC Bridge, go to **"Setting up the OOB Server" on page 96**.

Once the OOB Client is successfully installed and your server is set up correctly, go to the section appropriate to your appropriate client platform:

- **"Windows client setup" on page 126**
- **"Unix client setup" on page 142**

Windows client setup

This section explains the steps you should take to connect an ODBC application on a Windows machine (where the OOB Client is installed) to the data source on your remote server (where the OOB Server is installed).

The instructions in this section show you how to connect to an example data source on a server at Easysoft, but you should follow the same procedure to connect to the data source on your own server.

The OOB Client is an ODBC driver. You configure a data source using the OOB Client in the same way that you configure a data source using any other ODBC driver, so many of the steps in this section are similar to those in **"Windows server setup" on page 97**.

Firstly, open the Microsoft Data Source Administrator:

1. Select **Start > Settings > Control Panel** and open the `ODBC` icon.

2000	To find the ODBC icon in Windows 2000, open `Administrative Tools` in Control Panel. The ODBC icon is called `Data Sources (ODBC)`.

   The ODBC Data Source Administrator opens.

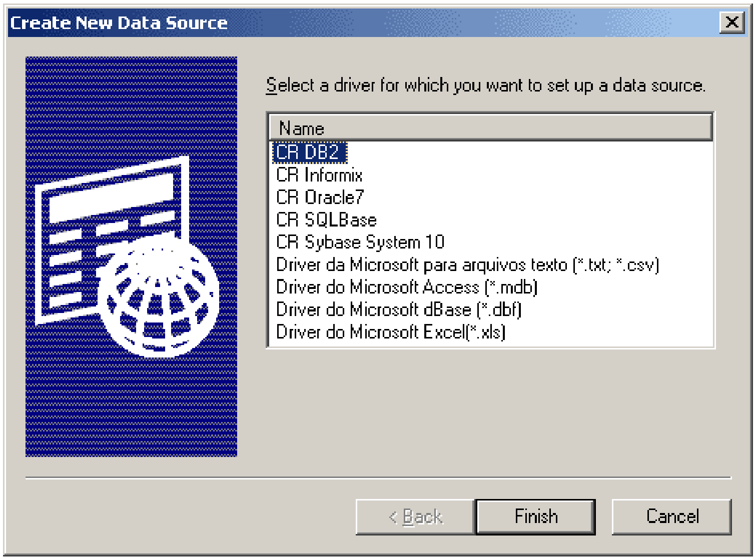
2. To create a data source that is only available to the user currently logged into this machine, select the **User DSN** tab.

   – OR –

   To create a data source that is available to any user who logs into this machine, select the **System DSN** tab.

3. Click **Add...** to add a new data source.

The **Create New Data Source** dialog box is displayed, containing a list of drivers:



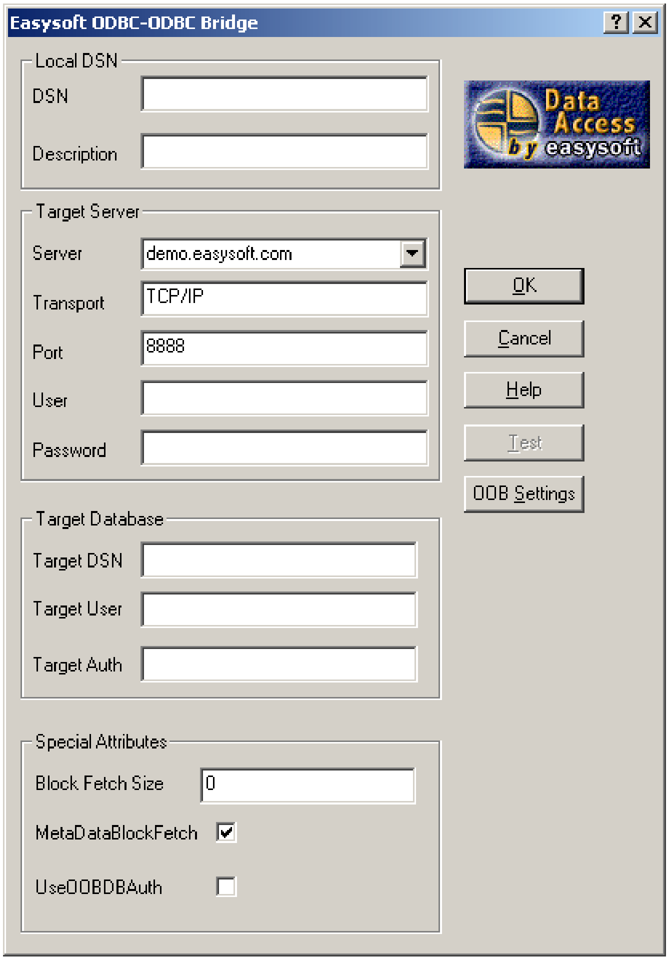
Select a driver for which you want to set up a data source.

Name
CR DB2
CR Informix
CR Oracle7
CR SQLBase
CR Sybase System 10
Driver da Microsoft para arquivos texto (*.txt; *.csv)
Driver do Microsoft Access (*.mdb)
Driver do Microsoft dBase (*.dbf)
Driver do Microsoft Excel(*.xls)

Figure 37: The Create New Data Source dialog box

4. Select `Easysoft ODBC-ODBC Bridge` and click **Finish**.

A a DSN configuration dialog box is displayed:



Server: demo.easysoft.com
Transport: TCP/IP
Port: 8888
Block Fetch Size: 0
MetaDataBlockFetch ☑
UseOOBDBAuth □

Figure 38: A blank Easysoft ODBC-ODBC Bridge DSN dialog box

The options on this dialog box are split into four sections, arranged by functionality, from top to bottom:

- how the data source appears to the driver manager and ODBC application
- how to attach to the OOB Server on the remote machine
- the data source to attach to on the remote machine
- special settings for tuning

The **Test** button allows you to check that the client is able to connect to the specified server data source.

NB	**Test** will not be enabled until an entry with which to define the data source has been made into the **Target DSN** field.

The **OOB Settings** button allows the user to specify further network options relating to the specified data source.

5. Enter a name for this data source.

   Choose carefully because you will not be able to change this after pressing **OK**.

NB	A `demo` data source will probably already exist because it is set up during the OOB Client installation. It is required if you intend to use the `demo.exe` client application.

6. In the **Description** field, enter something that would help a user faced with a choice of data sources.

Specifying the remote machine

7. In the **Server** box, type the name of the machine on which the OOB Server is running, or `demo.easysoft.com`.

   You can also select a server name that has been used previously.

8. In the **Transport** box, enter the network transport protocol to use.

   Currently, only TCP/IP is supported.

9. You should also use the default value for the **Port** box unless you know that the OOB Server is listening on another port (the port can be a number or a service name listed in `/etc/services`).

10. In the **User** and **Password** boxes, enter a valid logon account and password for the machine on which the OOB Server is running (if required).

    The OOB Server carries out all activities as this user.

NB	If your server is a Windows NT machine, you may need to include the NT domain name with the user name, using the format `domain/user name`. For example: `admin/John Smith`.

    – OR –

    If you are connecting to the Easysoft `demo` server, enter `demo` and `easysoft` respectively in these boxes.

Specifying the DSN on the remote machine

11. In **TargetDSN**, enter the data source name on your remote machine.

    – OR –

    If connecting to the Easysoft `demo` server, enter `pubs`.

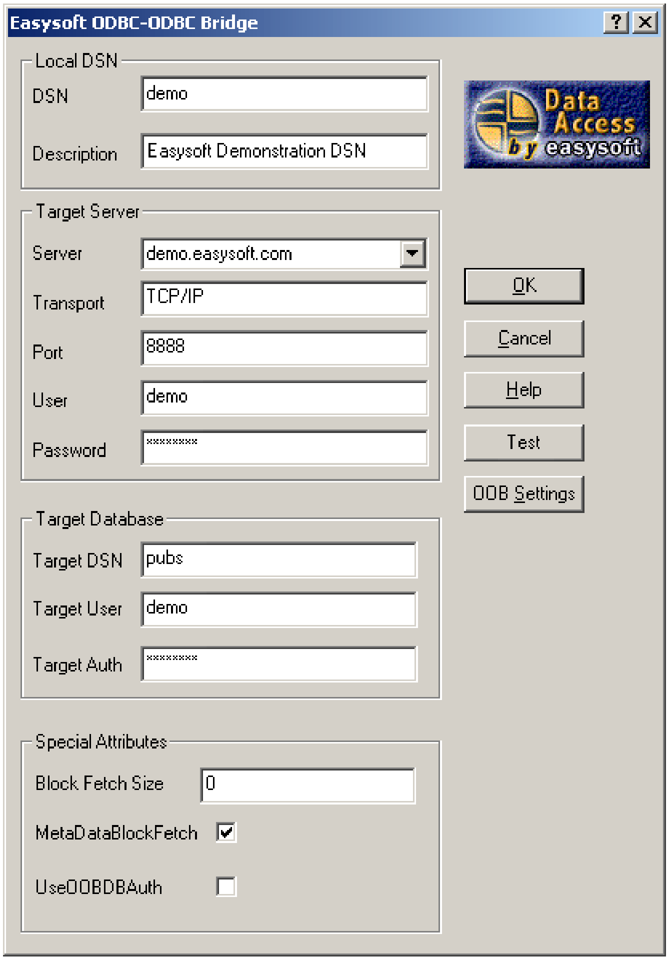
12. If your remote data source (i.e. the database itself) requires a user name and password apart from the user logon account for the machine, then enter these in **TargetUser** and **TargetAuth**.

    – OR –

    If you are connecting to the Easysoft demo data source, enter `demo` and `easysoft` in **TargetUser** and **TargetAuth**.

    – OR –

If your data source does not need separate authentication details then leave these fields blank.



Easysoft ODBC-ODBC Bridge

Local DSN

DSN: demo

Description: Easysoft Demonstration DSN

Target Server

Server: demo.easysoft.com

Transport: TCP/IP

Port: 8888

User: demo

Password: ********

Target Database

Target DSN: pubs

Target User: demo

Target Auth: ********

Special Attributes

Block Fetch Size: 0

MetaDataBlockFetch ☑

UseOOBDBAuth □

Figure 39: The DSN set up for the Easysoft demo data source

SPECIAL ATTRIBUTES

Leave the following settings at their default values if you are connecting to the Easysoft demo data source or if you are unsure about how to use them.

13. If you set **Block Fetch Size** to a value greater than 0, the Easysoft ODBC-ODBC Bridge determines whether to perform an optimization which retrieves multiple rows of data instead of one row at a time.

    This optimization is not performed if the application itself binds columns in the result set. If your ODBC application uses cursors or positioned updates/deletes, you should not set this to greater than 1. Refer to the **Easysoft ODBC-ODBC Bridge FAQ** for more information about this value.

14. Select the **MetaDataBlockFetch** option to enable blockfetching for meta data without affecting other result sets.

    This will increase the speed of retrieving metadata, such as lists of tables or columns in a data source. This option is selected (set to "`On`") by default, but needs to be disabled for a few ODBC drivers that do not support it (see the **Easysoft ODBC-ODBC Bridge FAQ** for a list of these drivers).

15. When the **UseOOBDBAuth** option is selected it causes the Easysoft ODBC-ODBC Bridge to ignore the UID and PWD values passed from the application in the connection string and use the `TargetUser` and `TargetAuth` values for UID/PWD.

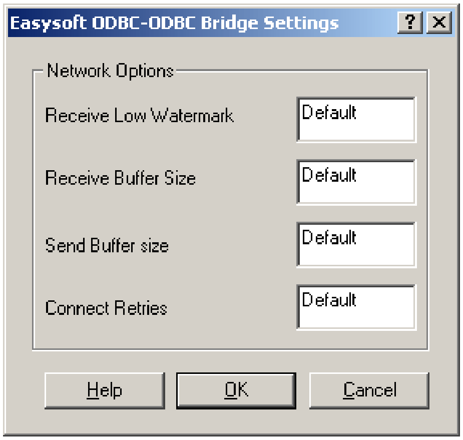
    This option may need to be selected for some software (e.g. Crystal Info which passes an empty PWD field and expects the driver to issue a prompt dialog box), but this is not always possible when the driver is on a remote machine. By default this option is not selected.

16. There is also a `DisguiseWide` attribute which allows support for wide characters.

Easysoft have not found any Windows ODBC applications with this requirement, so it can only be passed in the connection string, using `DisguiseWide=1` to hide wide character datatypes (`SQL_WCHAR` etc.) so that they are received by the application as normal character datatypes (some versions of StarOffice need this enabled when connecting to Microsoft SQLServer).

OOB Settings displays a second dialog box with further settings, which are initially set to `Default` (i.e. the internal Easysoft ODBC-ODBC Bridge default of -1). In most cases you will not need to change these settings, but if you do change a setting, enter either `Default` or `-1` in the appropriate field to revert to the default.

Note that if you do change any of these settings, the values are changed for *all* your Easysoft ODBC-ODBC Bridge data sources. You cannot specify different values per data source.



Easysoft ODBC-ODBC Bridge Settings

Network Options

Receive Low Watermark: Default

Receive Buffer Size: Default

Send Buffer size: Default

Connect Retries: Default

Figure 40: The Easysoft ODBC-ODBC Bridge Settings dialog box

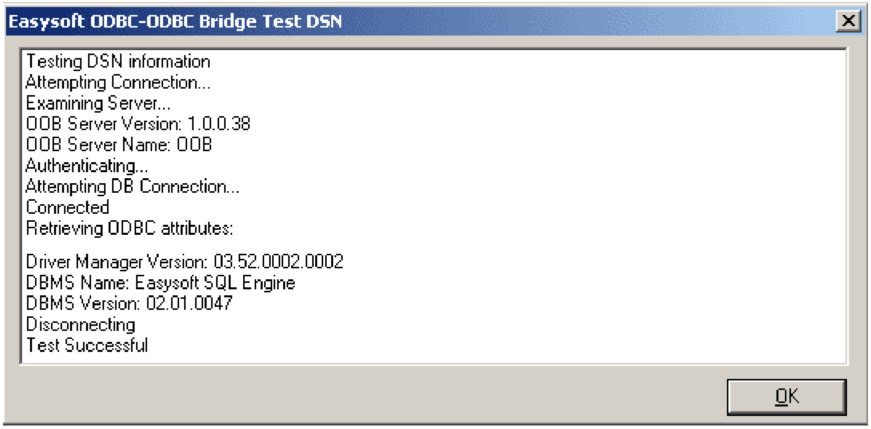
- **Receive Low Watermark** is the size of the receive low-water mark. Easysoft recommend that you do not change this setting unless advised otherwise.
- **Receive Buffer Size** is the size of the receive queue buffer. The internal Easysoft ODBC-ODBC Bridge default (-1) is 16K. To turn this setting off, set the value to 0; to increase the receive queue buffer, set it to a value greater than 0. On some platforms, increasing the receive queue buffer will speed up the transfer of data across the network though this will be at the expense of memory. Easysoft recommend that you do not change this setting unless advised otherwise.
- **Send Buffer Size** is the size of the send queue buffer. The internal Easysoft ODBC-ODBC Bridge default (-1) is 16K. To turn this setting off, set the value to 0; to increase the send queue buffer, set it to a value greater than 0. On some platforms, increasing the send queue buffer will speed up the transfer of data across the network though this will be at the expense of memory. Easysoft recommend that you do not change this setting unless advised otherwise.
- **Connect Retries** is the number of times that the OOB Client will attempt to connect to the OOB Server if the first attempt is refused by the server operating system. The internal Easysoft ODBC-ODBC Bridge default is 5, so the OOB Client will make five attempts to connect to the OOB Server before reporting a failure. If you are connecting to OOB Enterprise Servers and using the fallback server capability (i.e. a list of multiple OOB Servers in the DSN), you should set `Connect Retries` to 1 to minimise the time taken to connect to subsequent servers (see **"Client support for Fallback OOB Servers" on page 237**).

Receive Low Watermark, **Receive Buffer Size** and **Send Buffer Size** can also be set via the **Configuration** screen of the Web Administrator (see **"Configuration" on page 157**).

CHECK YOUR VALUES

17. Now click **Test**.

    Using the information on the dialog box, the OOB Client attempts to connect to a data source on the server machine, send an ODBC request and display the results:



```
Testing DSN information
Attempting Connection...
Examining Server...
OOB Server Version: 1.0.0.38
OOB Server Name: OOB
Authenticating...
Attempting DB Connection...
Connected
Retrieving ODBC attributes:

Driver Manager Version: 03.52.0002.0002
DBMS Name: Easysoft SQL Engine
DBMS Version: 02.01.0047
Disconnecting
Test Successful
```

Figure 41: The Easysoft ODBC-ODBC Bridge Test DSN dialog box

18. If you see an error message you need to check some of the fields:

    - If you get `"Authentication Failure"` on its own, then it is `User` and `Password` that are causing the problem. If you get `"Failed to connect to remote driver"` and some other text in which a login failure is noted then `TargetUser` and `TargetAuth` are causing the problem.
    - Any reference to data sources or DSNs refers to the Target DSN.

- Messages refering to RPC mean that the client cannot find the OOB Server. Check `Server` and `Port`. `Transport` has no effect at this stage, because only TCP/IP is supported.
- `DSN` and `Description` have no effect on the test procedure.

– OR –

If you see "`Test successful`" you have connected to the server:

19. Click **OK** in the test box and **OK** in the DSN dialog box.

The connection has been made.

Connecting a Client Application in Windows

You should now have a data source on your Windows machine that connects through the Easysoft ODBC-ODBC Bridge to a second data source on the remote server machine (either your own server or `demo.easysoft.com`).

To demonstrate that the Easysoft ODBC-ODBC Bridge is functioning correctly you can connect an example ODBC application to the local data source.

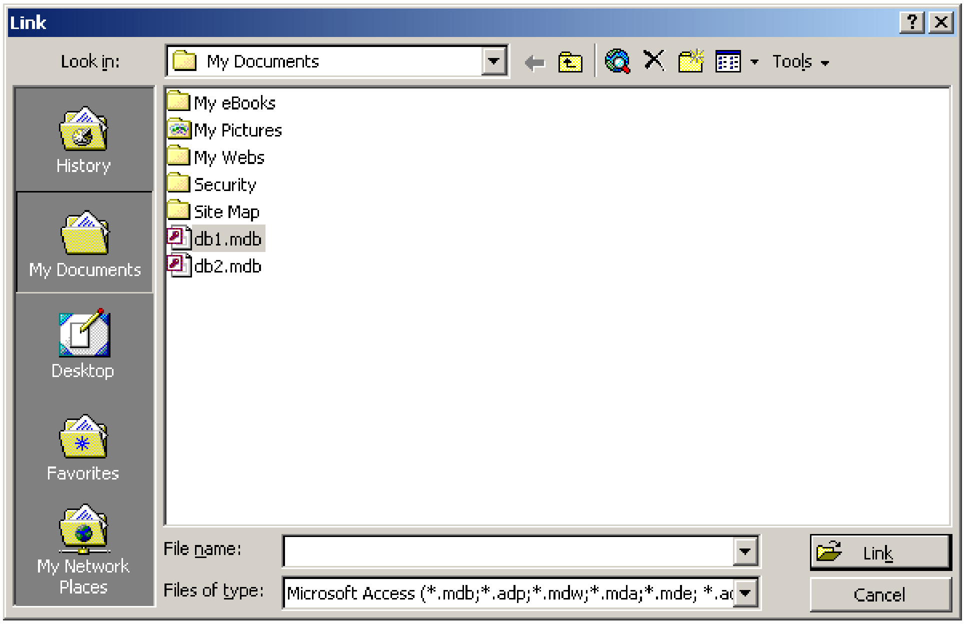
You will need experience of Microsoft Access to complete this section.

Caution!

> If you are using the Easysoft ODBC-ODBC Bridge across the internet (e.g. to contact the Easysoft demo server) then Microsoft Access is *not* recommended, as its very heavy use of ODBC calls generates significant network traffic when using the Easysoft ODBC-ODBC Bridge. **"The Demo.exe Client" on page 139** describes a more efficient client program.

1. Start Microsoft Access (for example) and create a blank database.
2. Select **File > Get External Data > Link Tables**.

The **Link** dialog box displays the existing databases on your system:



Look in: My Documents

My eBooks
My Pictures
My Webs
Security
Site Map
db1.mdb
db2.mdb

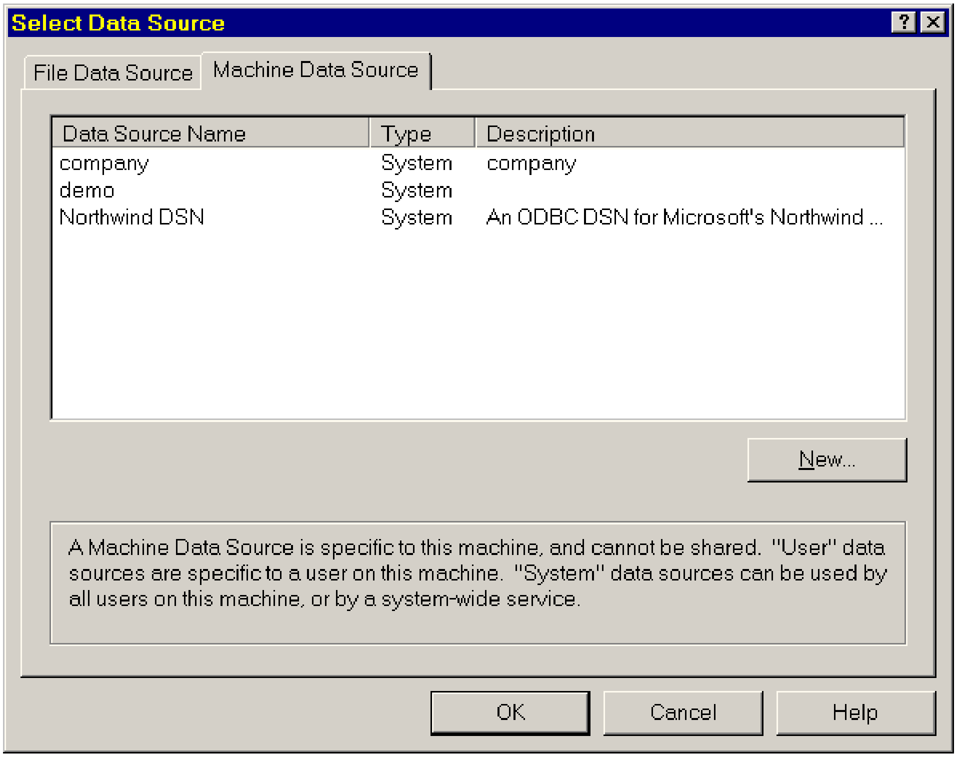
File name:

Files of type: Microsoft Access (*.mdb;*.adp;*.mdw;*.mda;*.mde; *.ac

Figure 42: The Link dialog box in Microsoft Access

3. From the **Files of type** drop-down list, choose **ODBC Databases**.

The Microsoft ODBC driver manager displays the **Select Data Source** dialog box:



Data Source Name	Type	Description
company	System	company
demo	System	
Northwind DSN	System	An ODBC DSN for Microsoft's Northwind ...

