•Staple Guide•

Easysoft® Data Access 2000

ODBC-ODBC Bridge

Installation Guide and User Manual

•Staple Guide•

Document Version 1.0

Publisher: Easysoft Limited

3 The Embankment
Sovereign Street
LEEDS
LS1 4BJ
United Kingdom

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

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

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

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

Last Modified: December 7, 1999 11:20 am

Printed: December 7, 1999.

Printed in United Kingdom.

PREFACE

About this manual

Intended Audience

This manual is intended to cover the full range of requirements for anyone wishing to install, use, or configure the Easysoft® Data Access 2000 ODBC-ODBC Bridge. Supplementary information is provided for those wishing to program ODBC applications that link either directly to the bridge (Unix only) or through a driver manager (Unix or Windows environments). This manual is not an ODBC programming manual.

The sections written for the MS-Windows platforms require some familiarity with the use of buttons, menus, icons and text boxes. Just about anybody with experience of Apple Macintosh computers, Microsoft Windows or the X Window System will have no difficulty with these sections.

The Unix-based sections require that you are comfortable with a Unix shell, and can perform basic functions like editing a file. More complex activities are spelled out more clearly, but it helps to understand how your system handles dynamic linking of shared objects.

Displaying the Manual

This manual is available in a variety of forms. The Portable Document Format (.PDF) file can be displayed on-screen using the Acrobat Reader, available free from Adobe, or printed. The PDF is set up for best results when printed double-sided, and staple guides on the front page allow you to bind the pages without losing text in the 'gutter' between facing pages.

The paper size in the PDF is short and squat to help you get a whole page on the screen at a time. PDF is not the best medium for reading online, however;— although the typesetting is consistent and the format does support hyperlinks, it is essentially a ‘paper simulator’ and navigation is a little more cumbersome than it needs to be. For online reading, we recommend the HTML version of the manual, or the soon-to-be-available HTMLHelp version. These have the advantages that screenshots will display without loss of information no matter what the resolution of the display screen, and that text will flow to fit whatever window size and font size you have set up.

Notational Conventions

Across the range of Easysoft manuals you will encounter passages that are emphasised with a box and label. There are four box types, explained below:

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 reiterate information clearly, where the body text may be unclear.

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 the preceding paragraph, as Caution boxes are important!

A reference box refers to resources external to the manual, whether suggested reading on paper or the internet, or a useful website or other source of useful resources.

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 need to be logged in as root to make many important changes.

Typographical Conventions

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

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

  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:

  1. At the command prompt type `admin`.

Additionally, interactive sessions are shown in a **`strong monotype`** font (for system output) and `monotype` (for user input), for instance:

2. Once you have logged in, type the following:

```
C:\>admin
```

- Keyboard Commands

  It is assumed that all typed commands will be terminated with the <*Enter*> key, and as such this will not normally be indicated in this manual. Other keypresses are indicated by the key names being enclosed by angle brackets, and presented in oblique text, e.g.

3. Hit <*F1*> for help.

NB	Key presses are shown in italic to distinguish between, for example, hitting <*del*> and typing `<del>`, that is, typing an open angle bracket (`<`), followed by the three character string `del` followed by a close angle bracket (`>`).

- Any file listings will be presented using the `monotype` style.
- Other names on the system, such as file names, directories and database fields are presented `monospaced` too.

Chapter Guide

Because Easysoft® Data Access 2000 middleware supports such a broad spectrum of platforms, we have grouped information within each chapter into two broad classes of system, Windows and Unix. When printed double-sided, useful tabs are marked to help you skip to the section relevant to your operating system.

Early indications are that the majority of installations will have a Windows NT server and a Linux-based client. For this reason, although either part of the bridge may be installed on any supported platform, the chapters each emphasise the more popular platform.

- **Chapter 1, “Introduction’’**

  gives an overview of the ODBC architecture, and shows what the ODBC-ODBC Brings to it.

- **Chapter 2, “Installation’’**

  is a step-by-step guide to installing the software.

- **Chapter 3, “Connection’’**

  explains everthing you need to know (almost!) to set up an ODBC connection across a network.

- **Chapter 4, “Configuration’’**

  explains the configuration options for the server, in windows or unix.

- **Chapter 5, “Interfacing’’**

  this chapter helps you to build your unix client application to connect through the bridge. It covers C, PHP under Apache, Perl with the DBI and DBD::ODBC modules installed, mxODBC and Rexx/SQL.

Trademarks

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

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

Easysoft, Easysoft Data Access 2000 and OOB are trademarks of Easysoft Ltd.

CONTENTS

INTRODUCTION

1

Introducing the the Easysoft ODBC-ODBC Bridge

Easysoft Data Access 2000 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 Data Access 2000 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 single-client access to data wherever it is stored.

Chapter Guide

Why ODBC?

Historically, corporate data was held on large, centralised 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 Data Base Management System (DBMS) they wished to use it with.

Relational databases and SQL went part way toward alleviating the problem. 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. 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. It allows an application to be written without considering the intricacies of the particular database engine it connects to — 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 1.1** illustrates the principle of separating the driver from the application.

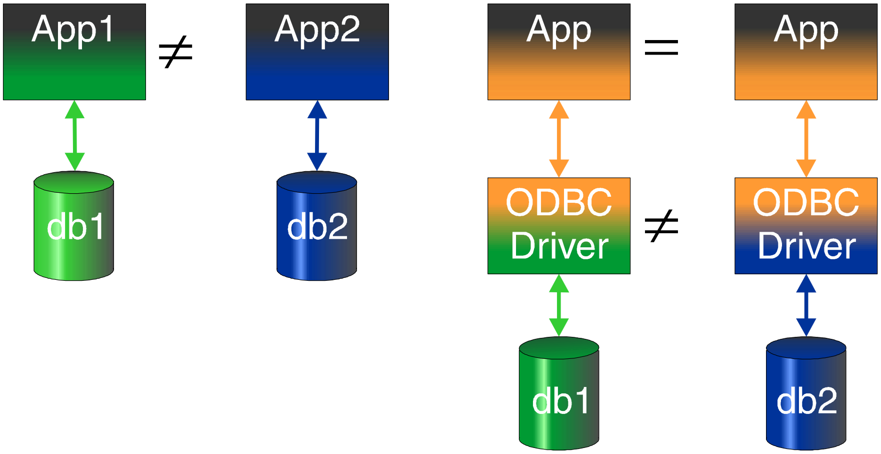


Figure 1.1: Before and after ODBC.

Before ODBC, even if App1 and App2 were functionally equivalent, there had to be two programs, one for each Data Base Management System (DBMS) — See left-hand side of **Figure 1.1**.

ODBC permits the DBMS-specific parts of the program to be separated from the part that fulfils the functional requirement (Right-hand side of **Figure 1.1**). The result is that the completed application can be attached to any DBMS that has a corresponding driver — currently there are over fifty DataBase Management Systems supported in this way.

Three Benefits of Driver Managers

ODBC on its own does not provide much — without drivers, it does nothing at all! Even with drivers, it would still be necessary to re-link an application's object code with a specific driver in order to access a given DBMS. In order to allow off-the-shelf software to take advantage of ODBC, Microsoft provide a Driver Manager with their operating systems.

The driver manager is essentially a dynamic linker, allowing the user to pair up application and driver in a clean and user-friendly manner.

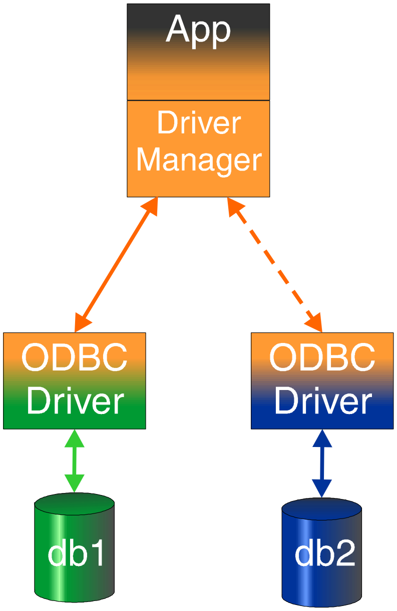


Figure 1.2: Driver Manager as Dynamic Linker.

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 client and server to add strategic functionality such as joining heterogeneous databases into one datasource or bridging a network.

Why the Easysoft ODBC-ODBC Bridge?

The ODBC-ODBC bridge from Easysoft (the OOB) is data access middleware that allows an application running on one platform to access an ODBC datasource on another platform. A typical business requirement is outlined below.

Example Business Requirement

The company's marketing coordinator has a PC running Windows NT, with a Microsoft Access database. The coordinator uses Access queries to generate statistical reports for higher management, and enjoys the consistent user interface and flexibility of the Windows system.

The company also has a website running on an Apache server under Linux, which tracks the progress of users and collects profile data. 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 website administrator collates the figures at the end of each week, and emails them to the marketing coordinator, who pastes them into Access. The web administrator has set up some scripts to generate the figures, but is often too busy to deliver the figures by the end of the week. The marketing coordinator has to take care that the columns in the Access table are displayed in the right order, or the paste operation results in figures appearing in the wrong columns.

A Solution using the OOB

The ODBC-ODBC Bridge (OOB) allows a script on the web server to run an SQL update query directly on the NT machine, removing the need for regular human intervention. A small script along with a very simple client program submit SQL to the bridge, which passes it across to the database software on the NT machine. The script is set up to trigger automatically at the end of the week, and pasting errors are removed because the SQL query explicitly names the destination columns.

Where the Bridge Fits In

The Easysoft ODBC-ODBC Bridge comes in two parts — a client and a server. The bridge client appears to the ODBC client program just as any other ODBC driver, and the bridge server connects to the database engine as an ODBC client application. This terminology may seem a little confusing at first, so we will take it step by step.

First, **Figure 1.3** shows the typical ODBC set up. The client 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.

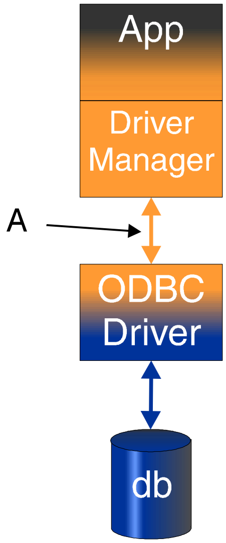


Figure 1.3: The typical ODBC configuration.

Imagine that the ODBC-ODBC Bridge is a black box that takes ODBC calls in at the top and passes them straight out again at the bottom. We can insert this box at *A* and as far as the application or the database are concerned, nothing has changed.

What *has* changed however, is that the application and the 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 — Microsoft's SQL Server allows ODBC client applications to connect to a remote SQL Server datasource. With the OOB however, the database can be any ODBC datasource (not just SQL server) and the client may be on Windows, Linux, Solaris or any other supported platform.

Inside this 'black box' we have a client-server pair that handle all the network operations, passing function references and parameters one way and return values the other (**Figure 1.4**).

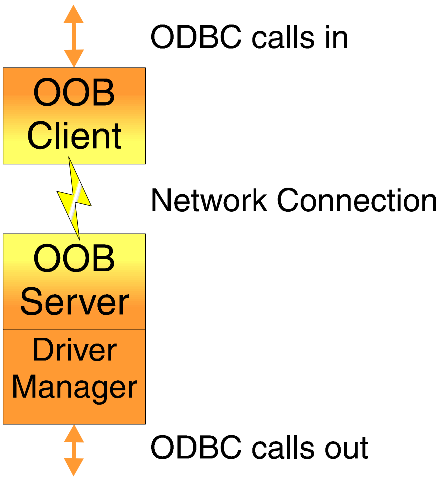


Figure 1.4: The ODBC-ODBC Bridge.

Figure 1.5 shows the system set up across a network. Compare it with the model in **Figure 1.3** — the client side fits, except that the database is replaced by a network connection; and the server side fits, with the OOB *Server* in the place of the ODBC *client*.

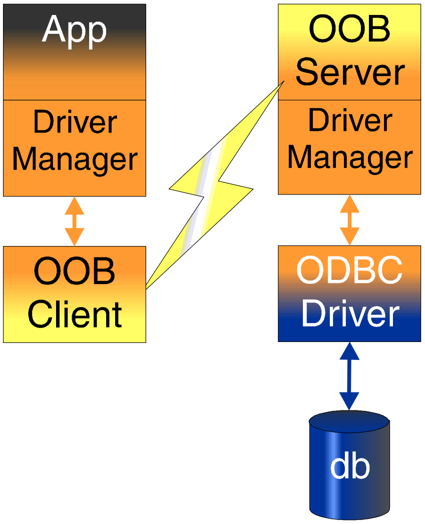


Figure 1.5: Bridging the network with the OOB.

REF

For the definitive SQL CLI document consult **Open Group CAE Specification C451, ISBN 1-85912-081-4**.

Microsoft ODBC 3.0 Programmer's Reference, ISBN 1-57231-516-4 explains the utilisation of ODBC in some detail.

INSTALLATION

2

Installing the Easysoft ODBC-ODBC Bridge

Installing the Easysoft ODBC-ODBC Bridge allows you to access ODBC data sources from other computers across the network running any supported Windows or Unix variant. This chapter covers the tasks necessary to install and remove the Easysoft ODBC-ODBC Bridge.

The Windows installation can be carried out by anyone. The Unix installation assumes you are, or have available for consultation, a system administrator.