A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service.

Figure 43: The Select Data Source dialog box Machine Data Source tab

4. Click the **Machine Data Source** tab.

   Find the local data source you created, somewhere in the list. Note that your description of the data source is displayed beside it.

5. Select your data source and click **OK**.

   Microsoft Accessconnects to the OOB data source and the OOB client relays the ODBC API calls to the OOB Server. Miscrosoft

Access will retrieve a list of tables in the remote data source and dissplay them in a window.

6. Click on a table and then click **OK**.

   After a short wait, you are returned to the **Database** window.

7. Double-click any table to open and browse it.

The Demo.exe Client

An application called `demo.exe` is included with the Easysoft ODBC-ODBC Bridge that can be used to verify that a connection has been made to an ODBC data source.

The `demo.exe` program contains the configuration for the OOB Server on `demo.easysoft.com` internally and therefore can link to that data source in order to test your OOB Client installation.

Note that you can also use `demo.exe` to connect to any Easysoft ODBC-ODBC Bridge data sources you have created yourself.

The source code is included in the demonstration program to give developers an insight into creating simple ODBC client applications.

NB	The `demo.exe` client application will not work without a local data source having been configured. Refer to the section **"Windows client setup" on page 126** and create the demo data source. Note that the data source must be named `demo` for some parts of `demo.exe` to work.

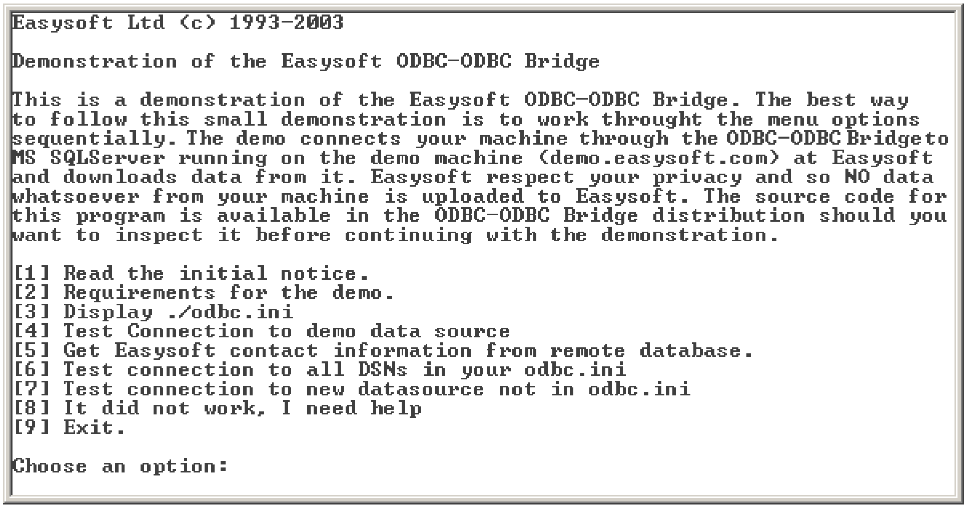
1. Start an MS-DOS window.

2. Change directory to the `Examples` directory within the `easysoft` install directory.

   e.g.

`Program Files\Easysoft\Easysoft ODBC ODBC Bridge`
`\Examples`

3. Execute the `demo.exe` program:



```
Easysoft Ltd (c) 1993-2003

Demonstration of the Easysoft ODBC-ODBC Bridge

This is a demonstration of the Easysoft ODBC-ODBC Bridge. The best way
to follow this small demonstration is to work throught the menu options
sequentially. The demo connects your machine through the ODBC-ODBC Bridge to
MS SQLServer running on the demo machine (demo.easysoft.com) at Easysoft
and downloads data from it. Easysoft respect your privacy and so NO data
whatsoever from your machine is uploaded to Easysoft. The source code for
this program is available in the ODBC-ODBC Bridge distribution should you
want to inspect it before continuing with the demonstration.

[1] Read the initial notice.
[2] Requirements for the demo.
[3] Display ./odbc.ini
[4] Test Connection to demo data source
[5] Get Easysoft contact information from remote database.
[6] Test connection to all DSNs in your odbc.ini
[7] Test connection to new datasource not in odbc.ini
[8] It did not work, I need help
[9] Exit.

Choose an option:
```

Figure 44: The demo.exe program

The demonstration program is self-explanatory: select an option, press *<Enter>* and then follow the information displayed on screen.

[1] displays the initial notice.

[2] displays the requirements for the demo.

[3] this option is not supported on Windows.

[4] tests the connection by contacting `demo.easysoft.com` through the Easysoft ODBC-ODBC Bridge.

[5] retrieves a table from the database on `demo.easysoft.com`.

NB	Option 5 will only retrieve data when your demo data source is configured to connect to the `demo.easysoft.com` server. You will probably receive errors if you configure your demo data source to connect to another database.

[6] attempts to connect to all user data sources one at a time. Whether the connection fails or succeeds is displayed in a Status column. If the connection fails, diagnostics are provided.

NB	If a data source points to a machine that cannot currently be contacted, the connection attempt may take a couple of minutes to time out.

[7] builds up a connection string to connect to the OOB Client ODBC driver but with values you specify to demo.exe (you do not need to create a data source). You are prompted for each value in turn (the default values build up a connection string to the demo data source on `demo.easysoft.com`). The connection string is displayed in full before demo.exe attempts to connect using that connection string.

[8] displays suggestions as to what may be the problem with your connection.

[9] exits from the `demo.exe` program.

Unix client setup

This section explains the steps you should take to connect an ODBC application on a Unix machine (where the OOB Client is installed) to the data source on your remote server (where the OOB Server is installed).

To connect the OOB Client in Unix to a remote data source, you need to define the data source on the remote machine by specifying its attributes in a data source on the local (client) machine.

You can create a data source for the OOB Client on the local Unix machine either by:

- directly adding the data source and its attributes to a configuration file (`odbc.ini`)
- if you are running an X server you can create a data source using the graphical ODBC Data Source Administrator (`ODBCConfig`).

Driver manager functionality in the Easysoft ODBC-ODBC Bridge

Although you are recommended to install the unixODBC Driver Manager as part of the Easysoft ODBC-ODBC Bridge installation, it is not compulsory, and because of this the Easysoft ODBC-ODBC Bridge has been designed with some driver manager functionality in place.

This does not mean that the OOB Client will load third-party ODBC drivers on the local machine, but it does support driver manager calls such as `SQLDrivers()` and `SQLDataSources()`, and will perform some mappings if, for instance, an ODBC 2.0 application connects to an ODBC 3.0 data source.

The most important driver manager functionality provided by the Easysoft ODBC-ODBC Bridge is the storing of data source attributes in order to be able to connect given a minimal connection string.

When an application connects through the OOB Client, it can pass in as little information as just a data source name (DSN).

The OOB Client will search for a configuration file in this sequence:

- $ODBCINI (i.e. it looks for the file whose name is stored in the environment variable `ODBCINI`)
- *`current-dir/`*`odbc.ini`
- *`current-dir/`*`.odbc.ini`
- `$HOME/odbc.ini`
- `$HOME/.odbc.ini`
- `/etc/odbc.ini` (for system data sources only)
- *sysconfdir/*`odbc.ini` (for system data sources only)

For the last location, the OOB Client queries the unixODBC driver manager for information about the DSN.

When unixODBC is installed with the Easysoft ODBC-ODBC Bridge, *sysconfdir* is set to `/etc`, but if you have built and configured your own unixODBC then it will be whatever you specified in the `sysconfdir=`*directory* configuration option, or `/usr/local/etc` if omitted.

NB	If you are running Apache/PHP or ColdFusion with the Easysoft ODBC-ODBC Bridge, Easysoft recommend you use `/etc/odbc.ini`. Putting the file in the web server document tree risks making it publicly accessible, and putting it in the ‘current directory’ may be meaningless as the web server can be started from any directory.

Creating a DSN by editing a configuration file

With unixODBC, data sources are stored in a configuration file called `odbc.ini`.

If you have accepted the default Easysoft ODBC-ODBC Bridge installation, user data sources are located in `$HOME/.odbc.ini`.

You usually have to be logged in as `root` to edit the system `odbc.ini`, which is located in the `/etc` directory.

Each section of the `odbc.ini` file starts with a data source name in square brackets [ ] followed by a number of *`attribute`*`=`*`value`* pairs.

NB Attribute names in `odbc.ini` are not case sensitive.

To create a data source by defining it in `odbc.ini`, open this file in an editor, add a new section and set of attributes:

Name	Description
Description	A single line of text to describe the data source to users.
Driver=OOB	unixODBC looks up this entry in `odbcinst.ini` to locate the shared object to use as the ODBC driver.
Server	The name or IP address of the remote host where the OOB Server is running.
Port	The port (or service name) on which the OOB Server is listening.
Transport	The network transport used across the Easysoft ODBC-ODBC Bridge. This currently must be set to `tcpip`

Name	Description
LogonUser	The name of a user on the remote machine. The OOB Server changes to this user when an incoming connection is made. On Windows NT, you may need to include the NT domain name in this attribute. If so, use the format `domain/user name`. For example: `admin/John Smith`.
LogonAuth	The password for `LogonUser`
TargetDSN	The system DSN on the remote machine
TargetUser	If present, this is passed to the database engine as the `UID` attribute. This is only required if your database engine requires a database logon.
TargetAuth	The password for `TargetUser`. This is passed to the database engine as the `PWD` attribute.
UseOOBDBAuth	When set to 1 (on) the Easysoft ODBC-ODBC Bridge ignores the UID and PWD values passed from the application in the connection string and use the `TargetUser` and `TargetAuth` values for UID/PWD instead. this attribute may need setting for some software (e.g. Crystal Info, which passes an empty PWD and expects the driver to issue a prompt dialog box), but it is not always possible when the driver is on a remote machine. By default this option is set to 0 (off).
BlockFetchSize	When set to > 0 the Easysoft ODBC-ODBC Bridge determines whether to perform an optimization which retrieves multiple rows of data instead of one row at a time. This optimization is not performed if the application itself binds columns in the result set. If your ODBC application uses cursors or positioned updates/deletes, you should not set this to greater than 1 (see **Easysoft ODBC-ODBC Bridge FAQ**).

Name	Description
MetaDataBlockFetch	Takes a value of 0 (off) and 1 (on), default is on. When set enables blockfetching for meta data without affecting other result sets, which will increase the speed of retrieving meta data such as lists of tables or columns.
Unquote_Catalog_Fns	If present, fixes a bug in Applixware (see **Easysoft ODBC-ODBC Bridge FAQ**).
MetaData_ID_Identifier	If present, fixes a bug with some older applications which accidentally use wildcards in catalog functions (see **Easysoft ODBC-ODBC Bridge FAQ**).
DisguiseWide	Set to a value of either 0 (off) or 1 (on). The default is off. When set, disguises the column types described by the ODBC driver as SQL_Wxyz for applications that do not understand wide characters. For example, some versions of StarOffice need this enabled when connecting to Microsoft SQLServer.

Figure 45: Local ODBC attribute settings

There are four classes of attribute here:

- `Description` is effectively a comment, although it may be displayed to users if you have a GUI-enabled driver manager installed.
- `Driver` is a driver manager attribute identifying the driver.
- `Server`, `Port`, `Transport`, `LogonUser` and `LogonAuth` are used to find a remote machine and connect to the OOB Server.

  An example `telnet` session would would be of the format:

  `telnet` *`server port`*

  `login:` *`logonuser`*

  `password:` *`logonauth`*

- `TargetDSN`, `TargetUser` and `TargetAuth` are renamed `DSN`, `UID` and `PWD` respectively, and passed by the OOB Server to its driver or driver manager.

  `UseOOBDBAuth` is used to control the implementation of these settings, which can be used in the format of the following example database session, run once logged on to a remote machine:

  `isql` *`targetdsn targetuser targetauth`*

- `BlockFetchSize`, `MetaDataBlockFetch`, `Unquote_Catalog_Fns`, `MetaData_ID_Identifier` and `DisguiseWide` are all flags to change the behavior of the Easysoft ODBC-ODBC Bridge, according to the specific application.

NB

You must provide the Easysoft ODBC-ODBC Bridge with `Server`, `Port`, `Transport`, `LogonUser`, `LogonAuth` and `TargetDSN`. This is enough information to allow the server to present the DSN to its driver or driver manager.

Don’t forget that any attributes not specific to the Easysoft ODBC-ODBC Bridge are passed through to the remote data source, so you can effectively set up the remote data source from the local machine.

The `odbc.ini` file can also contain global settings.

These will apply to *all* your data sources and should be included in a `{Settings}` section at the end of the `odbc.ini` file::

Name	Description
NetConnectRetryCount	The number of additional times that the OOB Client will attempt to connect to the OOB Server if the first attempt is refused by the server operating system. The internal Easysoft ODBC-ODBC Bridge default is 4, so the OOB Client will make five attempts to connect to the OOB Server before reporting a failure. If you are connecting to OOB Enterprise Servers and using the fallback server capability (i.e. a list of multiple OOB Servers in the DSN), you should set `Connect Retries` to 1 to minimise the time taken to connect to subsequent servers (see **"Client support for Fallback OOB Servers" on page 237**).
NetRCVLOWAT	The size of the receive low-water mark. Easysoft recommend that you leave this set to -1 (the internal Easysoft ODBC-ODBC Bridge default) unless advised otherwise. Other possible values are zero to not set the value at all (i.e. use the operating system default) or a value greater than zero.
NetRCVBUF	The size of the receive queue buffer. By default it is set to -1, the internal Easysoft ODBC-ODBC Bridge default which is 16384 bytes. To turn this setting off (i.e. use the operating system default), set the value to 0; to increase the receive queue buffer, set it to a value greater than 0. On some platforms, increasing the receive queue buffer will speed up the transfer of data across the network though this will be at the expense of memory. Easysoft recommend that you do not change this setting unless advised otherwise.

Name	Description
NetSNDBUF	The size of the send queue buffer. By default it is set to -1, the internal Easysoft ODBC-ODBC Bridge default which is 16384 bytes. To turn this setting off (i.e. use the operating system default), set the value to 0; to increase the send queue buffer, set it to a value greater than 0. On some platforms, increasing the send queue buffer will speed up the transfer of data across the network though this will be at the expense of memory. Easysoft recommend that you do not change this setting unless advised otherwise.

Figure 46: Global ODBC attribute settings

`NetRCVLOWATS`, `NetRCVBUFS` and `NetSNDBUFS` can also be set via the **Configuration** screen of the Web Administrator (see **"Configuration" on page 157**).

AN EXAMPLE ODBC.INI FILE

If you have a Linux box called `linus.mydomain` and Microsoft SQLServer and the OOB Server running on a remote Windows NT machine called `ntbox.mydomain` with a Windows NT user name of `myname` and your password of `mypassword`.

Imagine you have set up a system data source on `ntbox` (see **"Windows server setup" on page 97**) called `myNTdsn` which requires database authentication `dbuser` and `dbpassword`.

You want to access data in Microsoft SQLServer on `ntbox` from `linus` using your Perl program or some PHP.

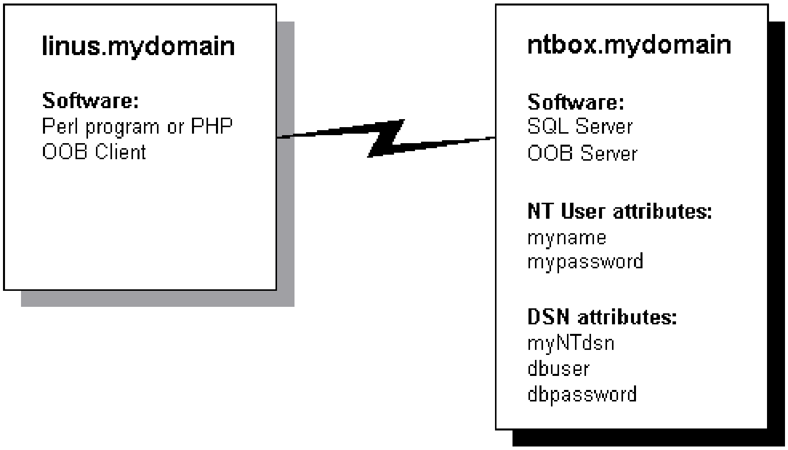


Figure 47: A Linux client connected to an NT server

The Linux box `odbc.ini` file would contain a section of the format:

```
[localdsn]
Driver=OOB
Server=ntbox.mydomain.com
Port=8888
Transport=tcpip
LogonUser=myname
LogonAuth=mypassword
TargetDSN=myNTdsn
TargetUser=dbuser
TargetAuth=dbpassword
```

NB	If using a driver manager at the client end, unixODBC looks up the `Driver=` entry in the `odbcinst.ini` file to locate the shared object to use as the ODBC driver.

When your application connects through the OOB Client, it needs to pass in the DSN `localdsn`, which the OOB Client then uses to access the correct section in the `odbc.ini` file.

Creating a DSN using the ODBC Data Source Administrator

To create a data source using the graphical ODBC Data Source Administrator:

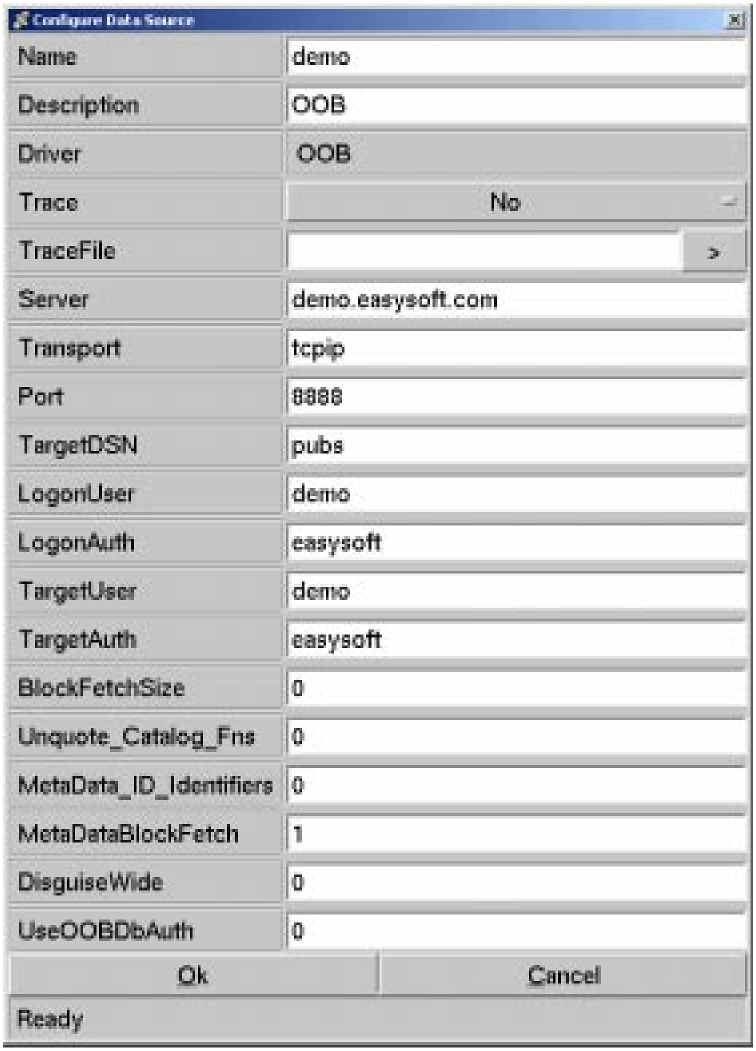
1. Run an X session on your Unix machine, logged in as `root`.
2. Change into the `<`*`InstallDir`*`>/easysoft/unixODBC/bin` directory.
3. Type `./ODBCConfig` and press <*`Enter`*>.

   The ODBC Data Source Administrator opens.
4. Click the **System DSN** tab to create a data source which is available to any user or service that logs into this machine.

   If you create a user DSN, only the `root` user will be able to access it because you logged in as `root` when you started your session. User DSNs will be placed in the `.odbc.ini` file in the home directory of the current user (i.e. `$HOME/.odbc.ini`).
5. Click **Add** to create a new data source.

   The **Adding a New Data Source** dialog box lists the available drivers.
6. Select the Easysoft ODBC-ODBC Bridge driver and then click **OK**.