Chapter Guide

What to Install (Windows and Unix)

The Easysoft ODBC-ODBC Bridge (OOB) consists of two main components, the OOB Client and the OOB Server. There are shared components which are needed to install either of these, and there are documentation files which you may leave out of the installation if you are comfortable with the OOB and want to save disk space. The Unix installation also offers a third-party driver manager . Windows already has one.

Installation of the OOB is more complicated than most software because the client and the server will typically be installed on different machines with different operating systems. Because of this, you will need to perform the basic installation procedure twice, probably in different environments.

Another important consideration is this: from time to time the protocol — the language spoken between the Client and Server — may change. For this reason you should try to ensure the client and server version numbers match in the first three fields, for example:

- Client version 1.0.0.3 may not work with Server version 1.0.1.3.
- Client version 1.0.0.3 will work with Server version 1.0.0.9. Strictly, only the first three fields constitute the version number. The last digit is called the ‘build number’ and does not affect compatibility.

NB If you connect a client and server with mismatched protocols, the bridge will detect this and fail straight away. What the user sees depends on the client application, so it is important to be aware of what server version(s) your target machine is running.

Obtaining The Software (Windows and Unix)

The Easysoft web site and FTP site are available 24 hours a day, offering Easysoft products for download, beta releases, free patches, upgrades, and documentation for Easysoft products. The web site can be found at **http://www.easysoft.com/**. Click **products** to go to the download area for definitive releases. If the browser asks you whether to open or save the file, save it to a temporary directory.

If you prefer to use the FTP protocol, the server is **ftp.easysoft.com**. This is a more efficient use of network and client resources and does not require a web browser. Change to the `pub/product` directory, switch to `binary` mode and `get` the file.

If you have an extremely slow connection, or no internet access at all, you can obtain the software on CD by writing to Easysoft at the address given in **Appendix D**. When you have the CD, the product can be found in the folder `\Products\OOB`.

The name of the install file varies from platform to platform, but you can expect something of the form:

- `EasysoftODBC-ODBCBridge`*v_w_x_z* (Windows), or
- `es-odbc-odbc-bridge-`*v*`.`*w*`.`*x*`.`*z*`.`*platform*`.tar.gz` (Unix platforms)

... where *v* is the major version number, and *w* and *x* are minor version numbers. *z* is build index, which is incremented when minor changes are made which do not affect the client/server protocol.

Within your licensed major version number, you should go for the highest release available for your platform. To install software of a different major number requires a new license.

Whichever medium you choose, you should now download the file and begin the installation process. The exact installation process depends on your operating system. You should refer to the section relevant to your platform to continue.

- **Installation Under Windows .... page 2-13**
- **Uninstalling Under Windows....................................... page 2-26**
- **Installation Under Unix .... page 2-29**
- **Uninstalling Under Unix... page 2-42**

Installation Under Windows

For Windows, the Easysoft ODBC-ODBC Bridge installation comes as an executable named `EasysoftODBC-ODBCBridge`*v_w_x_y*`.exe`, where *v* is the major version number, *w* and *x* are the minor version numbers, and *y* is the build number. If there is a choice of files then you should download the file with the highest version number.

1. From the web, click to download the installation file

   – OR –

   In your FTP client, switch to `binary` mode and `get` the installation file

   – OR –

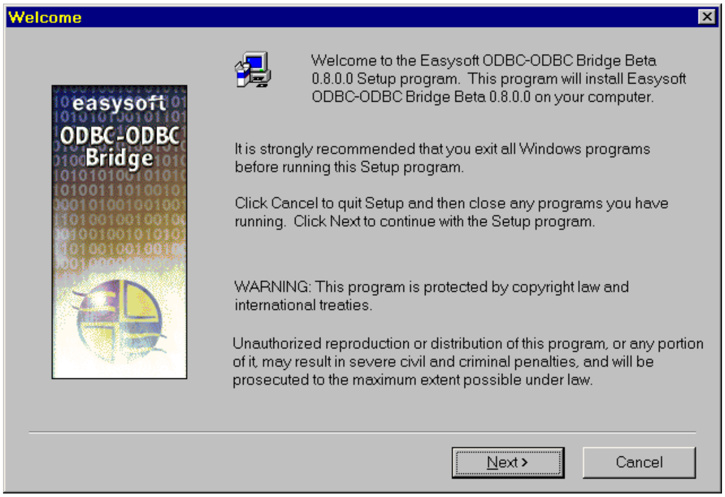
   If you have a CD, navigate to the folder with the installation file in it.

Caution!

> Please shut down other Windows programs before installing.
>
> If you are upgrading you must remove any previously installed version before continuing. Refer to **Uninstalling Under Windows** for more details about this procedure.
>
> If there is already an OOB Server running when you install a new version of the OOB, it will be stopped and all connections to it will be broken.

2. Execute the file `EasysoftODBC-ODBCBridge`*v_w_x_y*`.exe`. There will be a short delay while setup prepares the wizard to guide you

through the rest of the install procedure, then you are presented with the Welcome screen.



Welcome to the Easysoft ODBC-ODBC Bridge Beta 0.8.0.0 Setup program. This program will install Easysoft ODBC-ODBC Bridge Beta 0.8.0.0 on your computer.

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

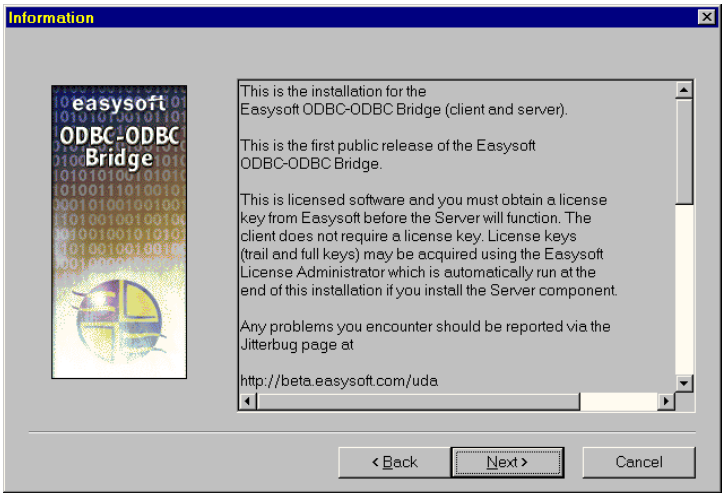
Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

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 2.1: Welcome Dialog.

3. Click **Next** when you have read the welcome screen. A window appears containing notes specific to that release.



This is the installation for the
Easysoft ODBC-ODBC Bridge (client and server).

This is the first public release of the Easysoft
ODBC-ODBC Bridge.

This is licensed software and you must obtain a license
key from Easysoft before the Server will function. The
client does not require a license key. License keys
(trail and full keys) may be acquired using the Easysoft
License Administrator which is automatically run at the
end of this installation if you install the Server component.

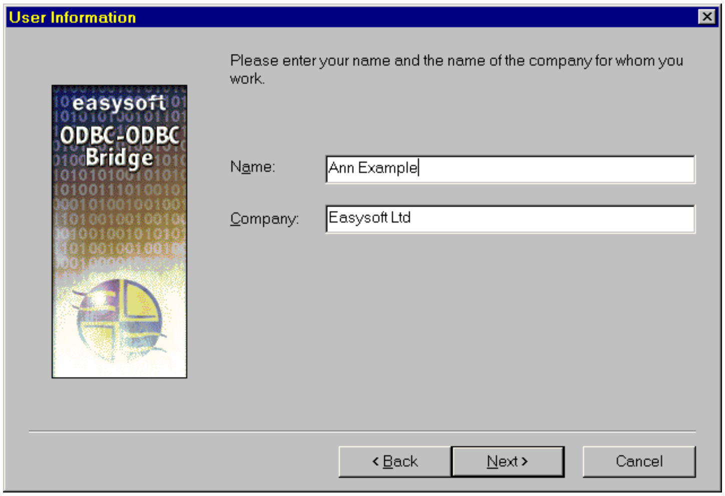
Any problems you encounter should be reported via the
Jitterbug page at

http://beta.easysoft.com/uda

Figure 2.2: Release Information Dialog.

4. Once you have read the release information, click the **Next** button.

The User Information dialog appears.



User Information

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

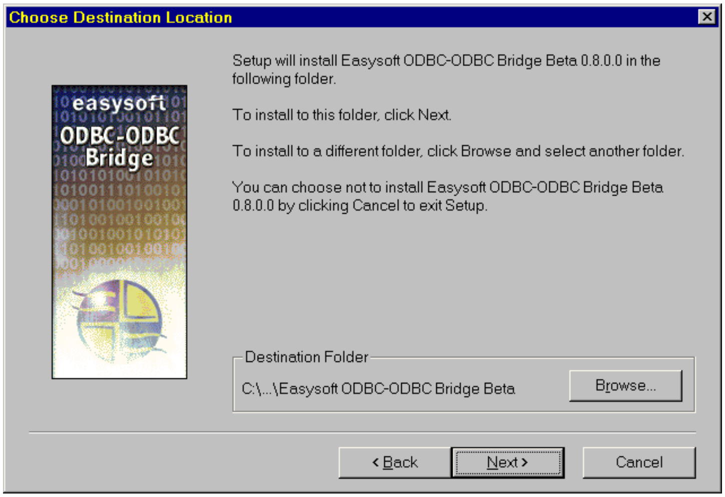
Name: Ann Example

Company: Easysoft Ltd

Figure 2.3: The User Information Dialog.

5. Enter your name and the name of your company. Click **Next**.

   The next window allows you to choose where to install the files.



Choose Destination Location

Setup will install Easysoft ODBC-ODBC Bridge Beta 0.8.0.0 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 Beta 0.8.0.0 by clicking Cancel to exit Setup.

Destination Folder

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

Figure 2.4: Destination Selection Dialog.

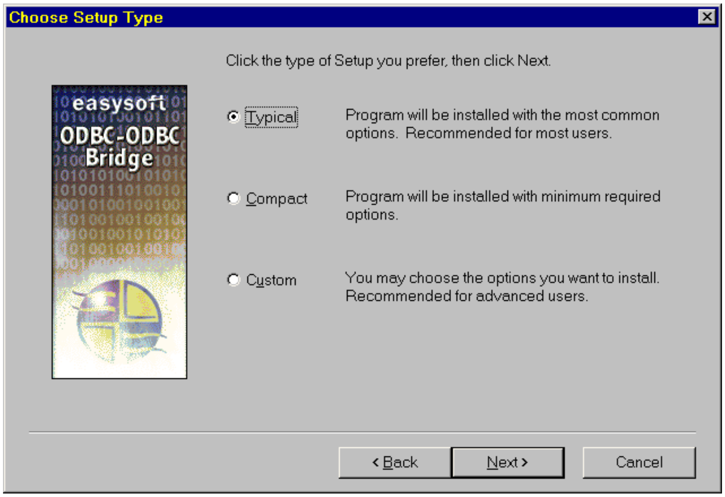
6. Click the **Next** button to install the application into the default directory location:

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

   – OR –

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

   The next screen offers you a choice of three setup types: Typical, Compact and Custom.



Choose Setup Type

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 2.5: Select Installation Type Dialog.

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

   – OR –

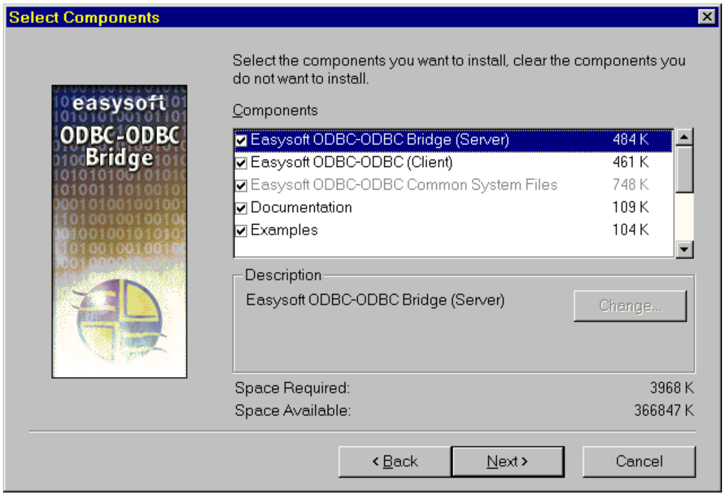
   Select **Compact** to install the client and server but no documentation or examples. This is recommended for users' machines and servers, if your administrator already has a ‘typical’ installation of the bridge elsewhere.

– OR –

Select **Custom** to choose what components to install. This is useful if disk space is at a premium, or you wish to restrict a machine to being either a client or a server.

8. Click **Next** to proceed. If you selected a Typical or Compact installation, skip to **step 10**.

   If you selected Custom, then you will be presented with a checklist of items to install. Note that the **Common System Files** box must be checked if either **Client** or **Server** is checked.