The **Configure Data Source** dialog box is displayed:



Name	demo
Description	OOB
Driver	OOB
Trace	No
TraceFile	
Server	demo.easysoft.com
Transport	tcpip
Port	8888
TargetDSN	pubs
LogonUser	demo
LogonAuth	easysoft
TargetUser	demo
TargetAuth	easysoft
BlockFetchSize	0
Unquote_Catalog_Fns	0
MetaData_ID_Identifiers	0
MetaDataBlockFetch	1
DisguiseWide	0
UseOOBDbAuth	0

Ready

Figure 48: The Configure Data Source dialog box

The options that you can set on this dialog box are the same as the attributes that you can specify when defining a data source in the `odbc.ini` file.

Refer to **"Creating a DSN by editing a configuration file" on page 144** for information about each setting and details of how they are handled by the Easysoft ODBC-ODBC Bridge.

7. Click **OK** when you have specified all the data source attributes and then close the ODBC Data Source Administrator.

   Once you have a data source on your Unix client connecting to a data source on your server, you can run an application on your client machine and connect to the data source on the server machine.

The Demo Application

The distribution includes a simple ODBC program for connecting to any ODBC data source in order to test that it is working.

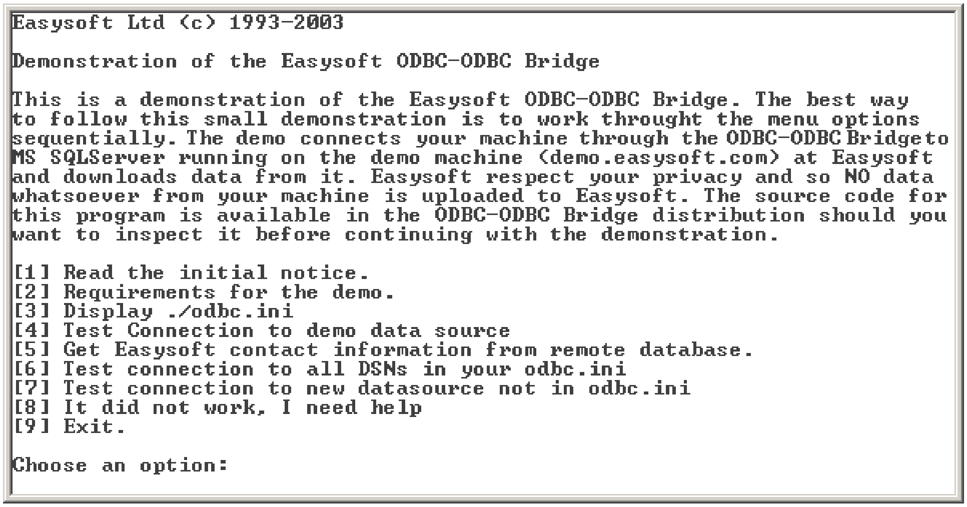
This program can be used in the event that your own remote ODBC data source is not yet configured.

To run the demo program:

1. Change into the `<`*`InstallDir`*`>/easysoft/oob/examples` directory.
2. Type:

   ```
   ./demo
   ```

   The source code for this demo program is in the `examples` subdirectory.



```
Easysoft Ltd (c) 1993-2003

Demonstration of the Easysoft ODBC-ODBC Bridge

This is a demonstration of the Easysoft ODBC-ODBC Bridge. The best way
to follow this small demonstration is to work throught the menu options
sequentially. The demo connects your machine through the ODBC-ODBC Bridgeto
MS SQLServer running on the demo machine (demo.easysoft.com) at Easysoft
and downloads data from it. Easysoft respect your privacy and so NO data
whatsoever from your machine is uploaded to Easysoft. The source code for
this program is available in the ODBC-ODBC Bridge distribution should you
want to inspect it before continuing with the demonstration.

[1] Read the initial notice.
[2] Requirements for the demo.
[3] Display ./odbc.ini
[4] Test Connection to demo data source
[5] Get Easysoft contact information from remote database.
[6] Test connection to all DSNs in your odbc.ini
[7] Test connection to new datasource not in odbc.ini
[8] It did not work, I need help
[9] Exit.

Choose an option:
```

Figure 49: The demo.exe program on Unix

The demonstration program is self-explanatory: select an option, press *<Enter>* and then follow the information displayed on screen.

[1] displays the initial notice.

[2] displays the requirements for the demo.

[3] displays the contents of `./odbc.ini`.

[4] tests the connection by contacting `demo.easysoft.com` through the Easysoft ODBC-ODBC Bridge.

[5] retrieves a table from the database on `demo.easysoft.com`.

NB	Option 5 will only retrieve data when your demo data source is configured to connect to the `demo.easysoft.com` server. You will probably receive errors if you configure your demo data source to connect to another database.

[6] attempts to connect to each data source in the local `odbc.ini` file. Whether the connection fails or succeeds is displayed in a Status column. If the connection fails, diagnostics are provided.

NB	If a data source points to a machine that cannot currently be contacted, the connection attempt may take a couple of minutes to time out.

[7] builds up a connection string to connect to the OOB Client ODBC driver but with values you specify to demo.exe. You are prompted for each value in turn (the default values build up a connection string to the demo data source on `demo.easysoft.com`). The connection string is displayed in full before `demo.exe` attempts to connect using that connection string.

[8] displays suggestions as to what may be the problem with your connection.

[9] exits from the `demo.exe` program.

This page left blank intentionally

CHAPTER 4 CONFIGURATION

Configuring the Easysoft ODBC-ODBC Bridge

This section explains the configuration options available for the Easysoft ODBC-ODBC Bridge (OOB) Server. The configuration for Unix subsection also covers how the OOB Server for Unix can be run as a standalone server process without `{x}inetd`.

Chapter Guide

- **The Web Administrator**
- **Configuring the OOB Server under Windows 2000/NT**
- **Configuring the OOB Server under Unix**

The Web Administrator

There are a number of configurable parameters for the OOB Server, irrespective of platform. There are platform-specific ways to change these settings, but the OOB Server also provides a simple HTTP server, known as the Web Administrator, that provides a web-browser based interface to use for updating these parameters.

The Web Administrator is available in the OOB Server for Windows and in the OOB Server for Unix when run as a standalone server process without `{x}inetd` (see **"Configuring the Server as Standalone" on page 190** for details).

This section describes the Web Administrator and shows you how to change the server configurable parameters.

In order to run the Web Administrator you will need:

- a Windows machine running the OOB Server or a Unix machine with the OOB Server running standalone
- an HTTPAdmin user name and password unless the OOB Server adminisrative user name is disabled

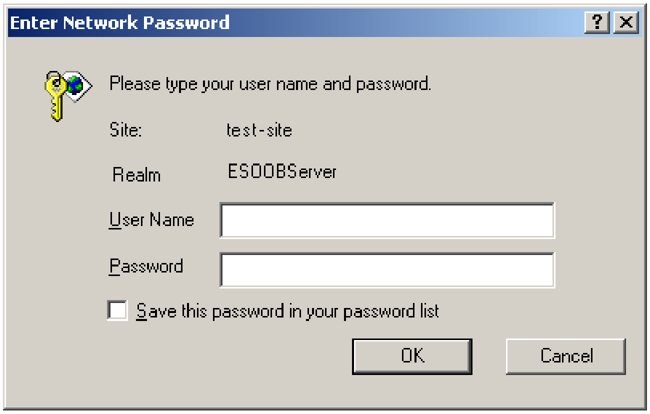
NB	To obtain the HTTPAdmin user name that was entered for your OOB Server when it was installed (HTTP authentication will have been disabled if this field was left blank) display the `HTTPAdmin` setting on the **Configuration** screen (see **"HTTPAdmin" on page 168**). This setting is hidden by default, but may be reset. The HTTPAdmin user name and password are required in order to make changes through the Web Administrator. If this value is not a valid user on the server system, then you must change it. The value is case sensitive.

Start the Web Administrator by opening the URL `http://`*oobhost*`:8890/` from a web browser (substituting your machine name in place of `oobhost` and the required port address if you have specified a port other than the Easysoft default of `8890`).

NB	To obtain the port number for the Web Administrator you need to refer to the `HTTPPort` setting on the **Configuration** screen (see **"HTTPPort" on page 167**).

If HTTPAdmin is set to anything other than `"disabled"` then some pages in the Web Administrator are protected with HTTP authentication.

If you click on a link to a protected page you will be prompted for the HTTPAdmin user name and password by your browser for the ESOOBServer realm:



Please type your user name and password.

Site: test-site

Realm ESOOBServer

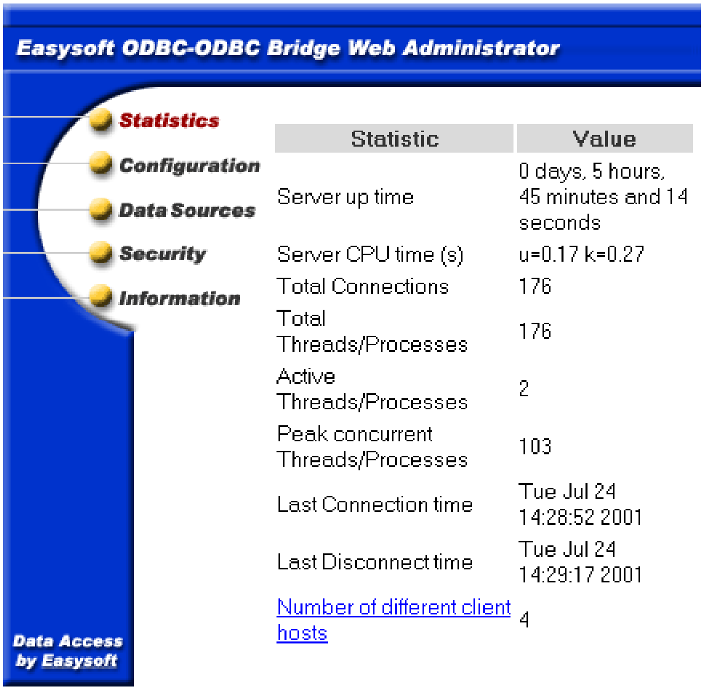
User Name

Password

□ Save this password in your password list

Figure 50: The Enter Network Password dialog box

The web page returned is generated by the server process and displays runtime statistics for the latest run of the server:



Statistic	Value
Server up time	0 days, 5 hours, 45 minutes and 14 seconds
Server CPU time (s)	u=0.17 k=0.27
Total Connections	176
Total Threads/Processes	176
Active Threads/Processes	2
Peak concurrent Threads/Processes	103
Last Connection time	Tue Jul 24 14:28:52 2001
Last Disconnect time	Tue Jul 24 14:29:17 2001
Number of different client hosts	4

Figure 51: The Statistics screen of the Web Administrator

The following additional screens can be accessed from the **Statistics** screen:

- **Configuration** defines OOB Server features.
- **Data Sources** displays data source and driver information.
- **Security** allows a user to define the IP addresses of those machines which have access to the OOB Server enabled or disabled.
- **Information** provides access to various Easysoft locations providing documentation and user support.

Client Hosts displays details of client machines which have connected to the OOB Server.

The **Statistics** and **Data Sources** screens display information, whilst the **Configuration** and **Security** screens allow you to modify the server’s runtime parameters.

THE STATISTICS SCREEN

The Web Administrator **Statistics** screen provides the following infromation:

- **Server up time -** The time in days, hours, minutes and seconds since the OOB Server was started.
- **Server CPU time(s) -** This is only visible if the `ShowProcessTime` option on the **Configuration** screen (see **"ShowProcessTime" on page 173**) is selected. One or more values are shown. If only one value is shown, it is the total CPU time consumed by the OOB Server. If two times are shown, the first is the user time and the second is the kernel time. Note that the CPU times shown include CPU time consumed by the ODBC Driver Manager, any ODBC drivers and any child processes.
- **Total Connections -** The total number of connections (or attempted connections) to the OOB Server. This includes connections dropped due to invalid license keys or insufficient license slots, port scanners or anyone using `telnet` to access the OOB Server ODBC port.

- **Total Threads/Processes -** The total number of threads or processes that the OOB Server has created during its execution. Connections denied access because of an access control rule or because the value of either `MaxThreadCount` (see **"MaxThreadCount" on page 169**) or `MaxClientConnect` (see **"MaxClientConnect" on page 169**) has been exceeded do not count because the OOB Server does not start a thread/process for these.
- **Active Threads/Processes -** The total number of active threads or processes the OOB Server has created to handle ODBC connections. However, this number may exceed the actual active count, as the OOB Server only looks for exited threads and processes when five seconds has elapsed without any connections (to give preference to incoming connections). Note that if `MaxThreadCount` (see **"MaxThreadCount" on page 169**) or `MaxClientConnect` (see **"MaxClientConnect" on page 169**) is set to anything other than 0 then the OOB Server has to reap exited threads and processes every time a new connection arrives. Also note that this is not a limit and may exceed your maximum number of licensed slots.
- **Peak concurrent Threads/Processes -** The maximum value recorded in the `Active Threads/Processes` field (see **"Active Threads/Processes" on page 167**).
- **Last Connection time -** The time of the last ODBC connection.
- **Connections/minute (Enterprise only) -** The Server up time (in minutes) divided by the number of `Total Threads/Processes` (see **"Total Threads/Processes" on page 163**).
- **Last Disconnect time -** The time ODBC was last disconnected.

- **Number of different client hosts -** The number of different client machines which have connected to the OOB Server (where a client machine is identified by its IP address). You can click on this link to get a list of IP addresses or machine names. Machine names are only displayed if you have `ReverseLookup` enabled (see **"ReverseLookup" on page 172**).
- **Audit File (Enterprise only) -** A link to another page which shows entries from the last audit trail file. As the audit trail file is renamed every day you will also see links to older audit trail files. Graphs of connections per hour and connections per minute are available for each audit file. Note that an audit trail file is only produced if `AuditODBCAccess` is enabled (see **"AuditODBCAccess" on page 172**).
- **DSN statistics (Enterprise only) -** Shows the first ten DSNs accessed over the Easysoft ODBC-ODBC Bridge, the number of times a connection to this DSN has been opened, the total time (in seconds) connections to this DSN have been open, the connections per minute and the average time per connection. Note that the total time is concurrent time (i.e. if the same DSN is open twice concurrently for ten seconds, Total time shows 20 seconds).

Changing the refresh frequency

The Web Administrator uses a set of template files into which the dynamic data is inserted before sending it back to your browser.

The template file for the **Statistics** screen is `index.html`, which is located in the `/templates` directory wherever you installed the OOB Server.

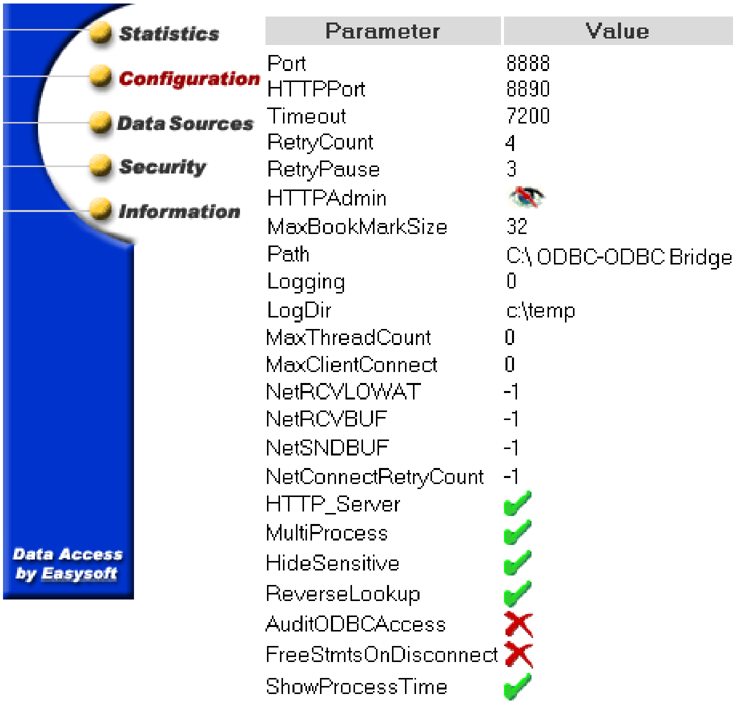
Edit the `index.html` file and near the top you will see:

```
meta http-equiv="refresh" content="60"; URL=/index.html
```

Change the 60 (the refresh time in seconds) to your preferred setting.

NB	Note that setting the refresh time to a very low value will increase the workload on the OOB Server process which handles HTTP requests. As this may reduce the reponse time to the Easysoft ODBC-ODBC Bridge ODBC server thread, times much less than 60 seconds are not recommended.

THE CONFIGURATION SCREEN



Parameter	Value
Port	8888
HTTPPort	8890
Timeout	7200
RetryCount	4
RetryPause	3
HTTPAdmin	👁
MaxBookMarkSize	32
Path	C:\ ODBC-ODBC Bridge
Logging	0
LogDir	c:\temp
MaxThreadCount	0
MaxClientConnect	0
NetRCVLOWAT	-1
NetRCVBUF	-1
NetSNDBUF	-1
NetConnectRetryCount	-1
HTTP_Server	✓
MultiProcess	✓
HideSensitive	✓
ReverseLookup	✓
AuditODBCAccess	✗
FreeStmtsOnDisconnect	✗
ShowProcessTime	✓

Figure 52: The Configuration screen of the Web Administrator

This section explains the parameters that are configurable via the Web Administrator **Configuration** screen. These parameters apply either to the OOB Server itself, or as a default value for all DSNs accessed via the server.

These settings, and some additional settings, can also be edited via the registry in Windows (see **"Server Configurable Parameters in the Registry" on page 185**) or in a file in Unix (see **"Changing Server Configurable Parameters under Unix" on page 192**).

Note that enabling the `HideSensitive` attribute will result in some sensitive settings not being displayed (see **"HideSensitive" on page 171**).

You can modify the server settings by clicking **Change** and then typing the HTTP Admin user name and password when prompted. This is the user name that you entered during the installation process and the password for that user on the system where the OOB Server is running.

NB	You are only asked for this if the HTTP authentication has been enabled by the HTTP Admin user name being previously entered as something other than "`disabled`".

Keys are not case-dependent, so for example `Port`, `PORT` and `pOrT` are treated as the same key. Values are case-dependent if the operating-system is, so it is best to match case where possible.

The **Configuration** screen contains the following fields (click **Submit** to make your changes):

- **Port -** The port where the OOB Server is listening for incoming OOB Client connections.

- **IPAddress (Enterprise-only) -** The IP address the OOB Server listens on.

  This is only relevant if you have multiple NICs and only want the OOB Server to listen on one of them. By default the server binds to port `8888` on all local NICs.

- **HTTPPort -** The port on which the OOB Server will listen for HTTP requests.

  Although the port number defaults to `8890`, any port not in use on your machine may be specified.

  If the `Flags` configurable option `HTTP_SERVER` attribute is set (see **"HTTP_Server" on page 171**) then the OOB Server will listen on the specified port for HTTP requests as well as acting in its normal role serving the OOB Client.

  You can communicate with the OOB Server by using the URL `http://machine_name:HTTPPort` from your browser (where `machine_name` is the name or IP address of the OOB Server machine and `HTTPPort` is the port number). It can show statistics, DSNs and configurable parameter values.

- **Timeout -** The inactivity timeout in seconds (the default is `7200` i.e. two hours).

  The OOB Server starts a new thread (or process) for each client that connects and if there has been no communication in `Timeout` seconds each thread or process exits.

  This ensures clients which fail to closedown properly do not cause increasing resource usage on the server. To disable the timeout, set it to `0`.

- **RetryCount -** The number of additional attempts the server will make at creating a thread or process to handle an incoming connection or obtaining a license slot for a new connection after the first attempt fails.
- **RetryPause -** The time in seconds between each retry attempt (see **"RetryCount" on page 168**).
- **HTTPAdmin -** The user name of the person allowed to make changes to the OOB Server via the Web Administrator.

  This must be a valid user name in the operating system the server is running on. If the string is set to "`disabled`" (omit the quotes) then HTTP authentication is not required (this does not mean the OOB Server stops authenticating incoming ODBC connections). The value is case-sensitive.

NB The "eye" symbol displayed against the `HTTPAdmin` parameter denotes that the value of the field is only visible to the user via the **Change Configuration** screen, which requires the entry of a user name and password.

- **MaxBookMarkSize -** The size of the largest bookmark with which the Easysoft ODBC-ODBC Bridge can function.

  ODBC 2.0 uses fixed length bookmarks of 4 bytes. In ODBC 3.0 bookmarks are all variable in size. `MaxBookMarkSize` defaults to 32 bytes. Contact **support@easysoft.com** if you use an ODBC driver requiring a bookmark size greater than 32 bytes.
- **Path -** The installation path of the Easysoft ODBC-ODBC Bridge.

  This is a read-only parameter for information only.

- **Logging -** A bitmask specifying the events to be logged by the OOB Server.

  `Logging` slows the Easysoft ODBC-ODBC Bridge down considerably and should only be set as directed by Easysoft support. The number may be specified as decimal (e.g. in the format `2047`)or hexadecimal (e.g. in the format `0x7ff`).

- **LogDir -** The directory to which log files (see **"Logging" on page 169**), audit files and the `esoob.exception` unhandled exceptions file are written. The default values are `/tmp` in Unix and `c:\temp` in Windows.

- **MaxThreadCount -** The maximum number of threads or processes the OOB Server will allow at any one time.

  There will be one thread or process for every ODBC connection. There is no limit if `MaxThreadCount` is set to 0 (the default), but setting this parameter will prevent too many simultaneous connections swamping the server.

- **MaxClientConnect -** The maximum number of concurrent connections from a single client (where a client is defined as the IP address of the machine where the client is running).

  If `MaxClientConnect` is set to 0 (the default), there is no limit. You can use this parameter to prevent people from swamping your machine with ODBC connections.

- **NetRCVLOWAT -** The size of the receive low-water mark in bytes.