Select Components

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

Components

☑ Easysoft ODBC-ODBC Bridge (Server)	484 K
☑ Easysoft ODBC-ODBC (Client)	461 K
☑ Easysoft ODBC-ODBC Common System Files	748 K
☑ Documentation	109 K
☑ Examples	104 K

Description

Easysoft ODBC-ODBC Bridge (Server)

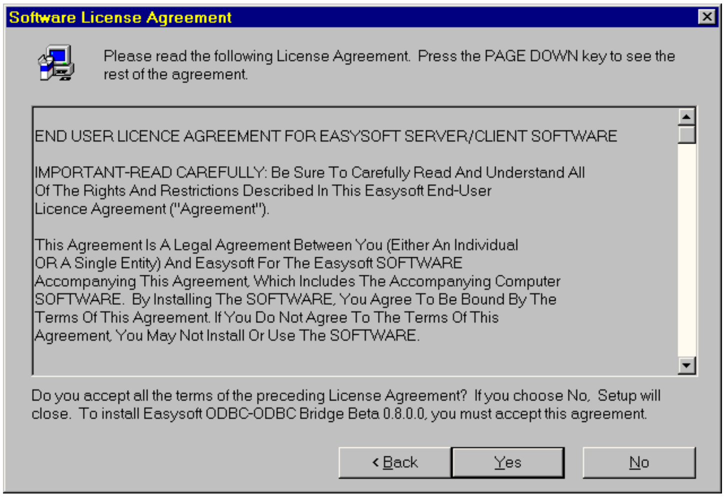
Space Required: 3968 K

Space Available: 366847 K

Figure 2.6: Custom Installation Dialog.

9. Select the items to install and click **Next** to proceed.

10. A License Agreement dialog appears. Different license text appears depending on whether or not you elected to install the server.



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 Beta 0.8.0.0, you must accept this agreement.

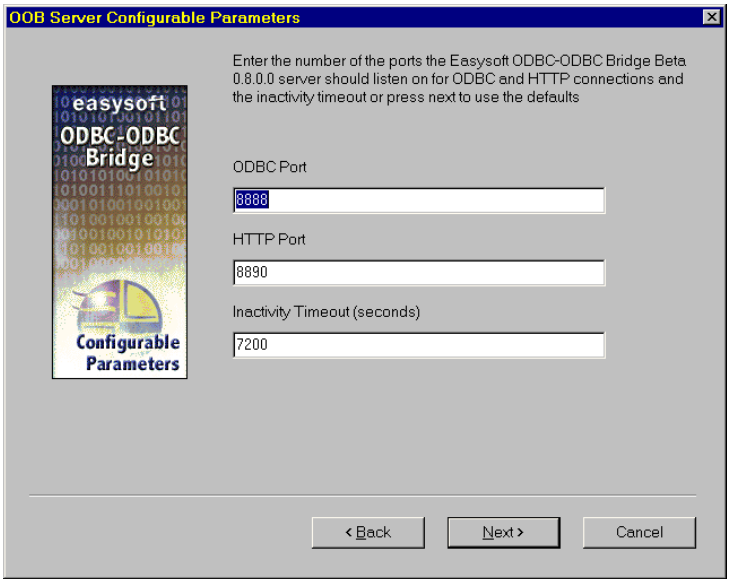
Figure 2.7: License Agreement Dialog.

11. If you agree to the License Agreement then click **Yes** to continue with the installation.

    – OR –

    If you do not agree with the license agreement click **No** and refer to the license agreement in **Appendix C** for more information. Please do not hesitate to contact Easysoft with any questions. Easysoft contact information is given in **Appendix D**, "Contacting Easysoft."

If you elected not to install the Server, then skip to **step 15** Otherwise, a server configuration dialog box will appear.



OOB Server Configurable Parameters

Enter the number of the ports the Easysoft ODBC-ODBC Bridge Beta 0.8.0.0 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

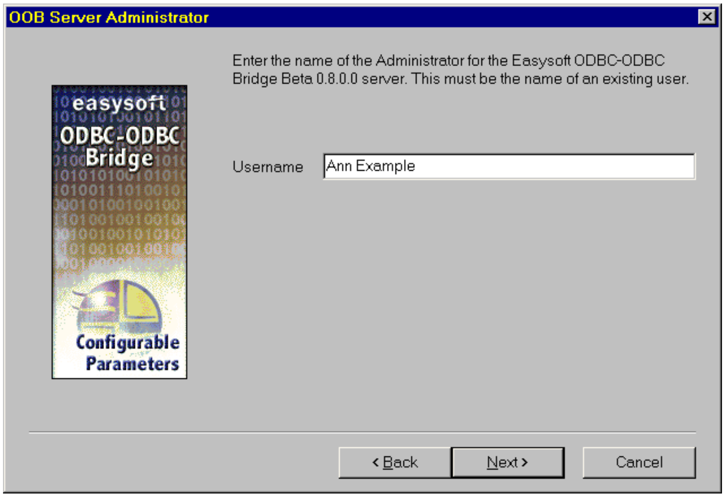
Inactivity Timeout (seconds): 7200

Figure 2.8: The Server Configuration Dialog.

12. The server configuration box contains three settings;

    - the port at which the ODBC bridge server will listen for ODBC bridge clients,
    - the port at which the server's web-based configuration facility will listen, and
    - the inactivity timeout (in seconds).

    Unless you think you will have a port conflict, or have some reason to change the timeout (See "Choosing another Port or Service Name" on page 4-90 for more information,) you should keep the default values and click **Next**. You are asked for a Server Administrator user name.



Enter the name of the Administrator for the Easysoft ODBC-ODBC Bridge Beta 0.8.0.0 server. This must be the name of an existing user.

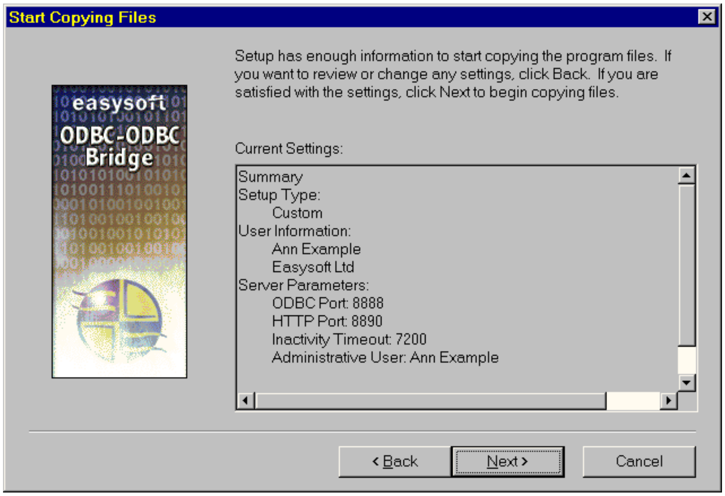
Username: Ann Example

Figure 2.9: Server Administrator’s Username Dialog.

This should be the login name of an existing Windows user account. Only this user will be able to use the online configuration tool to modify server parameters (See "The HTTP Configuration Interface" on page 4-81), so it may be worth creating a specific "OOB Admin" user for the purpose.

13. Enter your chosen administrator’s user name and click **Next**.

14. A summary screen is displayed.



Start Copying Files

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

Current Settings:

```
Summary
Setup Type:
      Custom
User Information:
      Ann Example
      Easysoft Ltd
Server Parameters:
      ODBC Port: 8888
      HTTP Port: 8890
      Inactivity Timeout: 7200
      Administrative User: Ann Example
```

Figure 2.10: Summary Dialog.

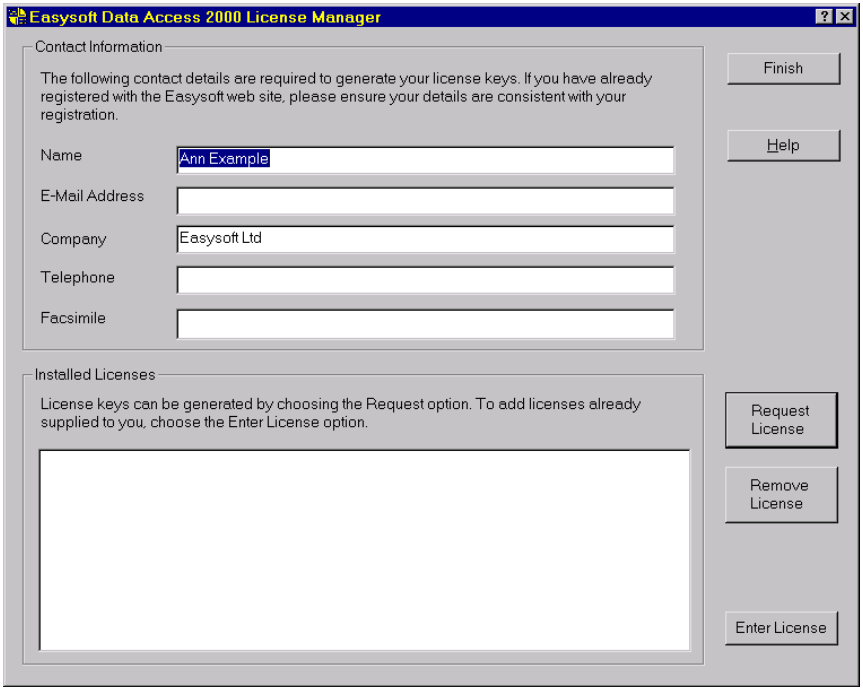
15. Click **Next**. There is now a short wait while the relevant bridge components are copied and configured.

LICENSING UNDER WINDOWS

16. If you selected the **Custom** installation, and elected not to install the server, then you do not need a license. Skip to **step 29**.

   – OR –

   The install program starts the Easysoft License Manager (For full information on the License Manager, refer to **Appendix B**). It is possible to complete the installation without licensing the product, but you will not be able to use the server until a license is obtained. A time-limited *trial license* can be obtained for free, or if you have already purchased a full license you will have an Authorisation 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: Ann Example

E-Mail Address:

Company: Easysoft Ltd

Telephone:

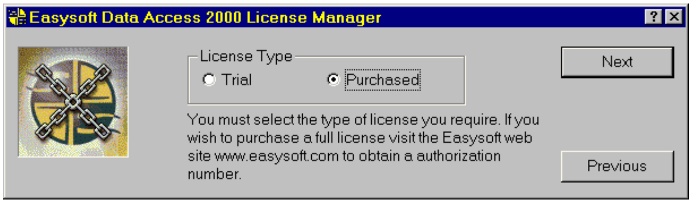
Facsimile:

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 2.11: License Manager Window.

17. Enter your contact details. You should include at least your name, email and company name. The **telephone** and **facsimile** fields are important if you want us to contact you by those methods. Click **Request License**. You are asked for a license type.



License Type

○ Trial ● Purchased

You must select the type of license you require. If you wish to purchase a full license visit the Easysoft web site www.easysoft.com to obtain a authorization number.

Figure 2.12: License Type Dialog.

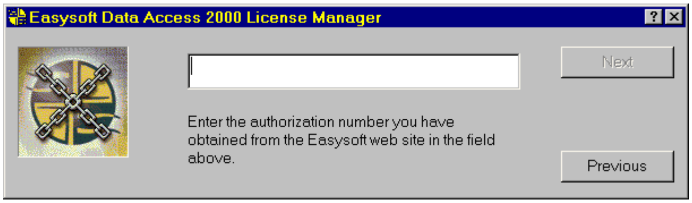
The next step depends on the type of license you want.

18. If you are installing the software on an evaluation basis you should click **Trial**, followed by **Next**. Skip to **step 20**.

    – OR –

    If you have purchased the software and obtained an authorisation code, you should select **Purchase** and click **Next.**

    The License Manager requests your authorisation number.



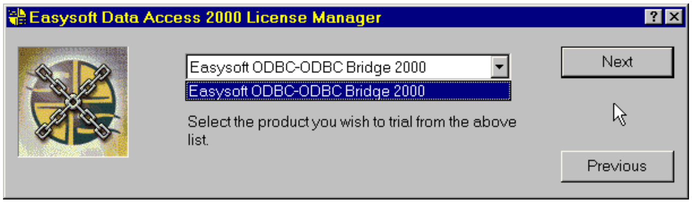
Easysoft Data Access 2000 License Manager

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

Figure 2.13: Authorisation Number Dialog.

19. Enter the authorisation code you received when you completed your purchase, and click **Next**. Skip to **step 21**.

20. The License Manager asks what software you are licensing.



Easysoft Data Access 2000 License Manager

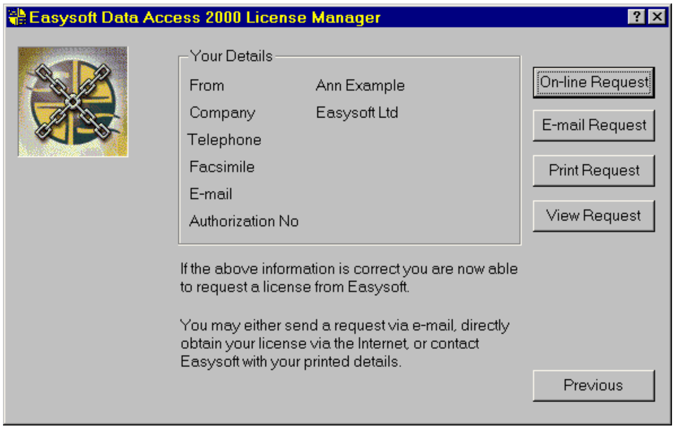
Easysoft ODBC-ODBC Bridge 2000

Easysoft ODBC-ODBC Bridge 2000

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

Check that `Easysoft ODBC-ODBC Bridge` is selected in the drop-down list, and click **Next**.

21. The license manager displays a summary of the information you entered and allows you to choose the method of applying for your license.



Your Details

From	Ann Example
Company	Easysoft Ltd
Telephone	
Facsimile	
E-mail	
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 2.14: License Application Dialog.