  Possible values are `0` (set to the operating system default), `-1` (set to 16384 bytes, the internal Easysoft ODBC-ODBC Bridge default - this is the default setting) or n (set to a value n where n > 0).

- **NetRCVBUF -** The size of the receive queue buffer in bytes.

  Possible values are `0` (set to the operating system default), `-1` (set to 16384 bytes, the internal Easysoft ODBC-ODBC Bridge default - this is the default setting) or n (set to a value n where n > 0).

- **NetSNDBUF -** The size of the send queue buffer in bytes.

  Possible values are `0` (set to the operating system default), `-1` (set to 16384 bytes, the internal Easysoft ODBC-ODBC Bridge default - this is the default setting) or n (set to a value n where n > 0).

- **Affinity (Enterprise-only) -** If the OOB Server is running on a multiple CPU Windows Server machine you will see a check box for each CPU (labelled CPU_0 to CPU_N).

  The OOB Server holds the `Affinity` value as a mask of processors in your machine which the OOB Server can run on.

  Initially, the value is 0, which means the OOB Server will make no changes at all and the Web Administrator will show all CPUs as available. However, if you make any changes to the CPU check boxes the OOB Server will call `SetProcessAffinityMask` which will define which processors the threads in the OOB Server are allowed to run on. In this way you can bind the OOB Server to particular CPUs in your machine.

- **Flags -** A bitmask of operational flags, split into check boxes, one for each bit in Flags:
  - **Authentication_Disabled** (0x1) If set then authentication is disabled in the OOB Server. The default is off. Although setting this parameter should be considered as a security risk, on some high hit servers in a controlled environment where you do not need to authenticate the connections this can save a considerable amount of time during connections (e.g. Windows NT can use between 0.25 and 0.75 seconds of the connection time for authentication).
  - **HTTP_Server** (0x2) If set then the Web Administrator is made available as soon as the OOB Server is started. The default is on.
  - **MultiProcess** (0x4) If set then the OOB Server starts a new process for each incoming connection, rather than a new thread. The default is off, so the OOB Server starts a new thread for each connection. Enable `MultiProcess` if an ODBC driver which is not thread-safe is being used or if your ODBC driver leaks memory. Currently on non-Windows platforms the server always starts new processes as it is NOT multi-threaded and so this flag cannot be changed.
  - **MetaDataBlockFetch_Disabled** (0x8) If set prevents the OOB Client from automatically starting block-fetch-mode for read-only metadata result-sets. The default is off.
  - **HideSensitive** (0x10) If set causes the Web Administrator to hide sensitive parameters on the **Configuration** screen. Currently the parameters removed when `HideSensitive` is

set are HTTPAdmin (see **"HTTPAdmin" on page 168**) and Authentication_Disabled (see **"Authentication_Disabled" on page 171**). All parameters are always shown in full on the password protected **Change Configurable** screen. The default is on.

– **ReverseLookup** (0x20) If set causes the OOB Server to call gethostbyaddr() on the connecting client's IP address to obtain the client's machine name. On machines where DNS is not set up properly this can cause problems and in any case adds time to the connection. ReverseLookup currently only affects the number of different clients page on the **Statistics** screen, where "unknown" will be displayed instead of the machine name for the machine names of connecting clients if ReverseLookup is off. The default is off.

– **AuditODBCAccess** (0x40) If set the OOB Server audits all ODBC connections to a log file (named esoob_access.log in the LogDir directory) which may be viewed through the Web Administrator. The default is off.

- **FreeStmtsOnDisconnect** (0x80) If set the OOB Server frees all existing statements in the driver before calling `SQLDisconnect`. By default this flag is off as this function should be executed by the ODBC driver, rather than the application, as detailed in the ODBC specification.

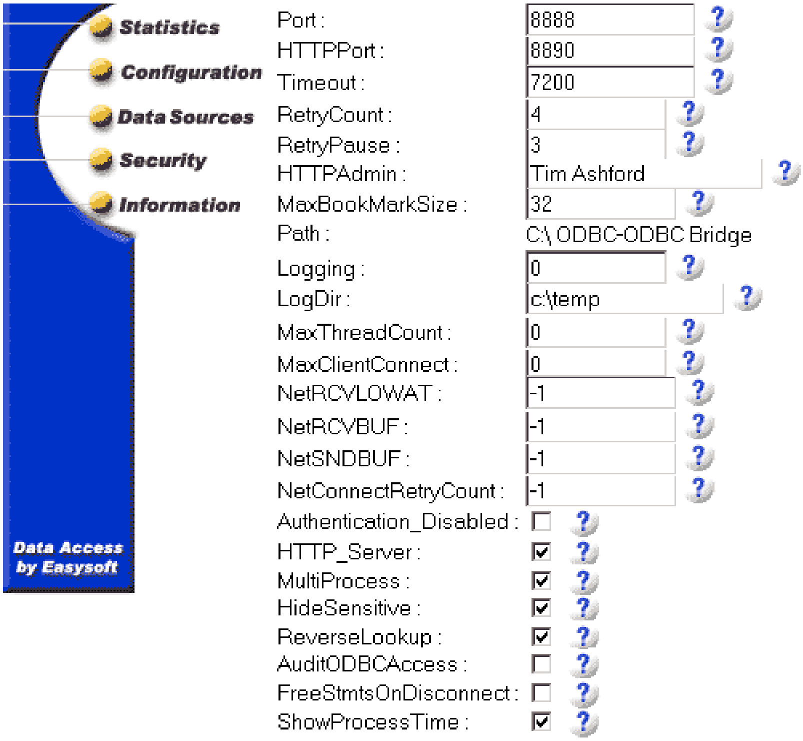
NB	Setting this flag fixes a potential crash in Navision's ODBC driver if `SQLDisconnect` is called from Perl when there are existing statements.

- **ShowProcessTime** (0x100) If set the OOB Server retrieves the user and kernel CPU times for the main OOB Server process and displays them on the **Statistics** screen. The values are only updated once every five seconds. The default is on.

- **AutoUpdateConfiguration (Deprecated)** (0x200) If set the OOB Server automatically updates its internal copy of the configuration settings from the registry when five seconds elapse without an incoming connection. Disabling this feature reduces the number of accesses to the registry, but means that the server has to be restarted to pick up configuration changes. The default is off.

- **HTTPAuthAll (Enterprise-only)** (0x400) If set then all pages (except the index page) in the Web Administrator are protected by HTTP authentication. The default is off.

- **ConnectionPooling (Enterprise-only)** (0x800) If set the OOB Server sets the `SQL_ATTR_CONNECTION_POOLING` attribute to `SQL_CP_ONE_PER_DRIVER` automatically. This enables connection pooling in the driver manager for the ODBC driver. The default is off.
- **LogFailingSQL (Enterprise-only)** (0x1000) If set when any call to `SQLPrepare/SQLExecDirect` fails the SQL is logged to the file `oob_sql.log` in the `LogDir`. As some DBMSs only parse the SQL at `SQLPrepare` time sometimes you do not find out the SQL is incorrect until `SQLExecute` is called. To catch these SQL errors too the OOB Server saves the first 1K of any SQL passing `SQLPrepare` successfully so it may be logged if `SQLExecute` fails. This is extremely useful when developing applications or web services and can be used to monitor errors. The default is off.
- **AllowDBBrowse (Enterprise-only)** (0x2000) If set the data sources listed on the **Data Sources** screen may be browsed to get lists of tables, columns in tables and rows of table data. The default is off.

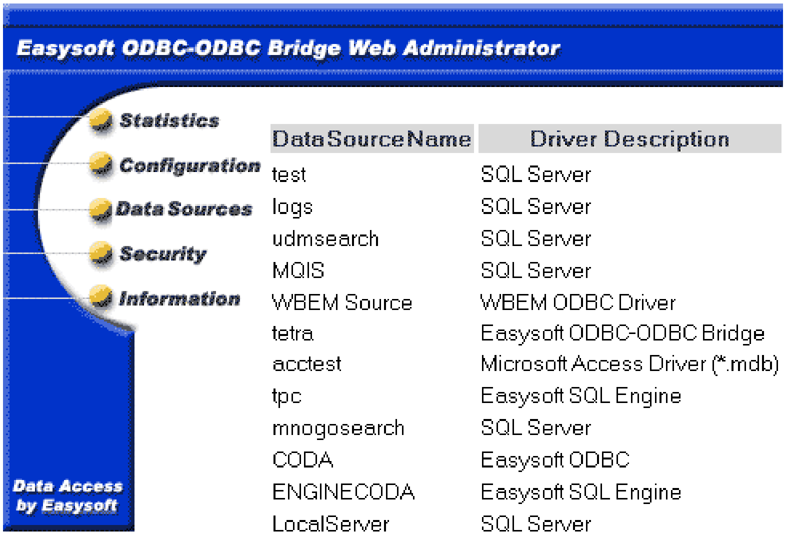


Port :	8888
HTTPPort :	8890
Timeout :	7200
RetryCount :	4
RetryPause :	3
HTTPAdmin :	Tim Ashford
MaxBookMarkSize :	32
Path :	C:\ ODBC-ODBC Bridge
Logging :	0
LogDir :	c:\temp
MaxThreadCount :	0
MaxClientConnect :	0
NetRCVLOWAT :	-1
NetRCVBUF :	-1
NetSNDBUF :	-1
NetConnectRetryCount :	-1
Authentication_Disabled :	☐
HTTP_Server :	☑
MultiProcess :	☑
HideSensitive :	☑
ReverseLookup :	☑
AuditODBCAccess :	☐
FreeStmtsOnDisconnect :	☐
ShowProcessTime :	☑

Figure 53: The Change Configuration screen of the Web Administrator

The Data Sources Screen

The Web Administrator **Data Sources** screen displays the ODBC system data sources found by the server and their configured drivers.



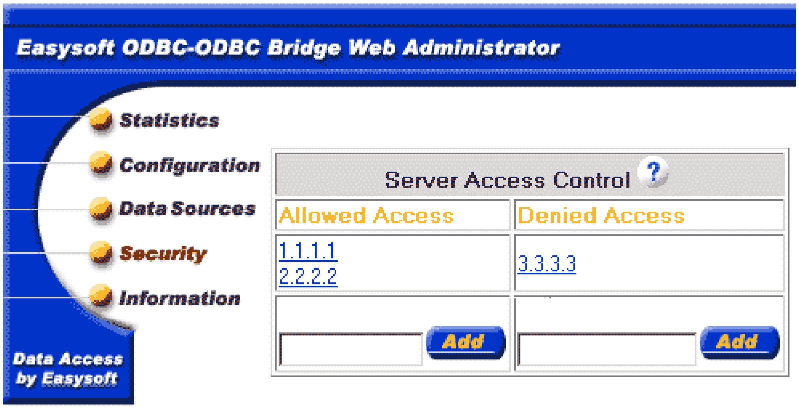
Data Source Name	Driver Description
test	SQL Server
logs	SQL Server
udmsearch	SQL Server
MQIS	SQL Server
WBEM Source	WBEM ODBC Driver
tetra	Easysoft ODBC-ODBC Bridge
acctest	Microsoft Access Driver (*.mdb)
tpc	Easysoft SQL Engine
mnogosearch	SQL Server
CODA	Easysoft ODBC
ENGINECODA	Easysoft SQL Engine
LocalServer	SQL Server

Figure 54: The Data Sources screen of the Web Administrator

The Security Screen

The Web Administrator **Security** screen displays and allows the user to change the set of hosts that are allowed to connect to the Easysoft ODBC-ODBC Bridge.

If they have not already been entered, you are then prompted for the HTTPAdmin user name and password before the **Security** screen is displayed:



Allowed Access	Denied Access
1.1.1.1 2.2.2.2	3.3.3.3

Figure 55: The Security screen of the Web Administrator

In addition to the user names and passwords of your system and of the database management system, the Easysoft ODBC-ODBC Bridge provides another layer of security by using access control lists.

To add an IP address to a list, type the address or address-stem into either the **Allowed Access** or **Denied Access** boxes and click **Add**.

Unix administrators will recognize this mechanism from the `hosts.allow` and `hosts.deny` files.

NB	Though the approach is similar, the rules for determining whether or not a host should be allowed to connect are different from those for `hosts.allow` and `hosts.deny`. Please read the rest of this section for more information.

Easysoft ODBC-ODBC Bridge Security takes the form of two lists of IP addresses.

When a host attempts to connect to the OOB Server, access is only granted if:

- the `allowed` list is empty and the IP address is not in the `denied` list
- the IP address is in the `allowed` list and not in the `denied` list.

The lists can be edited either in the registry (Windows), their relevant flat files (Unix), or via the Web Administrator (Windows and Unix when run standalone.)

Addresses must be entered using the IP ‘dot’ notation.

Entries which consist of fewer than four fields represent all the addresses that match the fields which are defined.

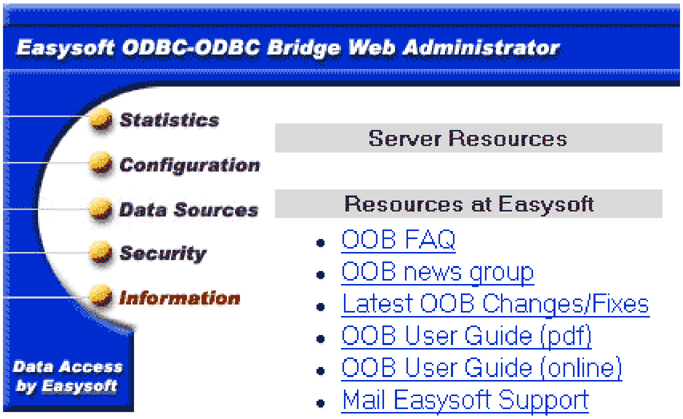
e.g.

`163.141.23.` (note the trailing dot) matches all IP addresses from `163.141.23.0` to `163.141.23.255`.

NB	Direct editing is not supported by Easysoft and will not allow access to changed values until the server has been restarted.

The Information Screen

The Web Administrator **Information** screen displays a list of links to Easysoft support resources:



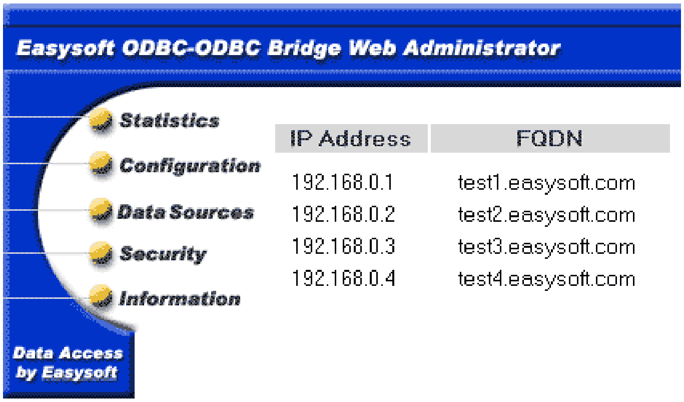
- OOB FAQ
- OOB news group
- Latest OOB Changes/Fixes
- OOB User Guide (pdf)
- OOB User Guide (online)
- Mail Easysoft Support

Figure 56: The Information screen of the Web Administrator

- **OOB FAQ -** a list of Frequently Asked Questions.
- **OOB news group -** a forum for posting questions and answering questions with other users.
- **Latest OOB Changes/Fixes -** a document listing any recent product updates.
- **OOB User Guide (pdf) -** the User Guide in Adobe Portable Document Format.
- **OOB User Guide (online) -** the User Guide in HTML format.
- **Mail Easysoft Support -** a form allowing the user to post a support request by email to Easysoft.

The Client Hosts Screen

The Web Administrator **Client Hosts** screen displays a list of the different clients which have connected to the OOB Server. These connection attempts may not have been fully successful (e.g. not authenticated or denied access).



IP Address	FQDN
192.168.0.1	test1.easysoft.com
192.168.0.2	test2.easysoft.com
192.168.0.3	test3.easysoft.com
192.168.0.4	test4.easysoft.com

Figure 57: The Client Hosts screen of the Web Administrator

The screen contains the following fields:

- **IP Address -** The IP addresses of connecting clients.
- **FQDN -** The Fully Qualified Domain Name of connecting clients. This is displayed as "`unknown`", unless `ReverseLookup` is enabled (see **"ReverseLookup" on page 172**).
- **Connections (Enterprise-only) -** A count of the number of connections from this client.

Configuring the OOB Server under Windows 2000/NT

The OOB Server under Windows 2000/NT is installed as a service, configured to be started automatically by the Windows Service Manager when the machine boots.

A `Services` entry is also added to the Windows Easysoft ODBC-ODBC Bridge menu.

NB	The OOB Server writes errors and startup information to the **Application** section of the **Event Log**. If you suspect that the OOB Server is not functioning correctly you are recommended to check the **Event Log** with the **Event Viewer** before pursuing other lines of enquiry.

The OOB Server runs with administrative privileges, allowing it to become the user specified in the client DSN.

When a user connects, the OOB Server effectively becomes the user specified in the `LogonUser` attribute so that it acts with the specified user's permissions.

You can configure the service like any other service. For instance, you can set it up not to start automatically on system boot.

You can also explicitly stop the service, as follows:

1. Choose:

   Start > Settings > Control Panel > Administrative Tools (in Windows 2000)

   – OR –

   Start > Settings > Control Panel (in NT)

2. Open the `Services` icon.

The Service Manager displays a list of services, along with their status (see **"To Start or Stop the Service in Windows" on page 182**) and their startup configuration (see **"To Configure Automatic/Manual/Disabled status" on page 182**).

To Start or Stop the Service in Windows

1. In the **Services** dialog box, ensure that the `Easysoft ODBC-ODBC Bridge Server` entry is selected.
2. If the service does not have `Started` recorded in the **Status** column, you can click **Start** to run the service.

   – OR –

   If the service has `Started` recorded in the **Status** column, you can click **Stop** to stop the service. Click **Yes** when the Service Manager asks you to confirm your action.

To Configure Automatic/Manual/Disabled status

When installed, the Easysoft ODBC-ODBC Bridge Server is configured to run automatically when Windows starts. Windows offers two other options for services.

The Manual startup mode dictates that the service should start only when requests come in or the user explicitly runs the service. This can save resources if the service in question is rarely used.

It may be that you need to make the service unavailable from time to time, for example if you need to restructure your data source or you have a security policy stating that services should not be available outside allotted times. Windows provides the Disabled state for this.

1. In the **Services** dialog box, ensure that the `Easysoft ODBC-ODBC Bridge Server` entry is selected.

2. Click **Startup...** (*not* **Start**!) to open the Service configuration dialog box.
3. In the **Startup Type** section, choose the relevant option:

   - If you want the service to be started automatically by windows, select **Automatic**. This is the recommended option.

   – OR –

   - If you want the service to be available only when started by a user on the local machine, select **Manual**.

   – OR –

   - If you want to disable the service, select **Disabled**.
4. Click **OK** to close the dialog box.

User Accounts and the Server

Windows distinguishes two types of service with respect to their privileges:

- services running in the System Account have highly privileged access to resources on the local machine whilst idle
- services set to run as a specific user are given the set of privileges assigned to that user.

The OOB Server is designed to be run in the System Account. As soon as a client connects authentication takes place using the `LogonUser` and `LogonAuth` attributes. The server calls `LogonUser()` and `ImpersonateLoggedOnUser()` in the Windows API, so that all subsequent actions are attempted with only the privileges of `LogonUser`.

This is the default configuration of the OOB Server, and it provides the greatest general flexibility and security.

The other option is to set up the server to always run with a specific user's permissions. This is not recommended, but if you do want to do this the procedure is as follows:

1. Go into the OOB web administrator and disable authentication.
2. Be sure that the user account you wish to use is set up and that you know its password. It may be an existing end-user's account, or one created specifically for the OOB Server.

Caution!

For the OOB Server to function correctly when running as a specific user, that user needs two important access rights:`Log On as a Service`, which is needed for the service to start, and `Act as Part of Operating System`, which is needed so that the service can authenticate connecting users for the OOB Server and for the Web Administrator.

3. In the **Services** dialog box, ensure that the `Easysoft ODBC-ODBC Bridge Server` entry is selected.
4. Click **Startup...** (*not* **Start**) to open the Service configuration dialog box.
5. In the **Log On As** section, select **This Account**.
6. In the **This Account** text box, type the user name or click **...** to browse for a user name.
7. In the **Password** box, type the user account password.
8. Type the same password in the **Confirm Password** box.
9. Click **OK** to close the dialog box and commit your changes.

You may have noticed the **Allow Service to Interact With Desktop** checkbox. This has no effect on the OOB Server.

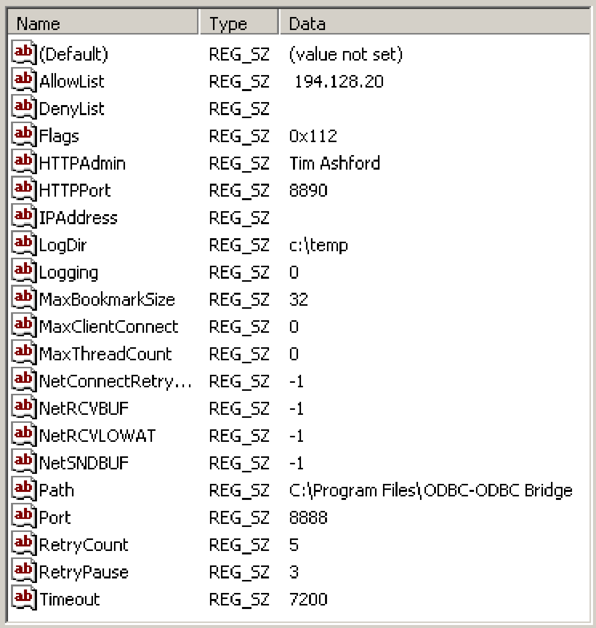
Server Configurable Parameters in the Registry

For experienced users, the OOB Server can be configured by editing the Registry directly. This should not be carried out by novice users, so no step-by-step guide will be provided here.