Four options are offered here:

- **On-line Request**: This is the quickest method of applying for a license. The License Manager fires off a network packet to the license server at the Easysoft office. The whole process is automatic and invisible.

NB	Only your license request identifier and contact details as they appear in the window are sent to easysoft.

- **E-mail Request**: You can use email to obtain your license. The license server at Easysoft replies immediately with your license number, and you should receive it without delay.
- **Print Request**: Use this option to apply for a license by fax or mail if you do not have an internet connection.
- **View Request**: You can obtain your license by telephoning Easysoft. The **View Request** button displays the license code you need to give to the Easysoft staff.

22. If you choose On-line request, then click **On-line Request** now. The license transaction is complete and you can skip to **step 27**.

    – OR –

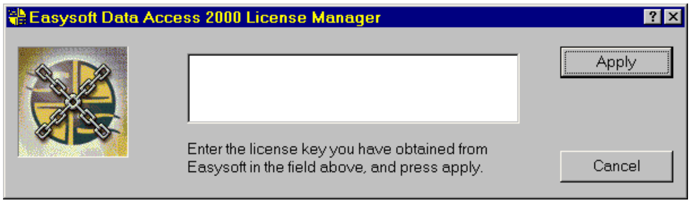
    To use email, fax or phone, click E-mail Request, Print Request or View Request now.

23. Read the information window and close it. You are returned to the license manager window, where you should wait until you receive a reply in order to proceed with the next step.

    – OR –

    If you wish, you can continue with the installation process and return later to license the product, referring to **Appendix B** for instructions.

24. When you receive your License Key(s), click **Enter License.** The following dialog appears:



Enter the license key you have obtained from Easysoft in the field above, and press apply.

Figure 2.15: License Key Dialog.

25. Enter your license key or keys in the box. If you received the license in an email, you can use Windows' Copy and Paste functions. You do not have to be very precise with the mouse, as this dialog will be quite tolerant of spurious characters before and after the license keys.
26. Click **Apply**.
27. Click OK in the dialog that pops up to confirm that the licenses have been added.
28. Click **Finish** in the License Manager window to return to the install program.
29. Click **Finish** again — 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 3-63** for more details.

Uninstalling Under Windows

This section explains how to remove the Easysoft ODBC-ODBC Bridge from your system. This is necessary if you are planning to install a more recent release of the OOB. This does not include service packs.

1. Select **Start > Settings > Control Panel** and then double-click the **Add/Remove Programs** icon.

   You will then be presented with a list of applications that can be automatically removed.

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

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

   The system will then begin to remove all the components. If shared components seem no longer to be required, then you will be prompted to decide whether or not to delete them.

NB	Windows' install/uninstall procedures incorporate 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 the **Yes** or **Yes to all** button to proceed.

   – OR –

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

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.

Installation Under Unix

PREPARATION

Because this section covers a range of platforms, there are many possible paths through the install process. It is worth having a notepad to hand, as information may be given during the install process which is of vital importance to successfully completing the installation. For example, if you are not logged in as the root user whilst installing, then there may be some manual configuration work for the root user to do later.

If you have not already done so, refer to "Obtaining The Software (Windows and Unix)" on **page 2-11**, to get the installation archive onto your system.

For Unix systems, the filename is of the following form:

`odbc-odbc-bridge-`*v*`.`*w*`.`*x*`.`*y*`.`*platform*`.tar`

This filename may be suffixed with `.gz` for a "`gzip`ped" archive, or `.Z` for a "`compress`ed" archive. *platform* may be `aix`, `sunos`, `linux-libc5`, `linux-glibc`, or any other supported platform.

NB	If you downloaded a Unix file using Windows, then there is a good chance that the browser has broken the filename. You can normally work out what is wrong by experimenting with `gunzip`, `uncompress` and `tar`.

1. If you have not already done so, consult "What to Install (Windows and Unix)" on **page 2-10** and "Obtaining The Software (Windows and Unix)" on **page 2-11**, and copy or download the file into a working directory.

   You must have write access to this directory, but it need not be the install directory — in fact, it is better if it is not. It is a good idea to put the archive in a temporary directory which can be removed after installation.

2. Determine whether or not you need root (superuser) access.

   If you are installing the server end of the Easysoft ODBC-ODBC Bridge then root access is required, as the installation script needs to modify the `services` and `inetd` configuration files.

   If you are installing only the *client* end then root access should not be required unless:

   - You are installing under a system that uses the `ld.so` dynamic linker (*e.g.* linux).
   - You wish to configure the OOB Client within unixODBC.

3. Start a terminal session, as root if necessary, and change to the directory the archive file resides in. The next step is to extract the installation files from the archive.

EXTRACTING THE INSTALLATION FILES

4. If the archive has been `gzip`ped (i.e. the filename ends in `.gz`), then use:

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

   – OR –

   If the archive has been `compress`ed (i.e. the filename ends in `.Z`), then use:

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

   – OR –

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

5. Extract the files from the archive:

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

   The `tar` program creates a directory of the same name as the `tar` file (without `.tar`), containing further archives, checksum files, a script called `install` and a text file called `INSTALL`. If you do not wish to keep the original downloaded archive you can now safely delete it.

6. Skim through the `INSTALL` file before continuing. It gives full installation instructions for the Unix-literate. If you are confident in the use and administration of your system, you can skip the rest of this section and rely on the `INSTALL` file.

Caution!

> Be sure to remove any old version of the Easysoft ODBC-ODBC Bridge before proceeding, or there may be conflicts after installation which will prove difficult to resolve.
>
> You will need approximately 5MB of free space for the Easysoft ODBC-ODBC Bridge, plus an additional 3MB for temporary files which can be deleted after installation.

7. There are two license files provided in the archive; one applies if you are installing only the client side, the other if you are installing the server or both the client and the server.

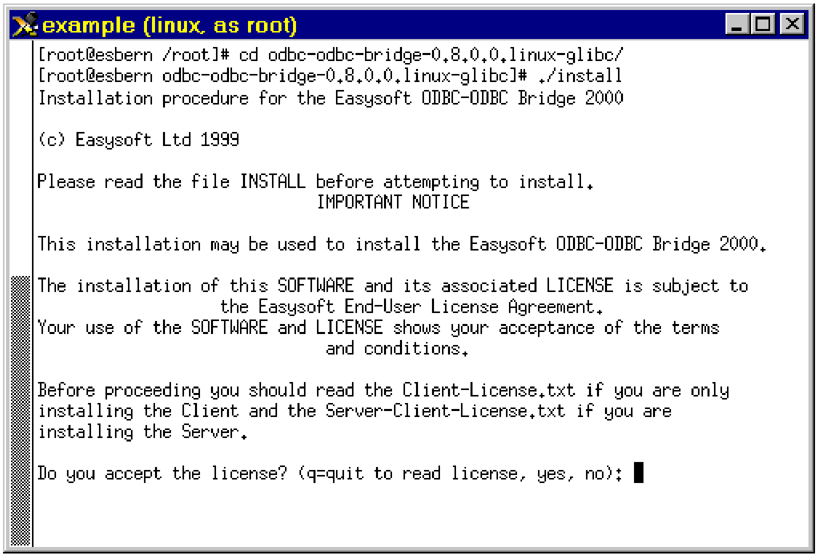
Caution!

> 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 its terms before continuing.

Beginning the Installation

The illustrations in the following section are taken from a session installing the pre-release (version 0.8.0.0) as the root user on a linux/libc5 system. Although the precise output will differ for other platforms, the installation process is essentially the same.

8. Once you have taken care of any matters arising from the `INSTALL` text file, begin the installation:



```
[root@esbern /root]# cd odbc-odbc-bridge-0.8.0.0.linux-glibc/
[root@esbern odbc-odbc-bridge-0.8.0.0.linux-glibc]# ./install
Installation procedure for the Easysoft ODBC-ODBC Bridge 2000

(c) Easysoft Ltd 1999

Please read the file INSTALL before attempting to install.
                             IMPORTANT NOTICE

This installation may be used to install the Easysoft ODBC-ODBC Bridge 2000.

The installation of this SOFTWARE and its associated LICENSE is subject to
                   the Easysoft End-User License Agreement.
Your use of the SOFTWARE and LICENSE shows your acceptance of the terms
                              and conditions.

Before proceeding you should read the Client-License.txt if you are only
installing the Client and the Server-Client-License.txt if you are
installing the Server.

Do you accept the license? (q=quit to read license, yes, no):
```

Figure 2.16: Starting the Installation under Unix.

9. If you have read and agree to the License Agreement, Enter `yes` to continue.

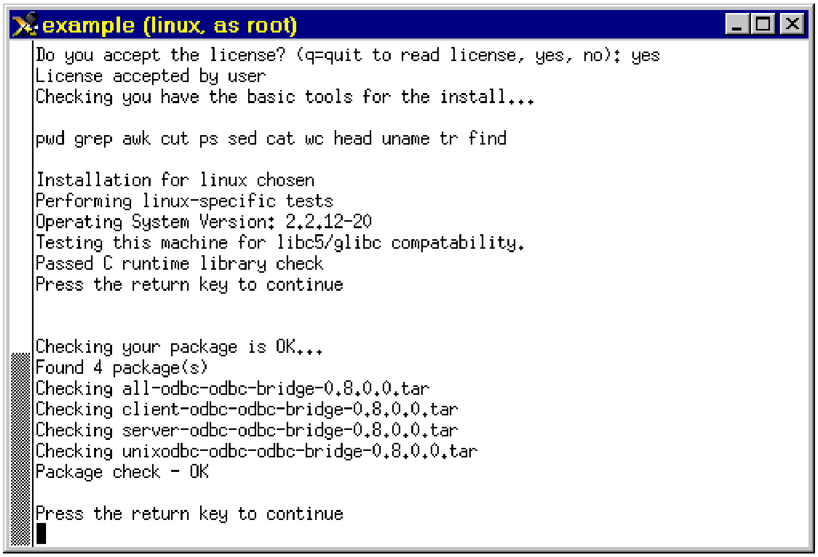
NB	You must type `yes`, not `y`, to continue.

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

    - that you have the minimal set of Unix programs it requires,
    - the platform you are running, and
    - any platform-specific checks, for example the version of the C runtime library.

11. Hit <*Enter*> to continue.

    The script checks the archive package.



```
Do you accept the license? (q=quit to read license, yes, no): yes
License accepted by user
Checking you have the basic tools for the install...

pwd grep awk cut ps sed cat wc head uname tr find

Installation for linux chosen
Performing linux-specific tests
Operating System Version: 2.2.12-20
Testing this machine for libc5/glibc compatability.
Passed C runtime library check
Press the return key to continue


Checking your package is OK...
Found 4 package(s)
Checking all-odbc-odbc-bridge-0.8.0.0.tar
Checking client-odbc-odbc-bridge-0.8.0.0.tar
Checking server-odbc-odbc-bridge-0.8.0.0.tar
Checking unixodbc-odbc-odbc-bridge-0.8.0.0.tar
Package check - OK

Press the return key to continue
```

Figure 2.17: The Script Checks the Archives.

There are three possible outcomes to this check:

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

12. 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. Return to "Obtaining The Software (Windows and Unix)" on **page 2-11** and download the install archive again.

    – OR –

If the files passed the check then read on.

13. Hit <*Enter*> to continue.

    The install script then explains the situation regarding root privileges, *i.e.* if you are not root then:

    - you cannot install the server,
    - you cannot configure unixODBC to support the client, and
    - in linux you cannot configure the dynamic linker to pick up the client.

14. If you are not root then you will be asked whether you want to continue. Enter `y` and read on, or enter `n` to quit.

    – OR –

    If you are root, read on.

    The script then asks 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 do not have root access, or you wish to install the software in another directory, you will need to specify a directory name here.

NB

The installation script will create a directory `easysoft` under the directory you specify here. All the Easysoft ODBC-ODBC Bridge files will be placed in the `easysoft` directory.

The installation script can accept a path to a directory that does not exist, provided its parent 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`.

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

    – OR –

    If you do not have root privileges, or have other reasons for a custom installation directory, type the desired directory and hit <*Enter*>.

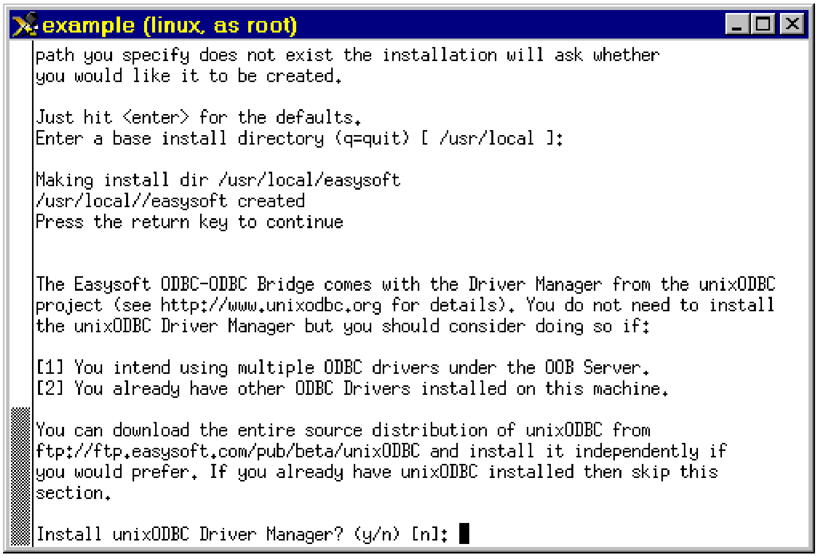
16. The script now creates the `easysoft` directory under the directory you specified. If your chosen directory does not exist, you will be asked whether or not the script should create it. Enter `y` or `n`.