NB	Direct editing is not supported by Easysoft and will not allow access to changed values until the server has been rebooted.

"The Configuration Screen" on page 165 explains the settings available, which can be found in the Registry under the key:

`\HKEY_LOCAL_MACHINE\SOFTWARE\Easysoft ODBC-ODBC Bridge\Configuration\System\Settings`.



Name	Type	Data
(Default)	REG_SZ	(value not set)
AllowList	REG_SZ	194.128.20
DenyList	REG_SZ	
Flags	REG_SZ	0x112
HTTPAdmin	REG_SZ	Tim Ashford
HTTPPort	REG_SZ	8890
IPAddress	REG_SZ	
LogDir	REG_SZ	c:\temp
Logging	REG_SZ	0
MaxBookmarkSize	REG_SZ	32
MaxClientConnect	REG_SZ	0
MaxThreadCount	REG_SZ	0
NetConnectRetry...	REG_SZ	-1
NetRCVBUF	REG_SZ	-1
NetRCVLOWAT	REG_SZ	-1
NetSNDBUF	REG_SZ	-1
Path	REG_SZ	C:\Program Files\ODBC-ODBC Bridge
Port	REG_SZ	8888
RetryCount	REG_SZ	5
RetryPause	REG_SZ	3
Timeout	REG_SZ	7200

Figure 58: Server parameters in the Registry

The access lists are stored as strings, with IP addresses separated by commas.

The bitmask values (`Flags` and `Logging`) are stored as *strings*, and may represent either the decimal value or the hexadecimal representation (preceded by '`0x`').

The `Flags` setting relates to the options on the **Change Configuration** screen of the Web Administrator (see **"The Configuration Screen" on page 165** for details of these settings).

To set the `Flags` value, choose your required options from **"Registry Flag settings" on page 187** and then enter the sum of their corresponding values.

The `Flags` setting is a bitmask which sets the following options:

Value	Name	Hex Value	Description
1	AUTH_DISABLE	0x1	Disable Authentication in the server for a much quicker connection setup.
2	HTTP_SERVER	0x2	Enable the Web Administrator (Windows and Unix)
4	MULTIPROCESS	0x4	Create a new process on connection rather than a thread (Windows NT). Must be set (Unix).
8	NO_METADATA_BLOCKFETCH	0x8	Disable automatic block-fetch mode for metadata calls.

Value	Name	Hex Value	Description
16	HIDESENSITIVE	0x10	Conceal security-sensitive fields.
32	REVERSELOOKUP	0x20	Enable reverse lookup.
64	AUDITODBCACCESS	0x40	Audit all ODBC connections to a log file written to the `LogDir` directory as `esoob_access.log`.
128	FREESTMTSONDISCONNECT	0x80	Free all existing statements in the driver before calling `SQLDisconnect` (a workaround for Navision ODBC driver problems).
256	SHOWPROCESSTIME	0x100	Display the user and kernel CPU times on the Web Administrator Statistics screen.

Figure 59: Registry Flag settings

Configuring the OOB Server under Unix

On Unix systems, the OOB Server is by default installed as a service under `inetd`, but it may be configured in its standalone mode.

This simply means that `inetd` does not know about the service and it must be run manually or by some other process.

When the OOB Server is installed as an `inetd`-controlled service, entries are added to the `inetd.conf` and `services` files. Examples of these entries might be

- in `inetd.conf`:

  ```
  esoobserver stream tcp nowait root /bin/sh
  /bin/sh /usr/local/easysoft/oob/server/server
  ```

- and in `services`:

  ```
  esoobserver 8888/tcp # Easysoft OOB Server
  ```

These are just examples and may differ if you picked a different port, service name or installation path.

The mechanics of `inetd` and the server

`inetd` handles incoming network connections and starts the relevant program to service each incoming request.

When `inetd` starts (or is sent a `SIGHUP`) , it reads `/etc/inetd.conf` for a list of what services to supply and to configure its handler for each service.

It then listens on each port used by those services that have been configured in the `/etc/services` file.

When a new connection is requested from outside, `inetd` starts the relevant program and hands over control.

The following example applies to the default installation: when a client connects to port 8888, `inetd` uses the information read from `/etc/inetd.conf` to decide which program to run and with what arguments.

NB	Note that `/bin/sh` is repeated. The first time it appears (in the sixth position on the line) it is the name of the executable to run. The second time it appears it is the value to be passed as the *zeroeth* argument to `/bin/sh`. Normally this is the same as the name of the executable.

In the default installation, the arguments on the line cause `/bin/sh` to run the OOB Server startup script `<InstallDir>/easysoft/oob/server/server`.

The script then sets any necessary environment variables required by the dynamic linker and runs `esoobserver`.

The script passes an argument of `inetd` to the server executable, which notifies the server that it is running under `inetd` rather than at the command line.

The OOB Server inherits the sockets from `inetd` and begins communicating with the OOB Client. `inetd` returns to listening on port 8888 for any new connections.

NB	The Web Administrator feature of the OOB Server does not run when the OOB Server is configured under `inetd`, because `esoobserver` itself is not run until an OOB Client driver connects.

Choosing another Port or Service Name

If you have a port conflict or a service name conflict, or for some other reason you want to change the port or service name of the OOB Server, then you will need to edit the configuration files manually.

The port number is only included in `/etc/services`, but the service name is required in both `/etc/inetd.conf` and `/etc/services`.

After making the necessary amendments you will have to send SIGHUP to the `inetd` process to make it re-read the files. This can be achieved with the command:

```
kill -HUP pid
```

where *pid* is the process ID of `inetd`. You need to be `root` to edit the configuration files and send `inetd` a SIGHUP.

Configuring the Server as Standalone

By default, on Unix the OOB Server is installed as a service with entries in the `/etc/services` and `/etc/inetd.conf` files (`inetd` is informed of the configuration file changes).

However, the OOB Server will also run standalone. When run as standalone, the OOB Server is much faster at accepting connections than the `inetd` started server. So if you are making a lot of small connections to the OOB Server, starting it standalone (without `inetd`) is preferable. Also, the Web Administrator is only available when the OOB Server is running standalone.

Note that the OOB Server no longer uses `inetd` to listen on the socket when running as standalone, so you cannot run `tcp` wrappers. You can use the facilities within the Easysoft ODBC-ODBC Bridge to set up access control.

To start the OOB Server in standalone mode:

1. Comment out the esoobserver entries in the `/etc/inetd.conf` and `/etc/services` files.
2. Force `inetd` to re-read the files using:

   ```
   kill -HUP inetd
   ```

3. Start the server with:

   path/`esoobserver standalone`

   This notifies the server that it must listen on the port itself and fork its own child process when a connection is made (the OOB Server reads the `port` and `httpport` settings in *path/*`oobserver.ini` for its configuration).

NB	You should start the OOB Server as the `root` user otherwise it will be unable to change to the `LogonUser` specified in the OOB Client data source. However, if you do not start the OOB Server as `root` then no logon authentication will be performed and all clients will run on the server as the user who started the OOB Server.

You could add a line to one of your `rc` scripts to have the OOB Server start up as standalone every time the machine boots.

The file you need to edit depends on your operating system so consult your system administrator for further information.

When the OOB Server is started standalone, by default it will fork a process to handle HTTP requests on port 8890 (you can change this in `esoobserver.ini`).

You can then use your browser to go to the Web Administrator (see **"The Web Administrator" on page 158**).

If you want to change configurable parameters or access control via the Web Administrator you must define HTTPAdmin in `esoobserver.ini` to an existing user on your machine, and you will need to enter the password when viewing or making changes on some protected pages.

In this case the OOB Server must be running as `root` on some machines to verify the user name and password.

Note that setting Admin to `"disabled"` will deactivate HTTP Authentication.

Changing Server Configurable Parameters under Unix

Experienced users can configure the OOB Server by editing the file *<InstallPath>*`/easysoft/oob/server/esoobserver.ini`.

If *<InstallPath>* is anything other than `/usr/local/` then there will be a symbolic link `/usr/local/easysoft` to the real `easysoft` directory.

This editing should not be carried out by novice users, so no step-by-step guide is provided here (see **"The Configuration Screen" on page 165** for details of the settings available).

Lines beginning with # are comments. The first non-comment line should be `{Settings}` and the rest of the file should be *Key=Value* pairs.

Note that the bitmask values are given either in decimal or in hexadecimal and the access control lists are separated simply by spaces.

```
{Settings}
Port = 8888
HTTPAdmin = examplea
HTTPPort = 8890
Timeout = 7200
LogDir = /tmp
Logging = 0
MaxBookMarkSize = 32
Flags = 0x26
RetryCount = 4
RetryPause = 3
MaxThreadCount = 0
MaxClientConnect = 0
AllowList =
DenyList =
AllowList = 192.141.32.,192.141.33.
DenyList = 192.141.33.8
```

The `Flags` setting relates to the options which are check boxes on the Web Administrator **Change Configuration** screen (see **"The Configuration Screen" on page 165** for details of these settings).

Set `Flags` by entering the sum of the corresponding bitmask values of your required options:

Value	Name	Hex Value	Description
1	AUTH_DISABLE	0x1	Disable Authentication in the server for a much quicker connection setup.
2	HTTP_SERVER	0x2	Enable the Web Administrator (Windows and Unix)
4	MULTIPROCESS	0x4	Create a new process on connection rather than a thread (Windows NT). Must be set (Unix).
8	NO_METADATA_BLOCKFETCH	0x8	Disable automatic block-fetch mode for metadata calls.
16	HIDESENSITIVE	0x10	Conceal security-sensitive fields.
32	REVERSELOOKUP	0x20	Enable reverse lookup.
64	AUDITODBCACCESS	0x40	Audit all ODBC connections to a log file written to the `LogDir` directory as `esoob_access.log`.

Value	Name	Hex Value	Description
128	FREESTMTSONDISCONNECT	0x80	Free all existing statements in the driver before calling `SQLDisconnect` (a workaround for Navision ODBC driver problems).
256	SHOWPROCESSTIME	0x100	Display the user and kernel CPU times on the Web Administrator Statistics screen.
512	AUTO_UPDATE_CONFIGURATION (Enterprise only)	0x200	Deprecated.
1024	HTTP_AUTH_ALL (Enterprise only)	0x400	see **"HTTP Authentication for all Web Administrator pages" on page 230**.
2048	CONNECTION_POOLING (Enterprise only)	0x800	see **"Connection Pooling" on page 239**.
4096	LOG_FAILING_SQL (Enterprise only)	0x1000	see **"Log of Failing SQL" on page 220**.
8192	ALLOW_DB_BROWSE (Enterprise only)	0x2000	see **"Viewing data sources, tables, columns and data" on page 233**.

Figure 60: Easysoft ODBC-ODBC Bridge Server Flag settings

CHAPTER 5 INTERFACING

Using the Easysoft ODBC-ODBC Bridge with other software

This section provides information about third-party programming languages, tools and applications that can be integrated with the Easysoft ODBC-ODBC Bridge (OOB), some of which have working examples at **http://www.easysoft.com/tech/main.phtml**. Please refer to *<InstallDir>*`/easysoft/oob/doc` for up-to-date documentation, in case these instructions have been superseded.

Chapter Guide

- **Introduction**
- **A Simple OOB Client in C**
- **Apache/PHP**
- **Applixware**
- **ColdFusion**
- **Perl DBI DBD::ODBC**
- **Rexx/SQL**
- **StarOffice**
- **mnoGoSearch (formerly UDMSearch)**
- **mxODBC**
- **SQLPlus_hsODBC**
- **unixODBC**

Introduction

Example programs

The Easysoft ODBC-ODBC Bridge comes with several example programs which are installed into the `examples` subdirectory and can help you to get started writing applications that will load the OOB Client.

The C examples can be compiled and run like any C program, but for the others you need to rebuild the interpreter or compiler with either the Easysoft ODBC-ODBC Bridge or a driver manager (recommended).

Information for Windows developers

Some of the programming languages, tools and applications referred to in this section (and in the files supplied in the `docs` subdirectory) are available on Windows as well as on Unix platforms, and work equally well with the Easysoft ODBC-ODBC Bridge in both environments.

However, the Windows versions of these products are usually pre-built and so you are unlikely to need the information about building or configuring them with the Easysoft ODBC-ODBC Bridge.

Since most of the following information is more likely to be used by the Unix community, all references to directory paths are given in their Unix format, even though the information may also apply to Windows.

A Simple OOB Client in C

In **"The Demo Application" on page 153**, you compiled and/or ran a simple C program that connected through the Easysoft ODBC-ODBC Bridge.

Other small programs are provided in the `examples` directory, and documented in the file `EXAMPLES`.

One of these is `getdata`, a simple C program that reads `stdin` for an SQL statement and passes it to a data source for execution.

NB	The syntax in this example serves as a guide for Unix platforms. Windows developers should refer to the manual for their chosen development platform.

Before you can use the program you must edit the call to `SQLDriverConnect()` to indicate a valid DSN.

You can then build the program by finding a suitable makefile in the `/examples` directory:

1. Change to the examples directory by typing:

   <InstallDir>`/easysoft/oob/examples`

2. Look for a makefile by typing:

   ```
   ls Make*
   ```

3. Edit the makefile appropriate for your platform (e.g. `Makefile.linux`), removing the comment from the second line and making sure that the installation path is correct (for example, `#INSTALLPATH=/usr/local` might become `INSTALLPATH=/opt/`).

   Note that you should remove the "`#`", whether or not you change the directory name.

4. Make the examples:

   ```
   make -f Makefile.linux
   ```

   The `getdata` program, the source to the demo program, and the makefile are enough to show you how to create C programs that connect through the Easysoft ODBC-ODBC Bridge.

   If you need more information, please refer to an ODBC Programmer's Manual.

Apache/PHP

Building Apache with PHP and the Easysoft ODBC-ODBC Bridge allows your web content to be closely integrated with a database running on a machine separate from your web server.

This is particularly useful for offices that have databases that would benefit from up-to-date information from a web server located on a different machine.

For information about building Apache and PHP with the Easysoft ODBC-ODBC Bridge and details of which versions of Apache and PHP the Easysoft ODBC-ODBC Bridge has been proven to work with, please refer to the Apache/PHP document **Apache_PHP** in *<InstallDir>*`/easysoft/oob/doc`.

Also check the INSTALL files in the Apache and PHP distributions, because they may provide more up-to-date information specific to Apache and PHP.

REF	Apache can be found at **http://www.apache.org** and PHP at **http://www.php.net**.

Please note that you cannot build PHP with multiple ODBC drivers or driver managers, so that, for example, using more than one of the following options will cause OOB to fail:

```
--with-custom-odbc

--with-iodbc

--with-esoob

--with-unixODBC

--with-openlink
```

If you want to access multiple ODBC drivers from PHP the best method is to install the unixODBC driver manager and then tell the unixODBC driver manager about your ODBC drivers.

You can build PHP with an ODBC driver and other database drivers such as Oracle or MySQL.

Applixware

Applixware Office is a multi-platform suite of applications.

For information about running Applixware Office with the Easysoft ODBC-ODBC Bridge, and details of which versions of Applixware Office the Easysoft ODBC-ODBC Bridge has been proven to work with, please refer to the Applixware document **Applixware** in *<InstallDir>*`/easysoft/oob/doc`.

REF	For information about Applixware Office, visit **http://www.vistasource.com** or **http://www.applix.com**.

ColdFusion

ColdFusion is a popular web application server and development environment produced by Allaire (**http://www.allaire.com**).

Using the Easysoft ODBC-ODBC Bridge, it is possible to bridge the gap between web servers on non-windows platforms and Microsoft SQL Server, as the Easysoft ODBC-ODBC Bridge interacts with ColdFusion like a standard ODBC driver.

For information about running ColdFusion with the Easysoft ODBC-ODBC Bridge, and details of which versions of ColdFusion the Easysoft ODBC-ODBC Bridge has been proven to work with, please refer to the ColdFusion document **ColdFusion** in *<InstallDir>*`/easysoft/oob/doc`.

Perl DBI DBD::ODBC

DBI is the database interface module for Perl, commonly used for developing CGI scripts on web sites for database driven web pages.

When DBD::ODBC (the database driver for ODBC) is built with the Easysoft ODBC-ODBC Bridge, it enables access to ODBC data sources on other machines.

For information about building Perl DBI with the DBD:ODBC module and the Easysoft ODBC-ODBC Bridge, and details of which versions of Perl DBI the Easysoft ODBC-ODBC Bridge has been proven to work with, please refer to the Perl document **Perl_DBI_DBD_ODBC** in *<InstallDir>*`/easysoft/oob/doc`.

There are also some Perl examples in the *<InstallDir>*`/easysoft/oob/examples` directory.

REF

The Perl home page is at **http://www.perl.com**.

Information about DBI (the database interface module for Perl) can be found at **http://dbi.perl.org**.

Links to other useful sites can be found in the Perl document **Perl_DBI_DBD_ODBC** within *<InstallDir>*`/easysoft/oob/doc`.

Rexx/SQL

Rexx/SQL provides Rexx programmers with a consistent, simple and powerful interface to SQL databases.

For information about building Rexx/SQL with the Easysoft ODBC-ODBC Bridge, and details of which versions of Rexx/SQL the Easysoft ODBC-ODBC Bridge has been proven to work with, please refer to the Rexx/SQL document **Rexx_SQL** in *<InstallDir>*`/easysoft/oob/doc`. Also check the INSTALL files in the Rexx/SQL and Easysoft ODBC-ODBC Bridge distributions because they may provide more up-to-date information.

REF

The Rexx/SQL home page at **http://www.lightlink.com/hessling** contains download links, documentation, mailing list instructions and links to other Rexx resources.

The Rexx/SQL anonymous ftp site is mirrored in the US at **ftp://ftp.lightlink.com/pub/hessling/REXXSQL**.

You will need a Rexx interpreter to run the Rexx/SQL test code and your Rexx programs using Rexx/SQL.

The Rexx interpreter called Regina was used by Easysoft during the testing of the Easysoft ODBC-ODBC Bridge with Rexx/SQL and is available at **ftp://ftp.lightlink.com/pub/hessling/Regina**.

You may use `./configure` (help in Rexx/SQL) to see which Rexx interpreters are supported.

Please ensure that you install a Rexx interpreter before attempting to build Rexx/SQL.

StarOffice

StarOffice is a multi-platform suite of applications.

For information about running StarOffice with the Easysoft ODBC-ODBC Bridge, and details of which versions of StarOffice the Easysoft ODBC-ODBC Bridge has been proven to work with, please refer to the StarOffice document **StarOffice** in *<InstallDir>*`/easysoft/oob/doc`.

REF	For product information about StarOffice, go to **http://www.sun.com/products/staroffice**.

mnoGoSearch (formerly UDMSearch)

mnoGoSearch is a website indexing tool and search engine you can install on your website. Both tools use a relational database backend to store the metadata and execute searches.

Building mnoGoSearch with the Easysoft ODBC-ODBC Bridge allows you to store the mnoGoSearch index information in a remote ODBC data source (e.g. running mnoGoSearch on your Unix web server, but storing the indexed web site information in your Microsoft SQLServer database).

For information about building mnoGoSearch with the Easysoft ODBC-ODBC Bridge, and details of which versions of mnoGoSearch the Easysoft ODBC-ODBC Bridge has been proven to work with, please refer to the mnoGoSearch document **MnoGoSearch** in *<InstallDir>*`/easysoft/oob/doc`.

Also check the INSTALL file in the mnoGoSearch distributions because it provides instructions for building mnoGoSearch with the Easysoft ODBC-ODBC Bridge.

REF	The official homepage for mnoGoSearch with links to the latest downloads, language dictionaries and documentation is at **http://search.mnogo.ru**.

mxODBC

mxODBC is a database API for the Python scripting language which provides an interface to ODBC data sources.

The Easysoft ODBC-ODBC Bridge has been tested with mxODBC-1.0.1 and mxODBC-pre1.0.2 with Python-1.5.1.

For information about running mxODBC with the Easysoft ODBC-ODBC Bridge, and details of which versions of mxODBC the Easysoft ODBC-ODBC Bridge has been proven to work with, please refer to the Perl document **mxODBC** in *<InstallDir>*`/easysoft/oob/doc`.

When building mxODBC, there is a configuration section specifically for the Easysoft ODBC-ODBC Bridge which makes the build very straightforward.

REF	mxODBC and instructions for building it are available at **http://www.lemburg.com/files/python/mxODBC.html**.

SQLPlus_hsODBC

hsODBC (Heterogeneous Services) allows an Oracle listener to connect an ODBC driver.

Once you have configured hsODBC for the Easysoft ODBC-ODBC Bridge to tell it the drivers you have and where to find them, you can use SQLPlus to connect to any remote database for which you have an ODBC driver (e.g. Microsoft SQL Server).

For information about running hsODBC with the Easysoft ODBC-ODBC Bridge and details of which versions of hsODBC the Easysoft ODBC-ODBC Bridge has been proven to work with, please refer to the hsODBC document **SQLPlus_hsODBC** in *<InstallDir>*`/easysoft/oob/doc`.

unixODBC

The Easysoft ODBC-ODBC Bridge does not require a driver manager on the client machine, although there are some advantages in using one, such as:

- being able to pick the ODBC driver from the DSN and have it loaded for you dynamically
- the ability to work with multiple ODBC drivers

There are currently two open source driver managers available for Unix, unixODBC and iODBC. If you want to use a driver manager then Easysoft recommend the unixODBC driver manager.