17. If the chosen directory already contains an `easysoft` directory, then the script warns you that you may be installing over a previous installation of the ODBC-ODBC bridge. It asks you whether or not you wish to continue with the installation. Enter `y` or `n`.

18. If you chose a directory other than /usr/local, then the script creates a symbolic link /usr/local/easysoft, pointing to the easysoft directory in the directory you specified.

19. Hit <*Enter*> to continue.

INSTALLING UNIXODBC AND COMMON FILES



```
path you specify does not exist the installation will ask whether
you would like it to be created.

Just hit <enter> for the defaults.
Enter a base install directory (q=quit) [ /usr/local ]:

Making install dir /usr/local/easysoft
/usr/local//easysoft created
Press the return key to continue


The Easysoft ODBC-ODBC Bridge comes with the Driver Manager from the unixODBC
project (see http://www.unixodbc.org for details). You do not need to install
the unixODBC Driver Manager but you should consider doing so if:

[1] You intend using multiple ODBC drivers under the OOB Server.
[2] You already have other ODBC Drivers installed on this machine.

You can download the entire source distribution of unixODBC from
ftp://ftp.easysoft.com/pub/beta/unixODBC and install it independently if
you would prefer. If you already have unixODBC installed then skip this
section.

Install unixODBC Driver Manager? (y/n) [n]:
```

Next, if your platform allows, the script offers to install the unixODBC driver manager. You should normally do this if you are installing the server. You may wish to skip this step if:

- You already have the latest version of unixODBC installed,
- You are using another driver manager, or
- You want to use the system without a driver manager, trading off flexibility of configuration for reduced use of disk space.

20. If you do not wish to install unixODBC, enter `n` and skip to "Installing the Client Side" on **page 2-37**.

    – OR –

    If you wish to install unixODBC, enter `y` at the prompt.

21. The script pauses at this point. Hit <*Enter*> to continue.

    The script extracts the unixODBC files.

22. The script pauses again. Hit <*Enter*> to continue.

    The parts of the OOB installation common to both client and server are extracted.

INSTALLING THE CLIENT SIDE

The script asks you whether you want to install the client side of the bridge. This is required if you want to connect an ODBC client application on the local machine to a remote datasource.

23. Enter `n` to skip the client. Skip to "Installing the Server" on **page 2-39**.

    – OR –

    Enter `y` to install the client. Read on.

24. Press <*Enter*> to continue.

    The script extracts the client files.

25. Press <*Enter*> to continue.

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

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 as 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 you were given in here, then run `/sbin/ldconfig` to re-read the changes.

26. Press <*Enter*> to continue.

CONFIGURING THE CLIENT IN UNIXODBC

The script offers you the chance to set up the client under unixODBC. Doing this permits ODBC applications on the local machine to choose a datasource on the fly. The driver manager loads the required driver (such as the OOB client). This is particularly useful if you intend to run local ODBC datasources as well as connect across the OOB.

27. If you do not want to set the OOB up under unixODBC or you do not have unixODBC, enter `n` and skip to "Installing the Server" on **page 2-39**

    – OR –

    If you have unixODBC installed on your system, and wish to use it with your bridge client, enter `y`.

    The script attempts to run `odbcinst`, the command for installing data sources under unixODBC.

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

    Enter q to abandon the attempt to configure unixODBC, and skip to "Installing the Server" on **page 2-39**.

    – OR –

    If known, supply the unixODBC install path.

    The script checks for the OOB being already set up under unixODBC.

29. If the unixODBC driver manager is already configured for the ODBC-ODBC bridge, then a warning is given. Make a note of this warning.

Caution! If unixODBC has been previously configured with the ODBC-ODBC Bridge then it will continue to pick up the OOB 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 bridge, resulting in incorrect behaviour.

30. The script now calls `odbcinst` with the relevant details for the OOB client. Should this part of the installation fail, you will need to resort to the unixODBC documentation and 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`.

INSTALLING THE SERVER

The script asks you whether you wish to install the server. This allows you to make datasources on the local machine available to remote ODBC clients.

31. If you wish to skip the server install, enter `n` and skip to the end, checking the list on **page 2-41**.

– OR –

Enter `y`, and read on.

32. Hit <*Enter*> to continue.

The necessary files for the server end of the bridge are extracted.

33. Hit <*Enter*> to continue.

34. If you are not root, then the installation terminates. Skip to **step 40**

    – OR –

    If you are root user, the script now begins to install the server.

    It only remains to modify your system's configuration to register the bridge as a service. The script asks whether you would like to install the `services` and `inetd` entries.

35. Enter `n` and the installation terminates. Skip to **step 40**.

    – OR –

    Hit <*Enter*> to continue.

    The script now amends the `/etc/services` and `/etc/inetd.conf` files.

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

    NB You must specify the actual name of the file, not just the location, so if your `inetd.conf` file is in the `/etc/net` directory then you must enter:

    /etc/net/inetd.conf

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

37. If there is no service name conflict, then skip to **step 38**.

    – OR –

    If there is a conflict and you wish to define a new service name for the server, then type `d`, hit <*Enter*>, then enter your chosen new name for the server service.

    – OR –

If there is a conflict and you wish to overwrite the existing service entry, then enter `r`.

The script creates backups of both the `services` and `inetd.conf` files to `services.pre_OOB` and `inetd.conf.pre_OOB`.

38. 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 four-digit port number — the script checks whether the new value conflicts with any other services.

39. If the script cannot determine what shell to use for the `inetd.conf` file, then you must enter one at the prompt.

    Now `inetd` is restarted so that it takes on the changed configuration files, and the server startup script is installed.

COMPLETING THE INSTALLATION

40. Hit <*Enter*> to return to the shell prompt.

    Don't forget:

    - If you were not root, then now is the time to arrange for the dynamic linker to be set up with those paths.
    - If you tried to register the client driver with unixODBC but failed, then don't forget to do this manually.
    - Make a note of the directory under which you installed the software. This will be required when you come to remove the software for upgrade.
    - You may wish to remove the installation files. Unless you specified the installation directory (at **step 15**) to be within the temporary directory, then you can safely remove the temporary directory and all its contents.

Uninstalling Under Unix

To remove the Easysoft ODBC-ODBC Bridge from your Unix system you must manually remove the entries for the server from `services` and `inetd`. This requires root access. If your system has a dynamic linker, such as linux's `ld.so`, then you must also remove the bridge directories from the dynamic linker search path. This may require root access, depending on the mechanism used on your platform. Finally you must remove the install directory tree. This requires the same privileges as the user who performed the installation (in practice, this will often be `root` too). A step-by-step guide follows.

1. Log in as root.
2. If you have the ld.so.conf file, make a backup copy, e.g.

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

3. Copy the `inetd` and `services` configuration files to make backups:

```
# cp /etc/inetd.conf /etc/inetd.conf.safe
# cp /etc/services /etc/services.safe
```

Deregistering the Client and Server Ends

4. Open `/etc/services` using your favourite editor.

Look near the end of the file for a line like this:

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

Again, if you created more than one server then you will have more than one line like this in the file. Each such line should have a comment like that above.

If you have more than one Easysoft ODBC-ODBC Bridge server, then there will be more than one line in the services file. `esoobserver` is the default name for an OOB service, but additional OOB services will have different names. In place of esoobserver will appear whatever name you gave to the additional service: You should make a note of the names of all the services you remove at this stage.

If you do not intend to install a later release of the Easysoft ODBC-ODBC Bridge then you will have to remove all services that were configured for use with it.

5. If you do not intend to re-install the Easysoft ODBC-ODBC Bridge, delete all lines pertaining to all OOB servers and write the file.

   – OR –

   If you are going to upgrade the OOB software and you want the configuration to remain the same, leave the lines in place and close the file.

6. 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
   ```

   esoobserver is the name as specified in the services file; so there should be one entry in inetd.conf for every entry you saw in the services file.

7. If you do not intend to re-install the Easysoft ODBC-ODBC Bridge, remove the lines pertaining to all OOB servers and write the file.

   – OR –

   If you are going to replace the OOB software and you want the configuration to remain the same, leave the lines in place and close the file.

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

    ```
    # kill -HUP pid
    ```

    The next step is to notify the dynamic linker that the shared objects will no longer be available.

Linux	If you have the file `/etc/ld.so.conf`, then manually remove the paths to the OOB's shared objects. The lines will be of the following form: *install-dir*`/easysoft/oob/client`, *install-dir*`/easysoft/oob/lib`, and possibly also *install-dir*`/easysoft/unixODBC`.

Removing the Software

The final task is to remove the software from your system's filestore.

9. Enter the following:

    ```
    # cd install-directory/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.

```
# 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.

```
# cd ..
# ls
```

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.

```
# rm -r easysoft
```

If there are other files in there, i.e. you have another Easysoft product installed too, then you must leave them.

10. If you left the installation files on your system then you may wish to remove them at this point.

    The uninstall process is complete.

CONNECTION 3

Connecting with the Easysoft ODBC-ODBC Bridge

This chapter helps you connect an ODBC-compliant client application through the bridge to a remote datasource.

Since the client and server ends of the OOB will normally be running on different platforms, the chapter is arranged into self-contained sections for easy reference. The approach is the same no matter what platforms are involved:

- Make your ODBC data source visible to the bridge.
- Configure a DSN for the client end of the bridge.
- Connect the client application to the bridge DSN.

If you have an internet connection and want a speedy demonstration, you can skip the server sections and set up a client connection to our server `demo.easysoft.com`. To do this, use the recommended settings in the relevant client subsection.

Chapter Guide

Setting up the OOB Server in Windows

The OOB server for Windows can connect to any System DSN (Data Source Name) configured on your machine, given the necessary information. If you do not already have a DSN on your machine then now is the time to create one. You should use the ODBC driver suitable for your datasource. This section presents a worked example creating a DSN for Microsoft's Northwind database, which is shipped with MS-Access.

You can follow the example as an exercise on your own computer, providing:

- You have Microsoft Access on your machine,
- You have Microsoft's ODBC driver for Access (Almost all Access installations have this)
- You have an Access database to connect to, for example, Microsoft's `northwind.mdb`.

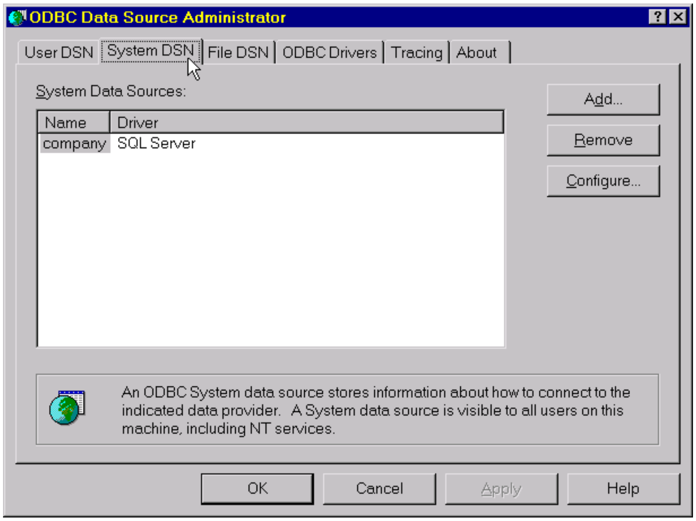
The first step is to open Microsoft's Data Source Administrator.

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 `Datasources (ODBC)`.

The ODBC Data Source Administrator Opens.

2. Click the System DSN tab.



Name	Driver
company	SQL Server

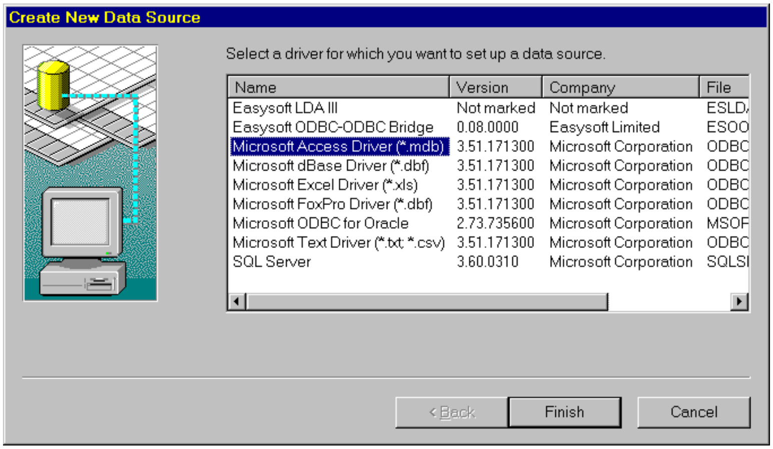
An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

Figure 3.1: ODBC Data Source Administrator — System DSN Tab.

It is important to create a System DSN, as opposed to a User DSN. User DSNs are visible only to the desktop user who created them. Since the bridge server runs as a service, User DSNs are not available to it.

3. Click the **Add...** button to add a new DSN.

The Create New Data Source window appears, containing a list of drivers.



Create New Data Source

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

Name	Version	Company	File
Easysoft LDA III	Not marked	Not marked	ESLD/
Easysoft ODBC-ODBC Bridge	0.08.0000	Easysoft Limited	ESOO
Microsoft Access Driver (*.mdb)	3.51.171300	Microsoft Corporation	ODBC
Microsoft dBase Driver (*.dbf)	3.51.171300	Microsoft Corporation	ODBC
Microsoft Excel Driver (*.xls)	3.51.171300	Microsoft Corporation	ODBC
Microsoft FoxPro Driver (*.dbf)	3.51.171300	Microsoft Corporation	ODBC
Microsoft ODBC for Oracle	2.73.735600	Microsoft Corporation	MSOF
Microsoft Text Driver (*.txt; *.csv)	3.51.171300	Microsoft Corporation	ODBC
SQL Server	3.60.0310	Microsoft Corporation	SQLSI

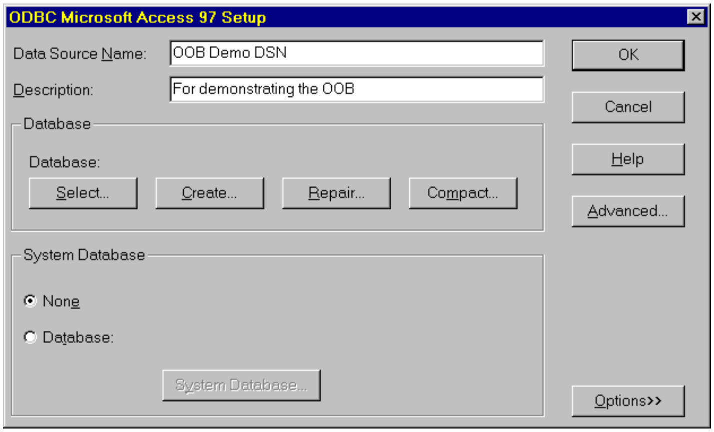
Figure 3.2: The Create New Data Source Window

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

   The ODBC driver for Microsoft Access presents a dialog box for configuring the Data Source. (This step differs from database to database)

   Enter your chosen name for this data source in the Data Source Name box, for example “`OOB Demo DSN`”.

In the Description field, enter something that would help a user faced with a choice of DSNs, for example, "The Bridge’s Target DSN".



ODBC Microsoft Access 97 Setup

Data Source Name: OOB Demo DSN

Description: For demonstrating the OOB

Figure 3.3: The MS Access ODBC driver configured

5. Click **Select...** to browse for the target database. Select your chosen database and click **Ok**. For our example, we used the database in `Microsoft Office\Office\Samples\Northwind.mdb`, but this database may not exist or may be in a different location on your system. Im this case, use any database you have to hand, preferably a small one.

   You should now have a dialog looking much like that of Figure 3.3.

NB	Note the Data Source Name, as it will be required when you come to connect through the bridge.

6. Click **OK**. You are returned to the ODBC Data Source Administrator window. Note:

   - The window now contains a line with your new DSN in it, and
   - The SystemDSN tab should be selected. If it is not, then you must remove the DSN, select the SystemDSN tab, and return to **step 3**.

7. Click **OK**.

   You have set up a System-wide Data Source Name on your machine to a local database, making it visible to the OOB server.

Starting the Server in Windows

For a bridge client to connect to the local DSN, the OOB Server must be running. In Windows NT, the installation program configures the OOB Server to start automatically as an NT Service. For Windows 95, the server must be run manually.

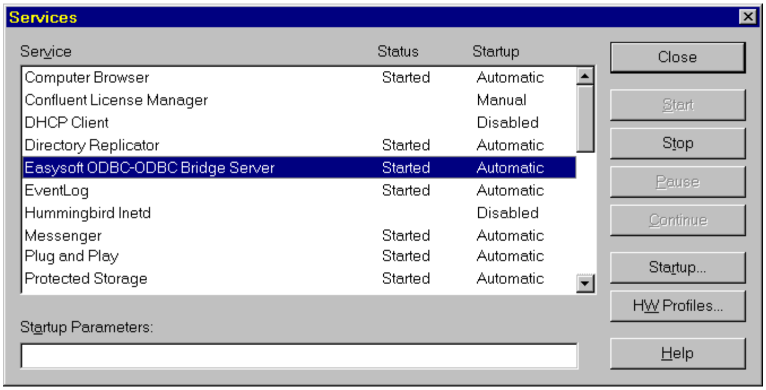
95/98	Choose **Start > Programs > Easysoft > ODBC-ODBC Bridge > Server**.

Checking the Bridge Server Service under Windows NT

This procedure is not normally necessary, but if you are having problems connecting with the bridge, it is a good idea to check through it.

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

You are presented with a dialog box containing all your NT system's registered services. Find the entry for the `Easysoft ODBC-ODBC Bridge Server` (see Figure 3.4).



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

Figure 3.4: The Bridge Server entry in the Services dialog

2. If the entry's `Startup` field says `Disabled`, then you should click **Startup...**. In the resulting dialog box, check the radiobutton marked **Automatic**, and click **OK**.
3. If the entry's `Status` field does not say `Started` then you should click **Start** to bring it on-line.
4. Click **Close**.
5. Close Control Panel.

Conclusion

You now have an ODBC-ODBC Bridge Server running on the Windows machine. You should now log in to the client machine and refer to the relevant section — either **Connecting to the Bridge Client in Windows** below, or "Connecting to the Bridge Client in Unix" on **page 3-65**.

Connecting to the Bridge Client in Windows

This section explains the steps you should take on the client machine for the Windows platforms. It applies to you whether you want to

- connect to the demo.easysoft.com demo server,
- connect to the ODBC-ODBC Bridge server created in the **Setting up the OOB Server in Windows** or **Setting up the OOB Server in Unix** sections, or
- connect to your own OBDC-ODBC Bridge server.

You should be sure which of the above you are attempting before you begin this section.

The Bridge Client Driver appears to the Client Application just as any other ODBC driver. For this reason many of the steps in this section are similar to those of the section **Setting up the OOB Server in Windows** above. The first step is to open Microsoft's Data Source Administrator.

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

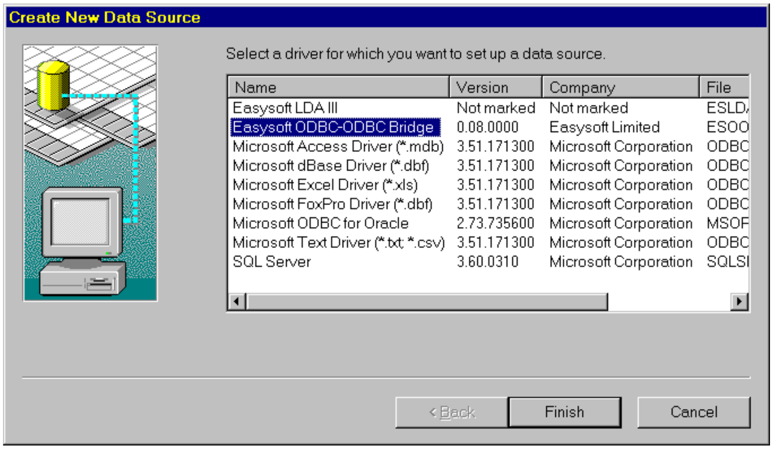
2000	To find the ODBC icon in Windows 2000, open `Administrative Tools` in Control Panel. The ODBC icon is called `Datasources (ODBC)`.

The ODBC Data Source Administrator Opens.

NB	Do not click the **System DSN** tab for the client DSN. DSNs created for the server end must be system DSNs, but client-end DSNs should normally be user DSNs.

2. If the **User DSN** tab is not selected, select it now.
3. Click the **Add...** button to add a new DSN.

The Create New Data Source window appears, containing a list of drivers.



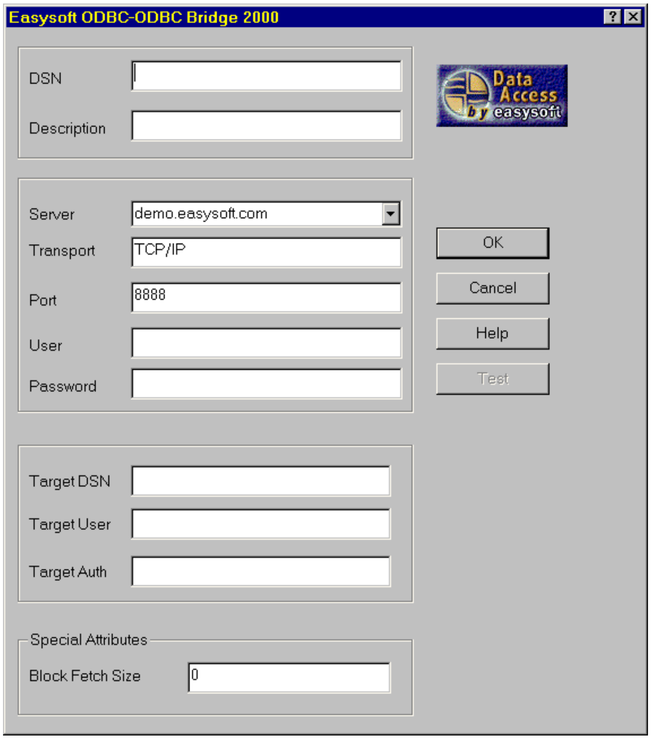
Select a driver for which you want to set up a data source.

Name	Version	Company	File
Easysoft LDA III	Not marked	Not marked	ESLD/
Easysoft ODBC-ODBC Bridge	0.08.0000	Easysoft Limited	ESOO
Microsoft Access Driver (*.mdb)	3.51.171300	Microsoft Corporation	ODBC
Microsoft dBase Driver (*.dbf)	3.51.171300	Microsoft Corporation	ODBC
Microsoft Excel Driver (*.xls)	3.51.171300	Microsoft Corporation	ODBC
Microsoft FoxPro Driver (*.dbf)	3.51.171300	Microsoft Corporation	ODBC
Microsoft ODBC for Oracle	2.73.735600	Microsoft Corporation	MSOF
Microsoft Text Driver (*.txt; *.csv)	3.51.171300	Microsoft Corporation	ODBC
SQL Server	3.60.0310	Microsoft Corporation	SQLSI

Figure 3.5: The Create New Data Source Window, with the Bridge Driver selected

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

The bridge driver pops up a configuration dialog box.



Easysoft ODBC-ODBC Bridge 2000

DSN
Description

Server demo.easysoft.com
Transport TCP/IP
Port 8888
User
Password

Target DSN
Target User
Target Auth

Special Attributes
Block Fetch Size 0

Figure 3.6: The ODBC-ODBC Bridge Driver Configuration Dialog

The text boxes on the dialog are split into four sections, arranged by functionality, from top to bottom:

- How the DSN appears to the driver manager and ODBC client,
- How to attach to the bridge server on the remote machine,
- The DSN to attach to on the remote machine through the server, and
- Special settings for tuning.

5. Enter your chosen Data Source Name for this data source, for example `demo`. Choose carefully, as you will not be able to change this after hitting **OK**.

NB	If you intend to use the `demo.exe` client application then the DSN *must* be named `demo`.

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

SPECIFYING THE REMOTE MACHINE

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

8. The **Transport** box allows you to specifiy 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 happen to know that the server is listening at some other port.

   The next two boxes are for the logon account and password for the machine on which the server is running. The bridge server carries out all activities as that user.

10. Enter the user name and password of a valid logon account on the server host in the **User** and **Password** boxes, if required.

    – 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 name of the DSN you created on the remote machine.

    – OR –

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

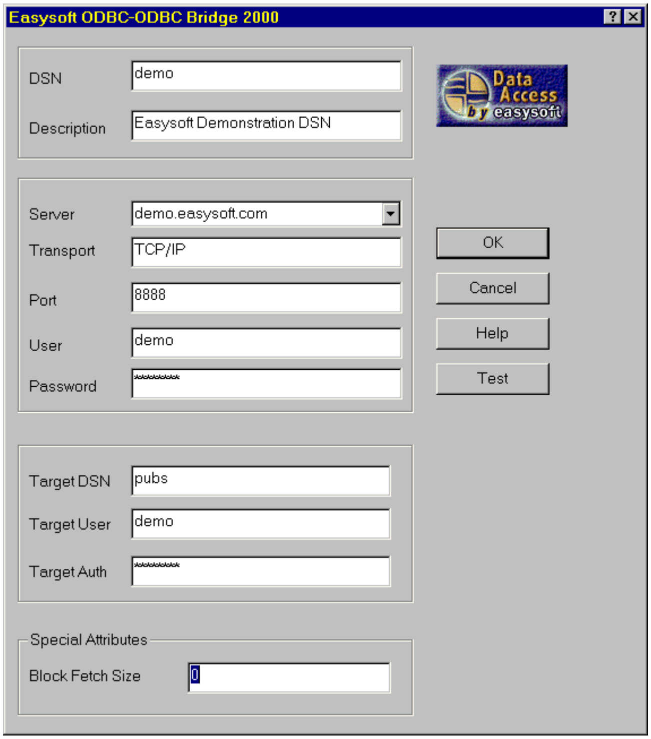
12. If your remote data source (i.e. the database itself) requires a user name and password apart from the user logon account for the machine, then enter these in **TargetUser** and **TargetAuth**.