There are a number of reasons for this:

1. The unixODBC project started by Peter Harvey is now maintained by Nick Gorham who is an Easysoft developer. This means there is much greater experience with unixODBC within Easysoft and we will be able to provide better support for OOB running under unixODBC. It also means that if you find a problem in unixODBC it is much easier for us to facilitate a fix.
2. Easysoft test all their ODBC drivers with unixODBC and although they may work with iODBC it is more difficult for us to support it as we are less familiar with it.
3. The unixODBC package contains much more than a driver manager. The aim of the unixODBC project is to provide all the ODBC functionality available on Windows for UNIX operating systems. The unixODBC package may be built with the QT libraries to allow GUI configuration of DSNs and drivers. It also contains the GUI DataManager program which may be used to explore your ODBC data. OOB contains the code and shared object which is used by

unixODBC's GUI ODBCConfig utility to add/delete and configure OOB DSNs.

4. From OOB 0.4.0.0, the OOB installation for UNIX can automatically install itself using the unixODBC installer program to run under unixODBC.
5. From OOB 1.0.0.6, the OOB installation can automatically configure a demo datasource in unixODBC for OOB.

For the latest information about unixODBC, please refer to the unixODBC document in *<InstallDir>*`/easysoft/oob/doc` in case any of the information here has been superseded.

REF	For more information on unixODBC see **http://www.unixodbc.org**.

CHAPTER 6 ENTERPRISE

The Easysoft ODBC-ODBC Bridge Enterprise Edition

The Easysoft ODBC-ODBC Bridge Enterprise Edition is intended for use in enterprise critical database applications with high transaction rates in organisations that cannot afford to let their data be unavailable at any time.

Building upon the standard version of the Easysoft ODBC-ODBC Bridge, the Enterprise edition includes additional features to provide additional protection from unauthorised access through enhanced access control, flexible configuration options and a suite of performance management reports and statistics to allow you to monitor and manage your system.

Chapter Guide

- **Introduction**
- **Reporting and Statistics**
- **Auditing**
- **Security**
- **Usability**
- **Availability**
- **Feature table**

Introduction

The standard Easysoft ODBC-ODBC Bridge (OOB) is installed on many customer sites world-wide, providing access to remote ODBC data sources.

By far the most popular configuration is a Unix web server (such as Apache) running PHP or Perl CGI to bring remote ODBC data to customer web sites.

In fact, the Easysoft web site itself uses the Easysoft ODBC-ODBC Bridge to store and retrieve information from our remote database to:

- register new customers
- deliver FAQs
- register trials, quotes and purchases and other shop activities
- deliver news
- provide details and links to our product downloads
- deliver Easysoft’s cinema.com web site at (**http://www.cinema.com**)

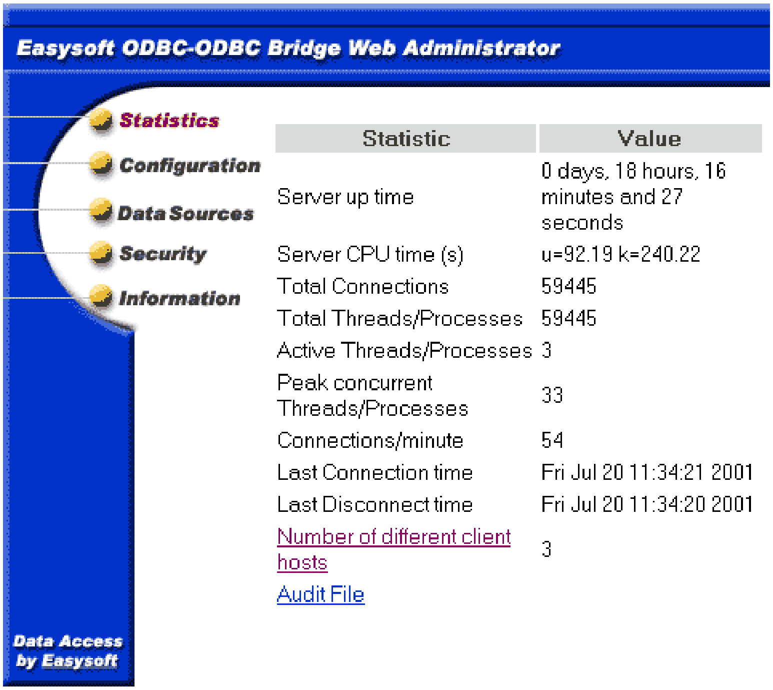
When using the standard Easysoft ODBC-ODBC Bridge, Easysoft recognised a requirement for an Enterprise Edition providing the following enhanced services for enterprise customers:

- extended reporting and statistics
- improved auditing
- greater security
- better usability
- higher availability

Reporting and Statistics

One of the the most frequent requests that Easysoft receive from Enterprise customers is for greater reporting and statistics, so that they can see exactly what the OOB Server is doing at any time.

The Web Administrator **Statistics** screen displays a number of statistics detailing server up time, total connections, total successful connections, active threads/processes, peak concurrent threads/processes and the time of the last connect or disconnect:



Statistic	Value
Server up time	0 days, 18 hours, 16 minutes and 27 seconds
Server CPU time (s)	u=92.19 k=240.22
Total Connections	59445
Total Threads/Processes	59445
Active Threads/Processes	3
Peak concurrent Threads/Processes	33
Connections/minute	54
Last Connection time	Fri Jul 20 11:34:21 2001
Last Disconnect time	Fri Jul 20 11:34:20 2001
Number of different client hosts	3
Audit File	

Figure 61: General data on the Enterprise Web Administrator Statistics screen

In the Easysoft ODBC-ODBC Bridge Enterprise Edition the following statistics have been added:

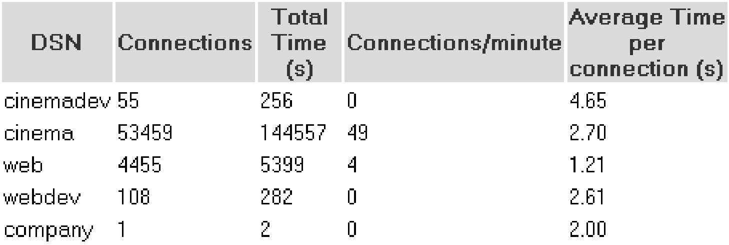
Average Connections per minute (server)

This is the server up time (in minutes) divided by the total number of successful connections. This is a very useful measure of how busy the OOB Server itself is.

However, if you are using persistent connections from your client application you should realise that this number is likely to always be low, since connections are being reused by your client.

DSN statistics

The Easysoft ODBC-ODBC Bridge Enterprise Edition records each DSN to which it is connected and displays a table of DSN statistics:



DSN	Connections	Total Time (s)	Connections/minute	Average Time per connection (s)
cinemadev	55	256	0	4.65
cinema	53459	144557	49	2.70
web	4455	5399	4	1.21
webdev	108	282	0	2.61
company	1	2	0	2.00

Figure 62: DSN data on the Enterprise Web Administrator Statistics screen

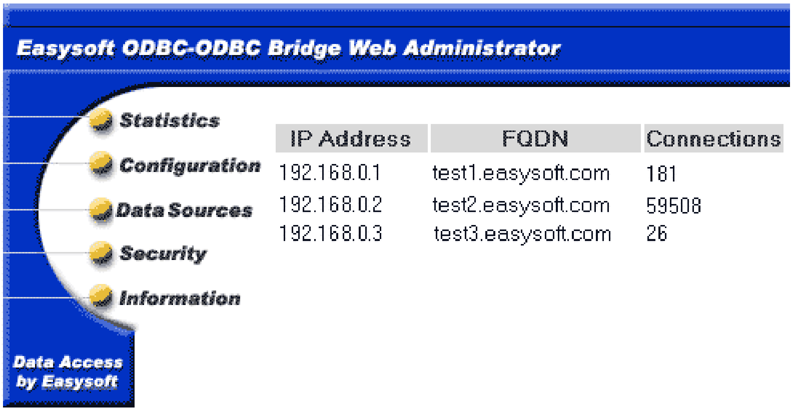
The information shown in the table per DSN is:

- **Number of connections to this DSN -** By comparing this number for each DSN you can see which is the busiest data source on your server.

- **Total time connected to this DSN -** This is concurrent time so two concurrent connections, connected for the same 5 seconds registers as 10 seconds total time.
- **Connections per minute to this DSN -** The server up time (in minutes) divided by the number of connections to this DSN.
- **Average time per connection -** The total time connected to this DSN divided by the number of connections to this DSN. If you know your client applications connect, issue some SQL and then disconnect, this is a good measure of how much processing there is for each connection. This number is likely to be higher for client applications using persistent connections, as there may be long periods of connected time where nothing is happening.

Host Statistics

From the Web Administrator **Statistics** screen there is a link called **Number of different client hosts** which takes you to a table of statistics per connecting client:



IP Address	FQDN	Connections
192.168.0.1	test1.easysoft.com	181
192.168.0.2	test2.easysoft.com	59508
192.168.0.3	test3.easysoft.com	26

Figure 63: The Enterprise Web Administrator Client Hosts screen

Information is per client and shows:

- **IP Address -** The IP address of the client.
- **FQDN -** The Fully Qualified Domain Name of connecting clients. This is displayed as "`unknown`" unless `ReverseLookup` is enabled (see **"ReverseLookup" on page 172**).
- **Connections -** The number of connections per client. If you have more than one connecting client then this allows you to see which client is putting the greatest load on your server.

TERMINATION STATISTICS

The contents of the Web Administrator **Statistics** screen are written on exit to a file called `esoob_stats_<dd>_<mm>_<yyyy>_<hhmmss>.stats` where `<dd>` is the day number, `<mm>` is the month number, `<yyyy>` is the year number and `<hhmmss>` is the time in hours, minutes and seconds.

This file will be written into the directory defined by the server configurable parameter `LogDir`:

```
Server up time 0 days, 18 hours, 5 minutes and 23 seconds

Server CPU time(s):                 u=92.66 k=242.72

Total Connections:                  58966

Total Threads/Processes:            58966

Active Threads/Processes:           4

Peak concurrent Threads/Processes:  74

Connections/minute:                 54
```

```
Last Connection time:              Wed Jul 04 08:55:13 2001
Last Disconnect time:              Wed Jul 04 08:55:12 2001

DSN  Connections Total   Connections Average
DSN              Time(s) per minute  Time per
DSN                                  connection(s)


dsn1 3217        4153    2           1.29
dsn2 55566       212747  51          3.83
dsn3 42          8060    0           191.90
dsn4 111         101     0           0.91

Number of different client hosts: 3
192.168.0.1 (test1.easysoft.com)
192.168.0.2 (test2.easysoft.com)
192.168.0.3 (test3.easysoft.com)
```

Average Connections per minute (data source)

This is the Server up time (in minutes) divided by `Total Threads/Processes`.

The average connections per minute includes connections that were refused by the server for various reasons e.g. access control or authentication.

Log of Failing SQL

The OOB Server can keep a log of any failed `SQLPrepare`, `SQLExecute` or `SQLExecDirect` functions.

To enable this feature check the `LogFailingSQL` parameter on the Web Administrator **Configuration** screen (see **"LogFailingSQL" on page 174**), which will create a file called `oob_sql.log` when the first failing SQL occurs.

This file is located in the directory defined by the `LogDir` parameter on the Web Administrator **Configuration** screen (see **"LogDir" on page 169**).

Any future SQL commands that fail are appended to the file once it has been created and you can rename or delete this file at any time.

The failing SQL log file has two types of entry:

[a] an entry showing a piece of SQL which failed:

```
date time "DSN" SQL text
```

e.g.

```
12-04-2002 10:32:55 "test" insert into mytable values (1,2)

---------- -------- ------ -------------------------------

   date          time      DSN name          the SQL text which failed
```

An error may be for a number of reasons, such as the SQL failing to parse correctly, or the database engine being unable to comply (a duplicate key error, for example):

```
31-07-2001 11:24:24 "example" update Title set Count=Count+1
where id = 1

01-08-2001 09:08:08 "web" select columnnoexist from stores
```

```
01-08-2001 09:05:05 "test" update sales set stor_id = 8042
where stor_id = "fred"

01-08-2001 09:06:06 "test" select * from tablenoexist
```

The DSN name field is extremely useful when identifying problems.

In this example the "`web`" and "`test`" DSNs both point at the same database, but "`web`" is the live version of the application and "`test`" is used by software developers.

As the client applications are on different machines, the `TargetDSN` used by the OOB Client is set to "`web`" on one client and to "`test`" on the other.

Failing SQL in the log to "`test`" would not be uncommon as this is under development, but failures for "`web`" indicate a problem a real user might have experienced running live software.

[b] an entry logging the first database error retrieved AFTER a piece of SQL failed:

```
date time SQLState="xxxxx", NativeError=nnnn, Msg="xxxxxxxx"
```

e.g.

```
12-04-2002 10:32:55 SQLState="23000", NativeError=2627
Msg="[Microsoft][ODBC SQL Server Driver][SQL Server]
Violation of PRIMARY KEY constraint"
```

The failing SQL file is written to as follows:

1. each piece of SQL executed with either `SQLPrepare` , `SQLExecute` or `SQLExecDirect` is documented.
2. an entry is written in the format in [a] if an `SQLPrepare` , `SQLExecute` or `SQLExecDirect` function fails.

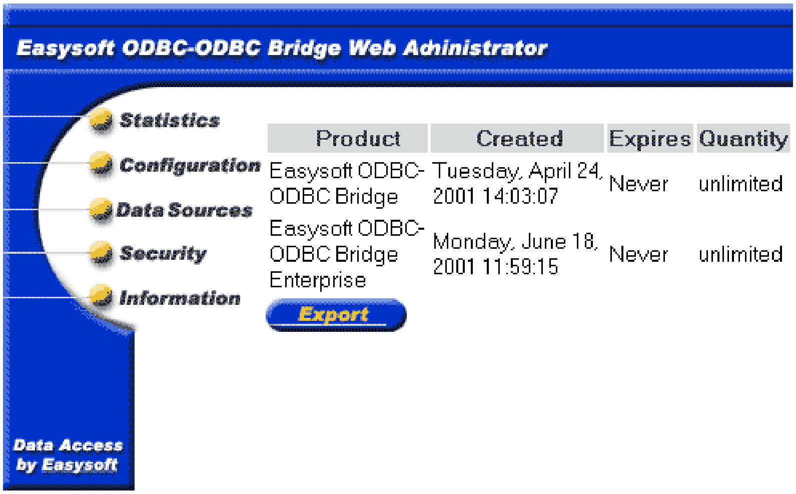
3. if [2] occurs then an entry is written in the format in [a] if the application calls `SQLError` or `SQLGetDiagRec`.
4. once the application is finished with the SQL in [1] (or executes more SQL) then the previous SQL is no longer available to document.

   This means that failing SQL is written to one line of the failing SQL log, followed by the error text returned from the database engine on the next line.

   However, this is NOT always the case, as the failing SQL statements and the error text can get mixed up if the server is very busy (for example, if multiple concurrent connections all fail at the same time).

Viewable Licenses

To view all your installed Easysoft licenses, click **Information** on the main web page followed by the **Installed Licenses** link:



Product	Created	Expires	Quantity
Easysoft ODBC-ODBC Bridge	Tuesday, April 24, 2001 14:03:07	Never	unlimited
Easysoft ODBC-ODBC Bridge Enterprise	Monday, June 18, 2001 11:59:15	Never	unlimited

Figure 64: The Enterprise Web Administrator Licenses screen

The table shows four columns:

- **Product -** The name of the product this license applies to.
- **Created -** The date the license was installed.
- **Expires -** The date the license expires or the string "`Never`", meaning the license does not expire.
- **Quantity -** A product specific limit which is either the maximum allowable number of concurrent connections or the string "`unlimited`", meaning that there is no limit on the number of concurrent connections.

Auditing

The OOB Server records all activity to an audit trail file if the `AuditODBCAccess` parameter is checked (see **"Server Configurable Parameters in the Registry" on page 185** on Windows or **"Changing Server Configurable Parameters under Unix" on page 192**) on Unix).

The **Audit File** link on the Web Administrator **Statistics** screen display a page showing:

- the first page from the current audit trail file.
- links for the current audit trail file to go a page forward or a page backward, and to the first or last page.
- a link to each recorded audit trail file.
- two graph links for each audit trail file, one showing connections per minute and another connections per hour over the period of that audit trail file (a new file is opened each day), which show when the OOB Server is busiest (so that better decisions can be made for scheduling administration tasks that require taking the server down, for example) and highlight unexpected bursts in activity (e.g. a search robot cataloging your web site).

AUDIT FILE DESCRIPTION

The current Audit File is called `esoob_access.log` and will be placed in the directory specified by the server configurable parameter `LogDir`.

Here is an example of a few lines from the audit file:

```
Mon, 02 Jul 2001 15:01:07 GMT CONNECT
192.168.0.1(test1.easysoft.com)

Mon, 02 Jul 2001 15:01:12 GMT DISCONNECT
192.168.0.1(test1.easysoft.com)

Fri, 06 Jul 2001 13:10:46 GMT AUTH_DENIED
192.168.0.1(test1.easysoft.com)

Fri, 06 Jul 2001 14:34:36 GMT REFUSED_BY_DSN_RULE
test(Martin Evans)

Fri, 06 Jul 2001 15:18:34 GMT REFUSED_MAXCLIENTCONNECT
192.168.0.3(unknown)

Fri, 06 Jul 2001 15:19:33 GMT REFUSED_MAXCONCURRENT
192.168.0.2(test2.easysoft.com)

Mon, 09 Jul 2001 13:04:02 GMT REFUSED_BY_RULE
192.168.0.1(test1.easysoft.com)
```

The fields in each line of the audit file are:

- day - the day of the week
- day number - the day of the month
- month - the month name
- year - the year
- time - hours, minutes and seconds (colon separated)
- timezone - the time zone

- event - an event code (see **"Graph Generation" on page 226**)
- event arguments - details of the client, DSN or user name (see **"Graph Generation" on page 226**)

The events and their descriptions are:

- **CONNECT -** An OOB Client has connected to the OOB Server. This does not mean the client gained successful access to the ODBC data source, but that the server accepted the connection. If the OOB Server `Authentication_Disabled` parameter was not set (see **"The Configuration Screen" on page 165**) then the client was authenticated by the operating system and also completed any access control test based on its IP Address. The last field of the audit file shows the client IP address and also the client FQDN in brackets if the `ReverseLookup` parameter is enabled (see **"ReverseLookup" on page 172**).
- **DISCONNECT -** An OOB Client disconnected from the OOB Server. The audit file does not indicate why the client disconnected. The client may have called `SQLDisconnect`, simply been interrupted, or the OOB Server could have timed out the connection because it was inactive for the number of seconds set in the `TimeOut` parameter (see **"The Configuration Screen" on page 165**).
- **AUTH_DENIED -** This event happens when an OOB Client connects to the OOB Server, OOB Server authentication is enabled and the client passes an invalid user name/password (`LogonUser/LogonAuth`) pair for the operating system. The OOB Server has turned down the connection request.

- **REFUSED_MAXCLIENTCONNECT -** The connection attempt has been turned down. The OOB Client attempting connection to the OOB Server already has too many open connections. The OOB Server limits the connections from a particular client to `MaxClientConnect` (see **"The Configuration Screen" on page 165**).
- **REFUSED_MAXCONCURRENT -** The connection attempt has been turned down. There are already too many open connections as defined by `MaxThreadCount` (see **"The Configuration Screen" on page 165**).
- **REFUSED_BY_RULE -** The connection attempt has been turned down due to an client access control rule defined in the server on the Web Administrator **Security** screen.
- **REFUSED_BY_DSN_RULE -** The connection attempt has been turned down due to a DSN access control rule. These rules are defined in the server on the Web Administrator **Security** screen.

Daily Renaming

The audit files are renamed once a day at midnight. At midnight the current audit trail file (`esoob_access.log`) is renamed to `esoob_access_<dd>_<mm>_<yyy>.log` where <dd> is the day number, <mm> is the month number and <yyyy> is the year number.

The audit file for each day should be visible in the Web Administrator.

Graph Generation

The Web Administrator is capable of analysing a particular audit file and producing graphs of connections per minute and connections per hour for any selected day.

These graphs are accessed from the **Audit File** link on the Web Administrator **Statistics** screen.

Connections are displayed as blue bars and warning events are displayed as coloured warning bars on top of the relevant blue bar:

failed connects (Purple bar):

- AUTH_DENIED

attempts denied access due to access control lists of server limits (Red bar):

- REFUSED_MAXCLIENTCONNECT
- REFUSED_MAXCONCURRENT

authentication failures (Orange bar):

- REFUSED_BY_RULE
- REFUSED_BY_DSN_RULE

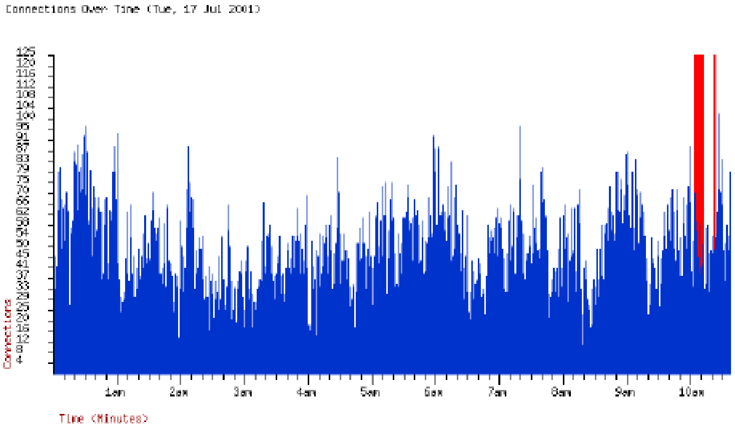


Figure 65: The Enterprise Web Administrator Connections per Minute graph

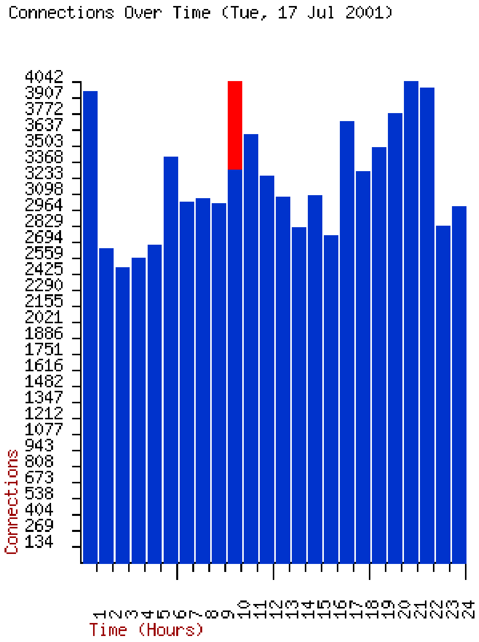


Figure 66: The Enterprise Web Administrator Connections per Hour graph

Security

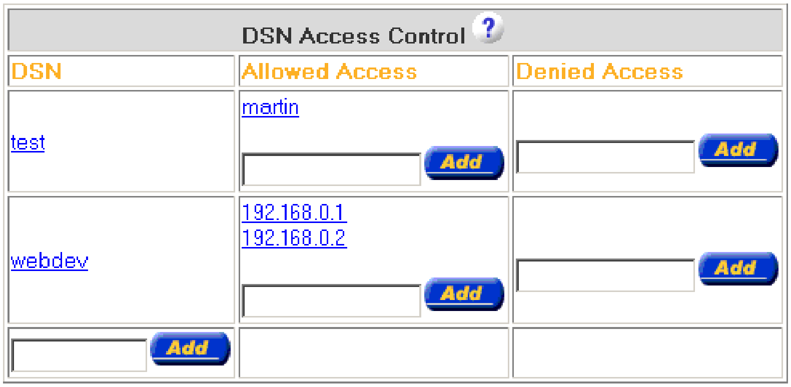
Access Control Per DSN

The standard Easysoft ODBC-ODBC Bridge provides two methods of accessing data sources:

- Authentication of clients who must provide a valid user name/password for the host operating system
- Access control by IP address

The Easysoft ODBC-ODBC Bridge Enterprise Edition provides the additional facility to build access control lists to give you even more control over specific data sources.

With DSN access control lists you can define exactly which machines and users access which DSNs via the **Security** screen on the Web Administrator:



DSN Access Control ?

DSN	Allowed Access	Denied Access
test	martin	
webdev	192.168.0.1 192.168.0.2	

Figure 67: The Enterprise Web Administrator Access Control table

The DSN `Allowed Access` list and `Denied Access` list define the hosts and users which may access a particular DSN.

The server consults the `Deny Access` list first, and access is denied if a client address/user name matches an entry in this list.

If there is no match for the client/user name in the `Denied Access` list then the Server consults the `Allowed Access` list, and if the `Allowed Access` list is empty the client/user is allowed access.

If there are any entries in the `Allowed Access` list then the client/user is only allowed access if it matches an entry in this list.

The entries in either list must consist of either IP addresses (e.g. `192.168.0.0`) or user names. Part of an address may be specified (e.g. `192.168.0`, which will match any machine on the `192.168.0` network).

Any user name added to either list must match the `LogonUser` attribute used by the OOB Client to log the OOB Server thread/process into that account before allowing access to a DSN.

NB	As a special case (to avoid having to name every user when adding an IP address to the DSN allow list) the deny list is checked for the current user, and allows access to the DSN if no matching entry is found.

HTTP Authentication for all Web Administrator pages

In the standard Easysoft ODBC-ODBC Bridge only a few web pages in the Web Administrator are protected with HTTP authentication.

Pages protected with HTTP authentication require the user to enter the `HTTPAdmin` user name and their password before the page is displayed.

The Easysoft ODBC-ODBC Bridge Enterprise Edition has a new OOB Server configurable parameter called `HTTPAuthAll` which forces all web pages to use HTTP authentication when it is enabled.

You can find this configuration option off the **Configuration** link on the main Web Administrator page.

Securing the Web Administrator from clients outside your network

The `HTTPNetworkAccess` configurable parameter can be used to restrict access to the Web Administrator by setting it to the network or IP address of clients that are to be allowed access.

For example, `192.168.0.1` only allows the machine with IP address `192.168.0.1` to use the Web Administrator, whereas `192.168.0` would allow access to anyone on that network.

Browser clients not permitted access will receive an "`HTTP 403 Forbidden`" error. By default, all browser clients have access to the Web Administrator, although they may be required to enter a user name/password if `HTTPAdmin` is set to anything other than "`disabled`".

PREVENTING DATA SOURCES BEING BROWSED IN THE ADMINISTRATOR

The Easysoft ODBC-ODBC Bridge Enterprise Edition is not only capable of displaying a list of system data sources, but can also allow those data sources to be browsed (see **"Viewing data sources, tables, columns and data" on page 233**).

This is a powerful, but potentially dangerous feature, as anyone with a browser and access to the Web Administrator might be able to see data in your database.

Although you can protect all of the Web Administrator pages with HTTP authentication, the `AllowDBBrowse` parameter allows the user to define a more precise configuration.

By default `AllowDBBrowse` is unset, which means that data sources shown on the **Data Sources** page are listed but not browsable. If `AllowDBBrowse` is turned on then the data sources become clickable links allowing you to browse down through the DSNs, tables, columns and data.

Usability

VIEWING AND EXPORTING LICENSES

The Web Administrator can display a list of installed licenses.

From the main page click on **Information** and then **Installed Licenses** (see **"Viewable Licenses" on page 222**).

On Windows the **Information** page also allows you to export the registry entries in RegEdit 4 format to a file called LogDir\easysoft_licenses.reg where LogDir is an OOB Server configurable parameter defined on the **Configuration** screen.

Click the **Export** button to export your licenses. If your registry becomes damaged you can restore the license entries by double clicking on the easysoft_licenses.reg file.

Exporting Server Configurations

OOB Server configuration is stored in the registry (in Windows) or in a plain text file (on Unix systems).

It is simple to safeguard this configuration or copy it to another OOB Server on Unix, but normally more difficult on Windows.

In Windows, at the bottom of the **Configuration** screen in the Web Administrator there is an **Export** button.

Clicking **Export** writes the entire OOB Server configuration out to the file LogDir\oob.reg where LogDir is an OOB Server configurable parameter defined on the **Configuration** screen.

This file is in RegEdit 4 format, so it can be copied to other servers where double-clicking on it will install that OOB Server configuration into the registry.

Support for multiple network interface cards

By default, the OOB Server will start to listen on port 8888 on all network interfaces, but if the server has network interface cards (NICs) on multiple networks the OOB Server can be associated with a particular NIC by specifying an IP address.

This will be necessary if the server is running on multiple networks and access to the OOB Server is only required from one network.

The `IPAddress` configurable parameter on the Web Administrator **Configuration** page is usually an empty field, signifying that the OOB Server is listening on all local interfaces.

Change this field to the IP address of a particular NIC to restrict access to a particular network.

VIEWING DATA SOURCES, TABLES, COLUMNS AND DATA

The standard Easysoft ODBC-ODBC Bridge opens up access to remote databases.

However, this main advantage is also an inconvenience when writing your client application without a means of viewing your database locally. Often you need to look up table names, columns and even sample data in the process of writing your client application.

The Easysoft ODBC-ODBC Bridge Enerprise Edition includes an additional Web Administrator facility that allows data sources to be browsed for lists of tables, table composition and rows of data in the tables.

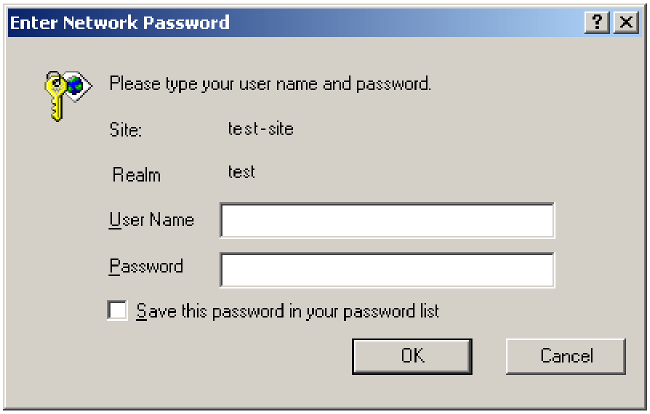
In order to browse system data sources on the machine where the OOB Server is running you must enable the `AllowDBBrowse` configurable parameter on the Web Administrator **Configuration** screen.

This converts the list of data source names on the `Data Sources` page into links to further pages, allowing you to browse down through the data source and its tables, columns and data.

You may be prompted for a database user name and password when first clicking on a DSN.

The OOB Server attempts an initial connection without the ODBC UID/PWD attributes, but if that fails with an authentication error (`28000`) an authentication challenge will be issued, where the realm is the name of the DSN.

NB	DSNs cannot be browsed using trusted connections.



Enter Network Password

Please type your user name and password.

Site: test-site

Realm test

User Name

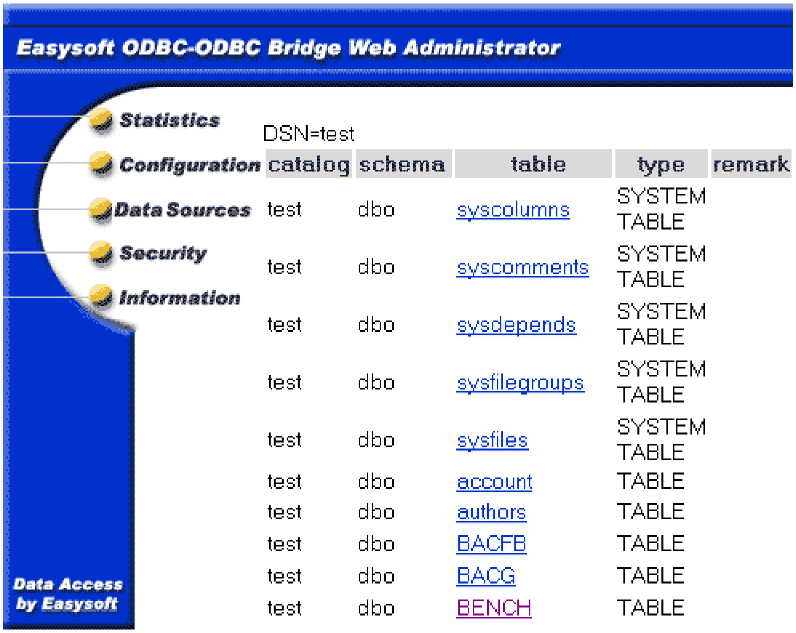
Password

□ Save this password in your password list

Figure 68: The Enterprise DSN Realm Enter Network Password dialog box

Having gained access, DSN details can then be displayed, as in the following example:

The initial screen displays the tables and views in the DSN `test` (i.e. it is the result set generated by calling `SQLTables` when connected to a DSN called `test`):

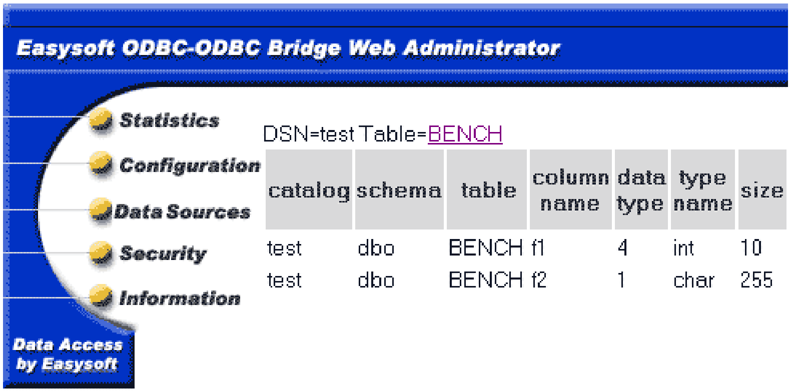


DSN=test

catalog	schema	table	type	remark
test	dbo	syscolumns	SYSTEM TABLE	
test	dbo	syscomments	SYSTEM TABLE	
test	dbo	sysdepends	SYSTEM TABLE	
test	dbo	sysfilegroups	SYSTEM TABLE	
test	dbo	sysfiles	SYSTEM TABLE	
test	dbo	account	TABLE	
test	dbo	authors	TABLE	
test	dbo	BACFB	TABLE	
test	dbo	BACG	TABLE	
test	dbo	BENCH	TABLE	

Figure 69: The Enterprise Web Administrator Data Sources screen DSNs

Click on **BENCH** to display a list of the columns of data in that table (generated by calling `SQLColumns` on table `BENCH` in DSN `test`):

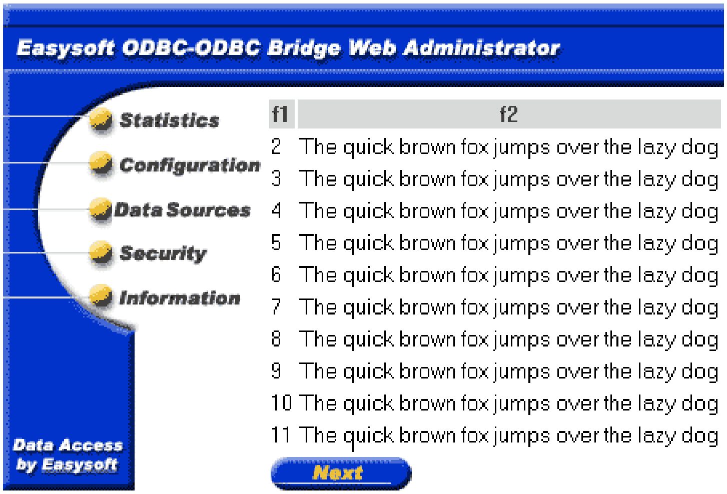


DSN=test Table=BENCH

catalog	schema	table	column name	data type	type name	size
test	dbo	BENCH	f1	4	int	10
test	dbo	BENCH	f2	1	char	255

Figure 70: The Enterprise Web Administrator Data Sources screen DSN data

Click on **BENCH** to display the first ten rows in that table (generated by a `"select * from BENCH"` statement):



f1	f2
2	The quick brown fox jumps over the lazy dog
3	The quick brown fox jumps over the lazy dog
4	The quick brown fox jumps over the lazy dog
5	The quick brown fox jumps over the lazy dog
6	The quick brown fox jumps over the lazy dog
7	The quick brown fox jumps over the lazy dog
8	The quick brown fox jumps over the lazy dog
9	The quick brown fox jumps over the lazy dog
10	The quick brown fox jumps over the lazy dog
11	The quick brown fox jumps over the lazy dog

Figure 71: The Enterprise Web Administrator Data Sources screen row data

Click **Next** or **Prev** to move forwards or backwards in the result set by ten rows at a time.

Availability

Client support for Fallback OOB Servers

Usually the OOB Client is passed a data source name (DSN) which defines a single remote server with which to connect.

If the client fails to connect to the specified server then it tries 4 more times by default, pausing (`0.1 * connection atttempts`) seconds between each attempt (the number of additional attempts can be configured with the parameter `NetConnectRetryCount`).

With the Easysoft ODBC-ODBC Bridge Enterprise Edition each client DSN can define multiple servers and ports and if the client fails to connect to the first server in the list it retries with the each server in the list until either a connection is made or the list is exhausted.

When specifying multiple servers in a DSN there must be the same number of ports in the `Port` list as servers in the `Server` list and each port and server must be separated with a comma.

e.g.

```
[data_source_name]

TargetDSN = demo

LogonUser = user name

LogonAuth = password

Transport = tcpip

Server = demo.easysoft.com,fallback.easysoft.com

Port = 8888,8888
```

In this example the OOB Client will attempt a connection to port 8888 on demo.easysoft.com. If this fails and after NetConnectRetryCount additional attempts a connection has still not been made then the client will try to connect to the backup server fallback.easysoft.com.

There may be an advantage in lowering NetConnectRetryCount from the default of 4 if multiple server definitions are being used in a single DSN, as this will reduce the time taken from the client failing to connect to the first server and starting a new connection attempt to the second and subsequent servers.

Specific support for multiple processors (Process Affinity)

When the standard OOB Server runs on multiple processor machines, it is left entirely to the operating system as to which CPUs are utilised.

When the Easysoft ODBC-ODBC Bridge Enterprise Edition is running on Windows NT or 2000, you can allocate particular processors on which to run the OOB Server. This is known as process affinity.

If your server has multiple CPUs then the Web Administrator **Configuration** screen will show each configured processor as CPU_n (where n is a number between zero and the number of configured processors minus one).

Initially, the OOB Server starts with its process affinity set to the default (available to run on all processors). However, unchecking the CPU_n check boxes on the change configuration page changes the OOB Server affinity so that it is restricted to a subset of the configured processors.

Note that the OOB Server only sets process affinity when starting up, so any changes made will only affect the server after it has been restarted.

Connection Pooling

The Easysoft ODBC-ODBC Bridge Enterprise Edition introduces automatic connection pooling in the OOB Server. Connection pooling in the OOB Server process may be enabled from the Web Administrator **Configuration** page (see **"ConnectionPooling" on page 174**).

If connection pooling is enabled then the OOB Server sets the `SQL_ATTR_CONNECTION_POOLING` attribute to `SQL_CP_ONE_PER_DRIVER` to enable connection pooling in the driver manager for the ODBC driver.

If connection pooling is enabled then connections through the ODBC driver to the database are held open by the driver manager for use the next time the application uses the same DSN.

This is only of any real benefit if the OOB Server is running in multi-threaded mode (the default), as the pooled connections are per process.

Enabling connection pooling in the OOB Server can vastly reduce connection times, especially through ODBC drivers connecting to remote databases.

Feature table

Feature	Standard	Enterprise
General		
ODBC 3.5 support.	Yes	Yes
ODBC 2.0 support.	Yes	Yes
Full integration with Microsoft's ODBC driver manager and the **unixODBC** driver manager.	Yes	Yes
Integral Web Administrator enabling remote administration, configuration and status reporting.	Yes	Yes
Support for thread-unsafe and threaded ODBC drivers and applications.	Yes	Yes
TCP/IP support.	Yes	Yes
Data and ODBC optimisations		
Data compression.	Yes	Yes
Block Fetch optimisations.	Yes	Yes
Column-wise and row-wise binding optimisations.	Yes	Yes
Security		
Native operating system user name/Password protection of OOB Server access.	Yes	Yes
Encryption of connection strings, user names and passwords.	Yes	Yes
Native operating system user name and password protection of configuration changes via the web interface.	Yes	Yes

Feature	Standard	Enterprise
Access Control List by client IP address protecting connections to the OOB Server.	Yes	Yes
Access Control List protection of data sources by client IP address and user name.	No	Yes
Native operating system user name and password protection of all web interface pages.	No	Yes
Restrict Web Administrator access to a machine or network.	No	Yes
Reporting/Statistics		
Server uptime, CPU usage, total connections, total successful connections, active connections, peak concurrent connections and last connect/disconnect time.	Yes	Yes
List of connecting clients.	Yes	Yes
Auditing of all connection activity to a file.	Yes	Yes
Average connections per minute.	No	Yes
Connections per client.	No	Yes
Statistics written to time stamped statistics file on exit.	No	Yes
Automatic audit trail file rotation.	No	Yes
Audit trail file viewable from remote browser.	No	Yes
Graphs of connection history and auditable events viewable from remote browser.	No	Yes
Data source connection statistics including connections per DSN, time connected per DSN and average connected time per DSN.	No	Yes
Licenses viewable from remote browser.	No	Yes
Log of all SQL executed which fails.	No	Yes

Feature	Standard	Enterprise
Usability		
List of configured System DSNs on Server viewable from remote browser.	Yes	Yes
Support for multiple Network Interface Cards.	No	Yes
Licenses may be exported for backing up or replication.	No	Yes
OOB Server configuration can be exported for backing up or importing into another OOB Server.	No	Yes
Navigation through DSNs from a remote browser can display all tables, columns in each table and even rows from in each table (useful when writing your client application on a remote machine).	No	Yes
Availability		
Client supports connection to fallback OOB Server if primary OOB Server is off line.	No	Yes
Process affinity: the OOB Server process may be tied to a specific CPU on a machine that has multiple CPUs.	No	Yes
Connection pooling implemented in the OOB Server.	No	Yes

Figure 72: A Standard/Enterprise Easysoft ODBC-ODBC Bridge Feature table

APPENDIX A TECHNICAL REFERENCE

Additional information for the Easysoft ODBC-ODBC Bridge

This section contains extra information relating to the deployment of the Easysoft ODBC-ODBC Bridge (OOB).

It documents where the Easysoft ODBC-ODBC Bridge API differs from other ODBC APIs, diagnostic functionality and tracing issues.

Appendix Guide

- **ODBC versions supported**
- **Unsupported ODBC 3.5 functionality**
- **Modifications to the API**
- **Understanding ODBC diagnostic messages**
- **Implementing ODBC diagnostics**
- **Tracing**

ODBC versions supported

The Easysoft ODBC-ODBC Bridge supports most of ODBC 2.0 and ODBC 2.5, and all of ODBC 3.0 and 3.5 with the exception of `SQL_IS_POINTER` (see **"Unsupported ODBC 3.5 functionality" on page 244**).

All the ODBC 2.0 and 2.5 API functions required to run Perl DBD:ODBC, PHP and mxODBC are present, but the full API of these deprecated versions is not supported, as new applications will be written using the ODBC 3.5 API.

Unsupported ODBC 3.5 functionality

SQL_IS_POINTER

The use of driver-specific pointer types (`SQL_IS_POINTER`) is not supported in calls to:

- `SQLSetConnectAttr()`
- `SQLSetStmtAttr()`
- `SQLSetDescField()`

Modifications to the API

SQLDRIVERS()

On non-Windows platforms `SQLDrivers()` is implemented by the Easysoft ODBC-ODBC Bridge, even though it is a driver manager function. This is primarily for Perl DBD::ODBC 0.21 support. The Easysoft ODBC-ODBC Bridge only returns details about itself and not any other drivers.