    – OR –

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

    – OR –

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



Easysoft ODBC-ODBC Bridge 2000

DSN: demo

Description: Easysoft Demonstration DSN

Server: demo.easysoft.com

Transport: TCP/IP

Port: 8888

User demo

Password: ********

Target DSN: pubs

Target User: demo

Target Auth: ********

Special Attributes

Block Fetch Size: 0

Figure 3.7: The Driver set up for `demo.easysoft.com`

CHECK YOUR VALUES

13. Ignore the special attributes section and click **Test**.

    This attempts to connect using the information on the form and send an ODBC request, displaying the results in a window.

14. If you see an error message, then you need to correct some of the fields in the configuration dialog. The messages are fairly self explanatory, but note the following:

    - 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 it is **TargetUser** and **TargetAuth**.

- Any reference to data sources or DSNs refers to the Target DSN.
- Messages refering to RPC mean that the client cannot find the bridge server. Check **Server** and **Port**. **Transport** has no effect at this stage since only TCP/IP is supported.
- **DSN** and **Description** have no effect on the test procedure.

– OR –

If you see a lot of lines starting with SQL then you have connected to the remote machine. Click **OK** in the test box and **OK** in the configure dialog.

The connection has been made.

Connecting a Client Application in Windows

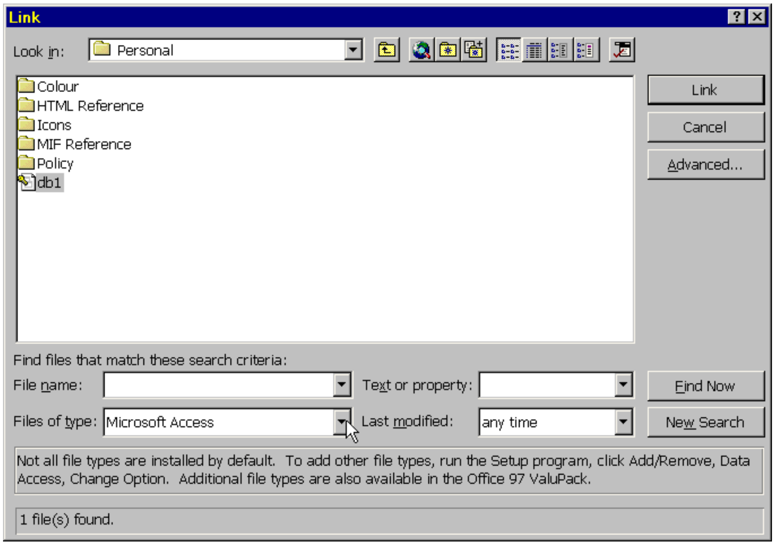
By this stage, you should have a DSN on your Windows machine that connects through the bridge to a second DSN on the remote machine — either your own server or our `demo.easysoft.com`. To demonstrate this to yourself, you can connect an ODBC application to the local DSN. Microsoft Access is used for this exercise, as most

Windows users have a copy. You must be comfortable with basic use of MS Access to proceed with this section.

Caution!

> If you are using the bridge across the internet — for example to contact the Easysoft `demo` server — then Microsoft Access is *not* recommended. MS Access is very heavy in its use of ODBC calls, and when using the bridge each of those calls generates network traffic. Skip to the next subsection, “The Demo.exe Client” on **page 3-63** for a more efficient client program.

1. Start Microsoft Access 97 (for example) and create a blank database.
2. Select **File > Get External Data > Link Tables** to connect to the sample database.



Link

Look in: Personal

Colour
HTML Reference
Icons
MIF Reference
Policy
db1

Find files that match these search criteria:

File name:

Text or property:

Files of type: Microsoft Access

Last modified: any time

Not all file types are installed by default. To add other file types, run the Setup program, click Add/Remove, Data Access, Change Option. Additional file types are also available in the Office 97 ValuPack.

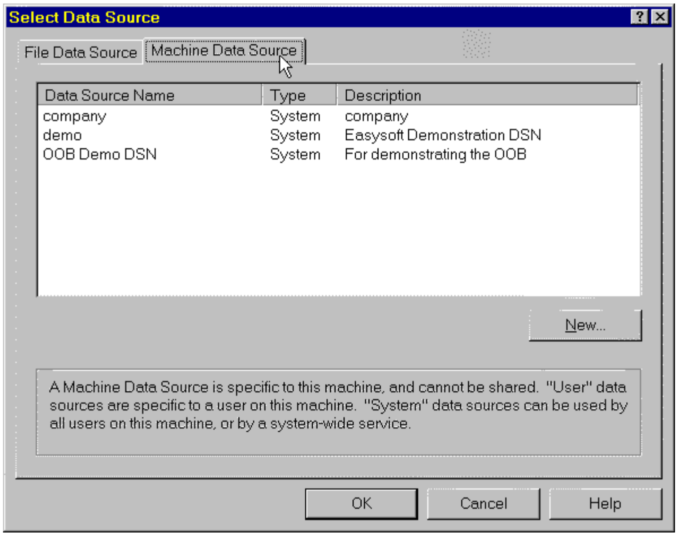
1 file(s) found.

Figure 3.8: The Link Tables Dialog in MS Access

Access presents the Link window, showing existing databases on your system.

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

Microsoft's ODBC driver manager presents the Select Data Source window.



Data Source Name	Type	Description
company	System	company
demo	System	Easysoft Demonstration DSN
OOB Demo DSN	System	For demonstrating the OOB

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 3.9: Machine Data Source tab in the Select Data Source dialog

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

   Find the local DSN you created, somewhere in the list. Note that the description you gave it appears beside it.

5. Select your DSN and click **OK**.

   Access 97 interrogates the bridge, which relays the questions to your remote data source. Access presents the Link Tables window, showing a list of available datasets.

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

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

7. Double-click on the table to open and browse it.

THE DEMO.EXE CLIENT

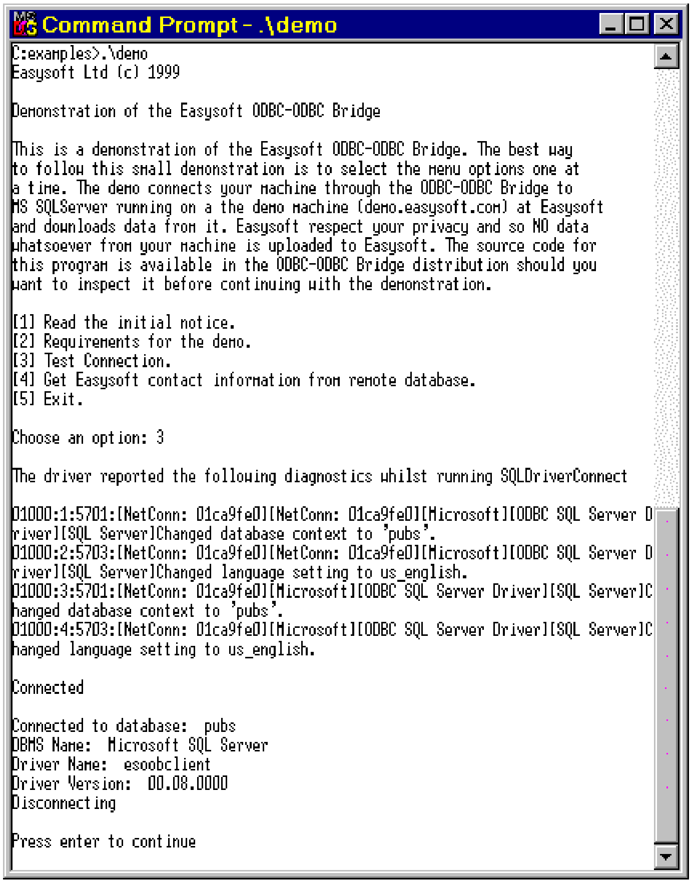
In the distribution archive, we include a lightweight client for connecting to a datasource at `demo.easysoft.com`. This program can be used in the event that your own remote ODBC datasource is not yet configured. The source code is also included in the demonstration program to give developers a kickstart into creating simple ODBC client applications.

NB	The `demo.exe` client application will not work without the local DSN configured. Refer to the section “Connecting to the Bridge Client in Windows” on **page 3-54** and create the demo DSN. Note that the DSN must be named `demo` for `demo.exe` to work.

1. Start an MS-DOS window.
2. Change directory to the `Examples` directory within the easysoft install directory, for example:

```
C:>cd Program Files\Easysoft\
     Easysoft ODBC ODBC Bridge\Examples
```

3. Execute the `demo.exe` program.



```
C:examples>.\demo
Easysoft Ltd (c) 1999

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 select the menu options one at
a time. The demo connects your machine through the ODBC-ODBC Bridge to
MS SQLServer running on a 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] Test Connection.
[4] Get Easysoft contact information from remote database.
[5] Exit.

Choose an option: 3

The driver reported the following diagnostics whilst running SQLDriverConnect

01000:1:5701:[NetConn: 01ca9fe0][NetConn: 01ca9fe0][Microsoft][ODBC SQL Server D
river][SQL Server]Changed database context to 'pubs'.
01000:2:5703:[NetConn: 01ca9fe0][NetConn: 01ca9fe0][Microsoft][ODBC SQL Server D
river][SQL Server]Changed language setting to us_english.
01000:3:5701:[NetConn: 01ca9fe0][Microsoft][ODBC SQL Server Driver][SQL Server]C
hanged database context to 'pubs'.
01000:4:5703:[NetConn: 01ca9fe0][Microsoft][ODBC SQL Server Driver][SQL Server]C
hanged language setting to us_english.

Connected

Connected to database:  pubs
DBMS Name:  Microsoft SQL Server
Driver Name:  esoobclient
Driver Version:  00.08.0000
Disconnecting

Press enter to continue
```

Figure 3.10: The demo.exe Progam in Action

The demonstration program is self-explanatory and extremely simple. Options 3 and 4 contact demo.easysoft.com through the bridge; option 4 actually retrieves a table from our database.

Setting up the OOB Server in Unix

At the time of writing, there is little demand for users to be able to configure ODBC data sources for the OOB Server on the Unix platform. Forthcoming Easysoft products will configure the OOB automatically where required without user intervention. If you are interested in linking the OOB server to your own ODBC drivers, please contact us and we will be glad to provide information and support.

Connecting to the Bridge Client in Unix

Since Unix does not yet have a standard driver manager, the OOB 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 client connects to an ODBC 3.0 datasource.

The most important driver manager functionality the OOB provides, however, is support for storing DSN attributes in order to be able to connect given a minimal connection string. When an application connects through the client, it can pass in as little information as just a

DSN. The OOB client will search for a configuration file in the following places, in order:

- `$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-dir*`/odbc.ini`
- *home-dir*`/.odbc.ini`
- `/etc/odbc.ini` — for system DSNs only

NB If you are running Apache/PHP with the OOB, Easysoft recommend you use `/etc/odbc.ini`. Putting the file in the Apache home directory risks making it publicly accessible, and putting it in the ‘current directory’ may cause problems when starting Apache in a different directory.

THE FORMAT OF THE ODBC.INI FILE

Each section of the odbc.ini file starts with a DSN in brackets (a.k.a. "Square Brackets"). The following lines are *attribute=value* pairs, which are passed to the datasource on the remote machine, with some important exceptions:

Name	**Description**
`Description`	A single line of text to describe the datasource to users.
`Server`	The name or IP address of the remote host where the OOB server is running.
`Port`	The port on which the OOB server is listening.

Name	Description
`Transport`	The network transport used across the bridge. Must be set to `tcpip`
`LogonUser`	The name of a user on the remote machine. The OOB server changes to this user when an incoming connection is made.
`LogonAuth`	The password for LogonUser
`TargetDSN`	The DSN on the remote machine
`TargetUser`	If present, this is passed to the database engine as the `UID` attribute. This is only required if you have user access control on your database.
`TargetAuth`	The password for TargetUser
`BlockFetchSize`	A value of 0 means ‘do not perform block fetches’. Values from 1 to 100 specify the number of rows that should be fetched at a time. You should read FAQ section on block fetch mode before using this.
`Unquote_Catalog_Fns`	If present, fixes a bug in Applixware. See the FAQ
`MetaData_ID_Identifier`	If present, fixes a bug with some older applications which accidentally use wildcards in catalog functions.

Note that the Attribute names are not case-sensitive. There are four classes of attribute here:

- `Description` is effectively a comment, though it may be displayed to users by a GUI-enabled driver manager should you have one installed.
- `Server`, `Port`, `Transport`, `LogonUser` and `LogonAuth` are concerned with finding the remote machine and connecting to the OOB server. Imagine a telnet session:

  `$ telnet` *server port*

login *logonuser*

password: *logonauth*

- TargetDSN, TargetUser and TargetAuth are renamed DSN, UID and PWD respectively, and repeated by the server to its driver or driver manager. Imagine a database session once you are logged in to the remote machine.

  $ isql *targetdsn targetuser targetauth*

- BlockFetchSize, Unquote_Catalog_Fns and MetaData_ID_Identifier are all flags to change the bridge's behaviour according to the specific application.

> **NB**
>
> You must provide the bridge with Server, Port, Transport, LogonUser, LogonAuth and TargetDSN. This is enough information to allow the server end to present the DSN to its driver or driver manager.
>
> Don't forget, any attributes not specific to the bridge are passed through to the remote data source, so you can effectively set up the remote datasource from the local machine.

EXAMPLE ODBC.INI FILE

As an example, let us say you are on a linux box called linus.mydomain. You have MS SQLServer and the OOB server running on a remote Windows NT machine called ntbox.mydomain. Your Windows NT user is myname and your password is mypassword.