SQLBrowseConnect()

The semantics of `SQLBrowseConnect()` are slightly difficult in the context of the OOB. Normally, `SQLBrowseConnect()` provides an iterative method of discovering and enumerating the attributes and attribute values required to connect to a data source. Each call to `SQLBrowseConnect()` returns successive levels of attributes and attribute values. When all levels have been enumerated a connection to the data source is completed and a complete connection string is returned by `SQLBrowseConnect()`.

This process works fine when `SQLBrowseConnect()` is called for a driver on the local machine, but when you introduce a bridge there are in effect two levels of browsing. The first level will browse the local data sources (defined in your `odbc.ini` file or in the registry), but these point to another real data source on a remote machine.

`SQLBrowseConnect()` only supports the browsing of Easysoft ODBC-ODBC Bridge data sources on the local machine. If the user tries to connect to one of these then they will be prompted for Easysoft ODBC-ODBC Bridge attributes such as `SERVER`, `TRANSPORT`, `PORT`, `LOGINUSER`, `LOGINAUTH`, `TARGETUSER` and `TARGETAUTH`.

Once sufficient attributes have been defined to allow a bridge connection to the server, the server side of the Easysoft ODBC-ODBC Bridge will return a list of DSNs retrieved by calling `SQLDataSources()`. The browse stops here and so the remote data source must already be set up with sufficient information to allow a connection.

This implementation avoids complications with possible clashes of attributes between the Easysoft ODBC-ODBC Bridge and the remote ODBC driver which would make it impossible to return a final connection string which allows a later connection without browsing.

Understanding ODBC diagnostic messages

The Easysoft ODBC-ODBC Bridge works in conjunction with a number of other types of software, such as ODBC applications, driver managers, drivers and DBMSs.

If a diagnostic (or *error*) message is displayed when a user connects via the Easysoft ODBC-ODBC Bridge, the error does not necessarily lie in the Easysoft ODBC-ODBC Bridge component of your configuration and the message should usually indicate where the problem lies.

For example, examine the following diagnostic messages:

```
[Easysoft ODBC (Client)]Invalid authorization specification
```

This error was produced when the `LogonUser/LogonAuth` attributes were invalid and the connection attempt has been refused. Only the Easysoft ODBC-ODBC Bridge was involved in this process.

```
[Easysoft ODBC (Server)][Microsoft][ODBC Driver Manager]
Data source name not found and no default driver specified
```

This error was produced by the Microsoft ODBC driver manager on the OOB Server machine when the `TargetDSN` attribute specified a DSN which does not exist on the server. You can see that the last item in square brackets was the `"ODBC Driver Manager"` and hence it is that component which generated the error text.

The text is also prefixed with `"[Easysoft ODBC (Server)]"`, which means that the error occured with the driver manager at the server end.

```
[Easysoft ODBC (Server)][Microsoft][ODBC SQL Server
Driver][SQL Server]
Login failed for user 'demo'.
```

This error was produced when the `TargetUser/TargetAuth` specified at the OOB Client was passed through the DBMS which refused the connection.

The last item in square brackets was `"SQL Server"` and so you know that SQLServer turned down the connection attempt.

Therefore, if you encounter any error messages when using the Easysoft ODBC-ODBC Bridge, please note the last item specified in square brackets, because that is the component that generated the error text and the error may not lie in the Easysoft ODBC-ODBC Bridge.

If you are integrating the Easysoft ODBC-ODBC Bridge with your own ODBC application, it is worthwhile implementing diagnostics in your application because then your error messages will indicate in which component the problem lies.

The next section explains how to implement ODBC diagnostic messages.

Implementing ODBC diagnostics

This section explains how to implement ODBC diagnostics in your own ODBC application, so that the application can display diagnostic messages if an error occurs at any point in the connection process.

STATUS RETURNS

All ODBC APIs return a status value which may be used to check whether the function succeeded or not.

In C you can test the return value from an ODBC function using the macro `SQL_SUCCEEDED`:

```
SQLRETURN fsts;
/* Assume environment has already been allocated
*/
SQLHENV  envh;
SQLHDBC  dbch;
fsts = SQLAllocHandle(SQL_HANDLE_DBC, envh,
&dbch);
if (!SQL_SUCCEEDED(fsts))
{
/* an error occurred allocating the database
handle */
}
else
{
/* Database handle allocated OK */
}
```

`The macro SQL_SUCCEEDED is defined as:`

```
#define SQL_SUCCEEDED(rc) (((rc)&(~1))==0)
```

Virtually all ODBC functions can return two values which indicate success:

- SQL_SUCCESS
- SQL_SUCCESS_WITH_INFO

Both of these returns cause the SQL_SUCCEEDED macro to result in 1. If a function returns SQL_SUCCESS_WITH_INFO it means that the call succeeded, but an informational message was produced.

For example, with some drivers you might set the cursor type, prepare a statement and then execute it. When SQLExecute is called the statement is acted upon, but the driver might change the cursor type to something else.

In this case, SQLExecute would return SQL_SUCCESS_WITH_INFO and the driver would add a diagnostic indicating the cursor type had been changed.

You should note that a few ODBC functions return a status which fails the SQL_SUCCEEDED macro, but do not necessarily indicate an error. For example, SQLFetch can return SQL_NO_DATA indicating there is no further rows in the result set), this is not necessarily an error.

Obtaining Diagnostics

When an ODBC function returns an error or SQL_SUCCESS_WITH_INFO then the driver will associate a diagnostic with the handle used in the ODBC call.

You can obtain the diagnostic to find out what failed by calling SQLGetDiagRec with the handle you used in the ODBC call that failed.

The driver may associate multiple diagnostic records with a handle. You can call SQLGetDiagField and request the SQL_DIAG_NUMBER attribute to find out how many diagnostics exist.

Alternatively, as diagnostic records start at 1, you can repeatedly call SQLGetDiagRec asking for record 1, then 2 (and so on) until SQLGetDiagRec returns SQL_NO_DATA.

As, an example, the following C function takes a function name string, handle type and handle and retrieves all the diagnostics associated with that handle.

```
void extract_error

    (char *fn,

     SQLHANDLE handle,

     SQLSMALLINT type)

{

    SQLINTEGER   i = 0;

    SQLINTEGER   native;

    SQLCHAR      state[ 7 ];

    SQLCHAR      text[256];

    SQLSMALLINT  len;

    SQLRETURN    ret;

    fprintf(stderr,"\n"

                 "The driver reported the following
diagnostics whilst running "

            "%s\n\n",

            fn);

    do

    {

        ret = SQLGetDiagRec(type, handle, ++i, state,
&native, text, sizeof(text), &len );

       if (SQL_SUCCEEDED(ret))
```

```
            printf("%s:%ld:%ld:%s\n", state, i, native,
text);
    }
    while( ret == SQL_SUCCESS );
}
```

Using this example, which attempts to allocate a database handle, you could use `extract_error` as follows:

```
SQLRETURN fsts;
/* Assume environment has already been allocated */
SQLHENV   envh;
SQLHDBC   dbch;
fsts = SQLAllocHandle(SQL_HANDLE_DBC, envh, &dbch);
if (!SQL_SUCCEEDED(fsts))
{
  extract_error("SQLAllocHandle for dbc", envh,
SQL_HANDLE_ENV);
  exit(1);
}
else
{
/* Database handle allocated OK */
}
```

ODBC 2.0 applications will use `SQLError` instead of `SQLGetDiagRec`.

Diagnostic Fields

When you call SQLGetDiagRec you can retrieve three diagnostic fields:

- State
- Native error code
- Message text

The state is a five character SQLSTATE code. The first two characters indicate the class and the next three indicate the subclass. SQLSTATE codes provide detailed information about the cause of a warning or error.

REF	For the definitive SQL CLI document consult the **Open Group CAE Specification C451, ISBN 1-85912-081-4** (**http://www.opengroup.org/pubs/catalog/c451.htm**). The **Microsoft ODBC 3.0 Programmer's Reference, ISBN 1-57231-516-4** explains ODBC usage in some detail.

The native error code is a code specific to the data source. This number is often extremely useful to the driver developers in locating an internal error or state.

If you are reporting a bug in the Easysoft ODBC-ODBC Bridge ODBC driver for which you obtained an error you should always quote the ODBC function called, the error text and this native number.

The message text is the text of the diagnostic.

This string takes one of two forms:

- For errors and warnings that do not occur in a data source the format is:

  `[vendor-identifier][ODBC-component-identifier]component-supplied-text`

- otherwise it is:

  `[vendor-identifier][ODBC-component-identifier][data-source-identifer] data-source-supplied-text`

See **"Understanding ODBC diagnostic messages" on page 246** to review some example messages.

Tracing

There are three ways to trace the ODBC calls an application makes through the driver manager and the OOB Client ODBC driver:

1. tracing in the driver manager may be turned on.
2. tracing in the OOB Client ODBC driver may be turned on.
3. an application can turn tracing on via the ODBC API `SQLSetConnectAttr` (...,`SQL_ATTR_TRACE`,...). The trace filename may also be specified with the `SQLSetConnectAttr` attribute `SQL_ATTR_TRACEFILE`.

Starting tracing in the driver manager is platform-specific:

[1a] Windows:

Start the ODBC driver manager administration interface via **Start Menu > Control Panel > ODBC Data Sources**.

Click on **Tracing**, ensure the specified filename is valid and click **Start Tracing Now**.

[1b] Unix:

If you are using the unixODBC driver manager then tracing is enabled in the `odbcinst.ini` file (usually `/etc/odbcinst.ini`).

To enable tracing you must add two attributes to the [ODBC] section (if you do not have an `[ODBC]` section, create one):

```
Trace = Yes

TraceFile = /path/filename
```

e.g.

```
[ODBC]

Trace = Yes

TraceFile = /tmp/sql.log
```

Make sure that the user who is running the application to be traced has write permission to `TraceFile` (and to the directory containing it), or you will not get any tracing at all.

Driver manager trace files show all the ODBC calls applications make, their arguments and return values, but OOB Client ODBC driver tracing is specific to the Easysoft ODBC-ODBC Bridge and is of most use when making a support call.

You can enable OOB Client ODBC tracing by method [3] or by a platform-specific method:

[2a] Windows

Update the **Registry** by running `regedit` and edit the key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\EASYSOFT ODBC-ODBC
BRIDGE\CONFIGURATION\SYSTEM\SETTINGS
```

Edit the `Logging` string value and set it to `0xffffffff`, which turns on all tracing (see **Easysoft ODBC-ODBC Bridge FAQ** for tracing bitmask values).

A trace file named `esoobclient.log` is written to the `LogDir` directory (which is defined in the same registry key).

[2b] Unix:

OOB Client ODBC driver tracing can be enabled by adding a `{Settings}` section to the end of your `odbc.ini` file.

e.g.

```
{Settings}

Logging = 0xffffffff
```

The `Logging` value is a bitmask and the values are listed in the **Easysoft ODBC-ODBC Bridge FAQ**.

APPENDIX B GLOSSARY

Terms and definitions

API

Application Programmer Interface. An API is a published set of function calls and constants allowing different programmers to utilize a ready-written library of subroutines.

Application

An Application Program ("`Application`" or "`App`") is a program that *applies* the computer to solving some real-world problem. In ODBC terms, it is the program connecting to the data source.

Authorization code

You must have an authorization code for the Easysoft product you wish to license in order to obtain a purchased license. When you purchase a product your authorization code is emailed to you. You do not need an authorization code to obtain a trial license.

Bitmask

A value which, when written out in binary, has a meaning assigned to each digit, which can be 0 or 1. This is a very efficient way of storing a number of *flags* in a small amount of memory.

When viewed in decimal it is a single number resulting from adding up the values of the individual bits. The bits are worth 1, 2, 4, 8, 16, 32 and so on.

Client/Server

The name given to the architecture whereby one process (the *server*) keeps track of global data, and another task (the *client*) is responsible for formatting and presenting the data. The client requests queries or actions be performed on the data by the server. Often these processes run on different *hosts* across a local-area network.

Column

The vertical dimension of a table. Columns are named and have a *domain* (or *type*). The term *column* might refer to only the *definition* of a column (i.e. its name and type), or to all the data in it.

Connection String

ODBC *driver managers* accept a connection string when a client connects. Ideally it contains all necessary attribute values to make the connection to a data source, but provision is made for the driver to negotiate with the application or the user for any missing information.

Data Source

In ODBC terms, a data source is a database or other data repository coupled with an ODBC Driver, which has been given a Data Source Name (see **“DSN” on page 259**) to identify it to the ODBC Driver Manager.

DLL

Dynamic Link Library. Windows’ mechanism for shared object code. See also **“Shared Object” on page 261**.

Download

The transfer of data from a remote machine (on the internet, for example) to your local machine. Mechanisms for achieving this include FTP and the World Wide Web.

Driver

See **"ODBC driver" on page 260**.

Driver Manager

Software whose main function is to load ODBC drivers. ODBC applications connect to the Driver Manager and request a *DSN*. The Driver Manager loads the driver specified in the DSN's configuration file. In Windows, the ODBC Data Source Administrator is used to set up the Driver Manager.

DSN

Data Source Name. This is a name associated with an ODBC data source. Driver Managers, such as unixODBC or the Microsoft Windows Driver Manager, use the Data Source Name to cross-reference configuration information and load the required driver.

Field

A placeholder for a single datum in a record, for example you can have a Surname field in a Contact Details record. Called a *cell* in Microsoft Access.

Flags

Single-bit values, representing 'Yes' or 'No'. When more than one flag is present, they are normally stored in a *bitmask*.

Host

A computer visible on the network.

HTTP

HyperText Transfer Protocol. The means of transferring web pages.

HTTPAdmin

An NT user name, valid on the machine the server is running on. This is the only user allowed to amend settings, or display certain items, through theWeb Administrator.

Middleware

Software that is placed between the *client* and the *server* to improve or expand functionality.

ODBC (Open DataBase Connectivity)

A standard *API* for connecting application programs to relational database systems through a suitable *driver*. ODBC is available on a wide number of platforms and the Easysoft ODBC-ODBC Bridge allows the database and the application to reside on different machines across the network.

ODBC driver

Software that accesses a proprietary data source, providing a standardized view of the data to ODBC.

Operating System

A collection of software programs, APIs and working practices that control and integrate the execution of system functions on behalf of application programs.

Platform

The term *platform* normally covers the hardware and operating system as a unit. For example; a PC running Microsoft Windows, a PC running BSD Unix, and a Sun running Solaris are three different platforms.

Server

A computer, or *host*, on the network, designed for power and robustness rather than user-friendliness and convenience. Servers typically run round-the-clock and carry central corporate data.

– OR –

A process performing the centralized component of some task, for example extracting information from a corporate database. See **“Client/Server” on page 258**.

Shared Object

A piece of object code (i.e. a program fragment) for loading and executing by other programs.

SQL (Structured Query Language)

A standard language for interacting with relational database systems, based on Relational Theory.

System Data Source

In the context of ODBC under Microsoft Windows, a data source which can be accessed by any user on a given system. See also **“User Data Source” on page 262**.

Table

A data set in a relational database, composed of rows and columns.

For example, the following table has two columns (`vendor` and `name`) and two rows (one for the Easysoft ODBC-ODBC Bridge and the other for MySoft's ODBC client software) :

software	
vendor	name
Easysoft	Easysoft ODBC-ODBC Bridge
MySoft	My ODBC Client Application

The term *table* can also apply to just the definition of the table, without its data.

User Data Source

An ODBC Data Source with access limited to a specific user on a given system. See also **"System Data Source" on page 261**.

Web Administrator

The OOB Server under Windows runs a simple web server which can be used to display and amend certain server statistics and parameters.

INDEX

Symbols

.odbc.ini
  see files
/easysoft directory 63
/etc/defaults/rc.conf 67
/etc/inetd.conf
  editing by hand 190
/etc/ld.so.conf 67
/etc/odbc.ini
  see files
/etc/services 73
  editing by hand 190
/sbin/ldconfig 67
/sbin/ldconfig -m 67
/usr/local 62
{x}inetd configuration files 73

A

Access 97, 104, 136
access control 177
  in the registry 186
Add/Remove Programs icon 56
allowed list 178
Apache/PHP 143
  building with OOB 201
API 257
application 257
Applixware 146, 203
attributes
  for OOB Client DSNs on Unix 144-147
  for OOB Client DSNs on Windows 128-133
  for OOB Server DSNs on Unix 106

for OOB Server DSNs on Windows 100
mandatory 147
passed through 147
authentication 90
administrative privileges of OOB server 181
failure 135
for OOB Client DSNs on Unix 145-147
for OOB Client DSNs on Windows 130-131
user accounts and the server 183

B

beta releases 30
bitmask 257
in esoobserver.ini 192
in registry 186
bunzip 59
bzip2 33

C

C runtime library
version of 61
Caution box 15
CD 30
client
setup 125
setup on Unix 142
setup on Windows 126
client-server 258
ColdFusion 143, 204
column 258
compress 33
configuring the server
Unix 188
Windows 2000/NT 181
connection failure 91

connection pooling 239
connection process 86-95
connection string 90, 94, 258
Control Panel 102
    ODBC 97
create data source
    for client on Unix 142
    for client on Windows 126
    for server on Unix 105
    for server on Windows 99

D

data source 258
    name 94
    see also attributes
    see also create data source
    system 261
    user 262
data source administrator
    on Unix 107, 151
    on Windows 126
DBMS 20
demo client
    for Unix 153
    for Windows 139
demo.easysoft.com 129, 136, 139
demo.exe 129
denied list 178
DLL 258
documentation 30
    additional resources 14
    text files 197
download 259
driver 259
driver manager 259

for Unix 95
for Windows 95
functionality in the OOB 142
role in the connection process 93
unixODBC 64
DSN 259
see also data source name
dynamic library locations 67

E

environment variables
ODBCINI 143
esoobserver 189
in inetd.conf 188
in services file 188
in startup script 189
esoobserver.ini 192
example C program 199
example files 198

F

fallback servers 237
FAQ 14
field 259
files
.odbc.ini 143
/etc/odbc.ini 105
odbc.ini 143
odbc32.dll 95
odbcinst.ini 107, 144, 151, 254
flags 259
FreeBSD 67
FTP 30

G

glibc
    see C runtime library
gunzip 59
gzip 33

H

HKEY_LOCAL_MACHINE 185
host 259
hosts.allow 178
hosts.deny 178
hsODBC 210
HTTP 260
HTTPAdmin 260
HTTPAdmin user name 166

I

inetd 73, 95, 190
    and the server 188
inetd.conf 188
install directory 62
installation
    file name 32
    requirements 58
    requirements for Unix 58
installing
    on Unix 58
    on Windows 35
interfacing 197
isql 91, 109

J

Java 19
JDBC-ODBC Bridge 19

Jet 100

K

kill 190

L

LD_LIBRARY_PATH 77
LD_RUN_PATH 77
libc6
    see C runtime library
LIBPATH 67
license
    authorization code 49
license agreement 43, 60
logon account 130
logonauth 183
    see also attributes
logonuser 183
    see also attributes

M

Microsoft Access 97, 104, 136
    connecting with 136
middleware 260
mnoGoSearch 208
-mt 33
multi-process 100, 104
multi-threaded 100, 104, 239
mxODBC
    building 209

N

Navision 173, 187, 195
NICs

multiple 232
Northwind 97
Note box 15
NT services 95, 103

O

ODBC 260
background to 20
versions supported 244
ODBC Data Source Administrator 97
ODBC driver 260
ODBCConfig 107, 151
odbcinst 68
OOB
components of 31
DSN configuration dialog box 128
installing on Unix 58
installing on Windows 35
uninstalling on Unix 78
uninstalling on Windows 56
OOB Client 24, 31
OOB Server 24, 31
oobping 111-124
open database connectivity 260
operating system 260

P

password
see authentication
patches 30
Perl DBI
building with DBD::ODBC 205
permissions to the OOB Server 184
PHP
building with Apache and OOB 143, 201

platform 260
Platform note 15
port 74, 130
    changing for OOB Server 190
process affinity 238
Python/mxODBC 209

R

Reference box 15
registry 56-57, 185
    access control 186
remote machine
    specifying 129
    specifying DSN 130
remote procedure call
    see RPC
Rexx/SQL
    building 206
RPC 136

S

security 177
    see also authentication
server 261
    installing 70
    setup 96
    setup on Unix 105
    setup on Windows 97
    standalone 190
    testing 111
    Windows 2000/NT 181
server flags
    see flags
server parameters
    advanced editing on Unix 192

advanced editing on Windows 185
service manager 182
service name 74
changing for OOB Server 190
service status
automatic 182
disabled 182
manual 182
services 73
NT 103
services file 188
Services icon 181
shared object 261
SHLIB_PATH 67, 77
SIGHUP 190
SQL 261
SQLBrowseConnect 245
SQLPlus_hsODBC 210
standalone server 190, 191
StarOffice 146, 207
startup script 189
structured query language 261
supported API 244
symbolic link 64, 192
system account 183
system data source 261

T

table 262
tar 59
targetauth 130
targetdsn 130
targetuser 130
TCP/IP 129
telnet 146

thread-safe 33, 100, 104
tracing 253
transport protocol 129

U

UDMSearch 208
uncompress 59
uninstall 56
unixODBC 96
    obtaining 96
    why it should be used 211-212
unsupported API 244
upgrades 30
UseOOBDBAuth 91
user data source 262

W

Web Administrator
    accessing 159
    Client Hosts screen 180
    Configuration screen 165
    Data Sources screen 176
    Information screen 179
    Security screen 177
    Statistics screen 162
web site 30
Windows 95 34
WinSock2 34

X

X server 105, 142
xinetd 73