Imagine you have set up a datasource on ntbox (refer to "Setting up the OOB Server in Windows" on page 3-48) called myNTdsn which requires database authentication dbuser and dbpassword. You want to access data in MS SQLServer on `ntbox` from `linus` using your Perl program or some PHP.

Your `odbc.ini` file would contain a section like this:

```
[localdsn]

Server=ntbox.mydomain.com

Port=8888

Transport=tcpip

LogonUser=myname

LogonAuth=mypassword

TargetDSN=myNTdsn

TargetUser=dbuser

TargetAuth=dbpassword
```

When your application connects through the OOB Client, it needs to pass in the DSN `localdsn`. The OOB client uses this DSN to access the correct section in the `odbc.ini` file.

The Demo Client

In the distribution archive, we include a lightweight client for connecting to a datasource at `demo.easysoft.com`. This program can be used in the event that your own remote ODBC datasource is not yet configured. The source code is in the `examples` subdirectory. To build it you need to find a suitable makefile in the `examples` directory:

```
$ cd InstallPath/easysoft/oob/examples

$ ls Make*

Makefile.linux
```

You must then edit the chosen makefile to remove the comment from the second line, and make sure that the path pointed to is the correct

InstallPath, for instance, `#INSTALLPATH=/usr/local` might become `INSTALLPATH=/opt/` — note that the `#` is deleted whether or not you change the directory.

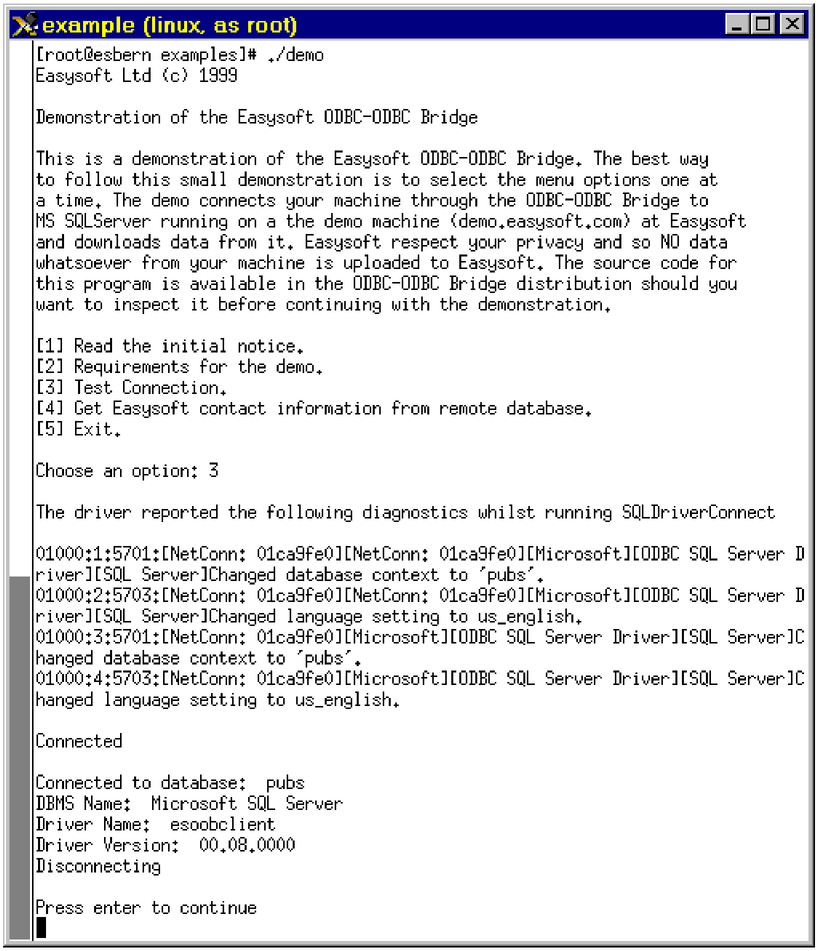
You can then `make` the examples:

```
$ make -f Makefile.linux
...
```

...and run the demo program:

```
$ ./demo
```

The demonstration program is self-explanatory and extremely simple. Options 3 and 4 contact demo.easysoft.com through the bridge. Option 4 actually retrieves a table from our database.



```
[root@esbern examples]# ./demo
Easysoft Ltd (c) 1999

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 select the menu options one at
a time. The demo connects your machine through the ODBC-ODBC Bridge to
MS SQLServer running on a 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] Test Connection.
[4] Get Easysoft contact information from remote database.
[5] Exit.

Choose an option: 3

The driver reported the following diagnostics whilst running SQLDriverConnect

01000:1:5701:[NetConn: 01ca9fe0][NetConn: 01ca9fe0][Microsoft][ODBC SQL Server D
river][SQL Server]Changed database context to 'pubs'.
01000:2:5703:[NetConn: 01ca9fe0][NetConn: 01ca9fe0][Microsoft][ODBC SQL Server D
river][SQL Server]Changed language setting to us_english.
01000:3:5701:[NetConn: 01ca9fe0][Microsoft][ODBC SQL Server Driver][SQL Server]C
hanged database context to 'pubs'.
01000:4:5703:[NetConn: 01ca9fe0][Microsoft][ODBC SQL Server Driver][SQL Server]C
hanged language setting to us_english.

Connected

Connected to database:  pubs
DBMS Name:  Microsoft SQL Server
Driver Name:  esoobclient
Driver Version:  00.08.0000
Disconnecting

Press enter to continue
█
```

CONFIGURATION

Configuring the Easysoft ODBC-ODBC Bridge

This chapter covers configuring the bridge at a lower level, for example to tune the system or work around bugs in certain database engines. For a straightforward explanation of how to configure a bridge DSN to connect to a remote data source, see **Chapter 3, Connection**.

Chapter Guide

ODBC Driver Managers

In Windows, a configured ODBC-ODBC Bridge DSN appears to the client program as an ordinary DSN, connected through the driver manager like any other ODBC driver. On non-Windows systems, which do not necessarily have a driver manager installed, it can be configured with or without one.

At the server end, the OOB acts like an ordinary ODBC-compliant application. In Windows NT, it takes the form of a network service, configured to start automatically. In Windows 95 it must be started manually or by placing it in the `Startup` menu directory. Under Unix, it is called by `inetd` and is linked to the driver or driver manager of your choice.

In general, ODBC applications must be linked with either an ODBC Driver or a Driver Manager. The Windows platform provides a single driver manager, to which programmers link their code (`odbc32.dll`). When a program calls `SQLDriverConnect` (or `SQLConnect`), it passes in a *connection string*, which normally contains at least a Data Source Name, or *DSN*. The driver manager examines the connection attributes and looks up the required driver from the DSN or the Driver name, and then dynamically loads the driver DLL. From then on the driver manager relays ODBC calls to the driver, and passes the result back to its caller.

There is no standard driver manager for Unix at present, although at the time of writing Easysoft know of two driver managers distributed as free software under the GNU Lesser General Public License.

REF The GNU Lesser General Public License (LGPL) basically protects your right to give and receive free copies of the library, whilst allowing you to ship commercial products that rely on it. For full details see **http://www.gnu.org/copyleft/lesser.html**

If your client application runs on Unix and the OOB Client driver will be the only ODBC driver used on that machine, then you do not need a driver manager and you can link your application directly with the OOB. An example of this is using Perl in Linux to access a Windows ODBC driver across the OOB.

If you already have a driver manager or would like to use more than one ODBC driver on a Unix system then you should set up a DSN for the OOB in the driver manager. How you do this depends on the driver manager.

Easysoft recommend that you use the unixODBC driver manager for Unix platforms in the first instance. Other driver managers are available, but at the time of writing, unixODBC is demonstrably the most flexible and reliable open source solution we have seen. unixODBC is not an Easysoft product, but is available free from the Easysoft web site and is distributed with the OOB.

REF The unixODBC driver manager is included with the OOB distribution (see **http://www.easysoft.com/**)

http://www.unixodbc.org/ is the home page for unixODBC. The distribution found here contains the unixODBC driver manager plus a set of ODBC drivers.

The OOB Server under Windows 95/98

For Microsoft Windows, the OOB Server is distributed as an executable linked with Microsoft's driver manager. In Windows 2000 or Windows NT it is a network service configured to be started automatically by the Service Manager when Windows starts. In other Win32 environments it must be started manually or by placing it in the Startup menu directory:

1. Choose **Start > Settings > Taskbar & Start Menu**.
2. Select the **Start Menu Programs** tab.
3. Click **Add...**
4. Type `\`*InstallPath*`\esoobserver standalone` in the **Command Line** box, substituting the full path to the `esoobserver.exe` program for *InstallPath*.
5. Click **Next**.

   Windows displays the Start menu's folder hierarchy.
6. Navigate to `Programs/Startup`.
7. Click **Next**.

   Windows suggests the name `esoobserver` for the menu entry.
8. Change the name if you wish, and click **Finish**.
9. click **Ok** in the Taskbar Properties dialog to close it.

   You have now added the OOB server to your startup menu. It will be run automatically when you start Windows.

   In Windows 95 and 98, the Easysoft OOB Server runs in a DOS box which starts minimised by default. It will not normally be necessary to display the window, but any error messages will be written in that window.

NB

The server runs with privileges of the user who started it. This means that connecting clients will only be able to see data sources to which you have access.

To run the Easysoft OOB Server under Windows 95 you will require Winsock 2. This can be obtained from **http://www.microsoft.com/**

The OOB Server under Windows 2000/NT

The OOB Server under Windows NT is installed as a network service, configured to be started automatically by the NT Service Manager when the machine boots.

The server runs with administrative privileges, allowing it access to any data source on the system. When a user connects, the OOB Sever effectively becomes the user specified in the `LogonUser` attribute so that it acts with the specified user's permissions.

For Windows NT and Windows 2000 we can configure the service like any other network service — for instance, we can set it up not to start automatically on system boot. We can also explicitly stop the service.

1. Choose **Start > Programs > Control Panel**.
2. Open the `Services` icon.

The service manager presents a list of services, along with their status (i.e. whether "Started" or not), and their startup configuration (Manual, Automatic, or Disabled). Refer to the relevant subsection below.

To Start or Stop the Service in Windows NT

3. In the Services dialog, ensure that the `Easysoft ODBC-ODBC Bridge Server` entry is selected.

4. 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.

5. In the Services dialog, ensure that the `Easysoft ODBC-ODBC Bridge Server` entry is selected.

6. Click **Startup...** (*not* **Start**!) to open the Service configuration dialog.

7. In the **Startup Type** section, choose the relevant radiobutton:

   - 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**.

8. Click **Ok** to close the dialog.

USER ACCOUNTS AND THE SERVER

Windows NT's server model distinguishes two types of service with respect to their privileges. Services running in the System Account have access to all resources on the local machine whilst idle. When the OOB Server runs in this mode, an authentication process takes place when a user connects using the `LogonUser` and `LogonAuth` attributes. The server then performs all actions with the privileges of the user specified in `LogonUser`.

This is the default configuration of the OOB Server, and it provides the greatest general flexibility.

The other option is to set up the server to always run with a specific user's permissions. This has two key effects, each of which may be useful on its own:

- When the server is running with administrative privileges, and switches to another user on connect, some database systems (notably SQL Server) grant greater permissions than they would if the server had been running as that user throughout. Setting the server to run as a normal user tightens up access control in DBMSes that behave this way.

- Windows NT's authentication procedure, while quite satisfactory for a human user, is often too slow for an intensive application like recording website hits in real time. By setting up the server to run with a specific user's attributes, the authentication process is only executed at server start time.

NB	If the server is set up to always run as a particular user, then you should not use `LogonUser` or `LogonAuth` when establishing a connection.

Whichever purpose you have in mind, the procedure is the same:

1. 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.
2. In the Services dialog, ensure that the `Easysoft ODBC-ODBC Bridge Server` entry is selected.
3. Click **Startup...** (*not* **Start**!) to open the Service configuration dialog.
4. In the **Log On As** section, select the **This Account** radiobutton.
5. In the **This Account** text box, type the user name or click **...** to browse for a user name.
6. In the **Password** box, type the user account password.
7. Type the same password in the **Confirm Password** box.
8. Click **Ok** to close the dialog and commit your changes.

   You may have noticed the **Allow Service to Interact With Desktop** checkbox. This has no effect on the OOB Server.

The HTTP Configuration Interface

There are a number of configurable parameters in the server, irrespective of platform. In Windows, the parameters are stored in the registry but the server provides an HTTP interface to avoid editing the registry by hand.

This subsection provides a tour of the HTTP interface and shows you where to go to change the server-configurable parameters. The parameters themselves are documented in "Server Configurable Parameters (Windows and Unix)" on **page 4-92**.

In order to follow this subsection you will need a Windows machine running the OOB server, and you will need to know the OOB Administrator's NT user name (HTTPAdmin) and password.

NB

If you do not know the HTTPAdmin user name for your OOB server, then you must browse the registry after all! Run `regedit.exe` and look in:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Easysoft ODBC
-ODBC Bridge\Configuration\System\setting
s
```

for the `HTTPAdmin` setting. This is the user whose password you need to know to make changes through the HTTP interface. If this value is not a valid NT user, then you must change it. The value is case sensitive.

1. Start a web browser and open the URL **http://localhost:8890/**. Depending on your individual machine settings, you may need to

specify your machine’s actual name in place of localhost, or you may need to use a different port address than the Easysoft default of 8890.

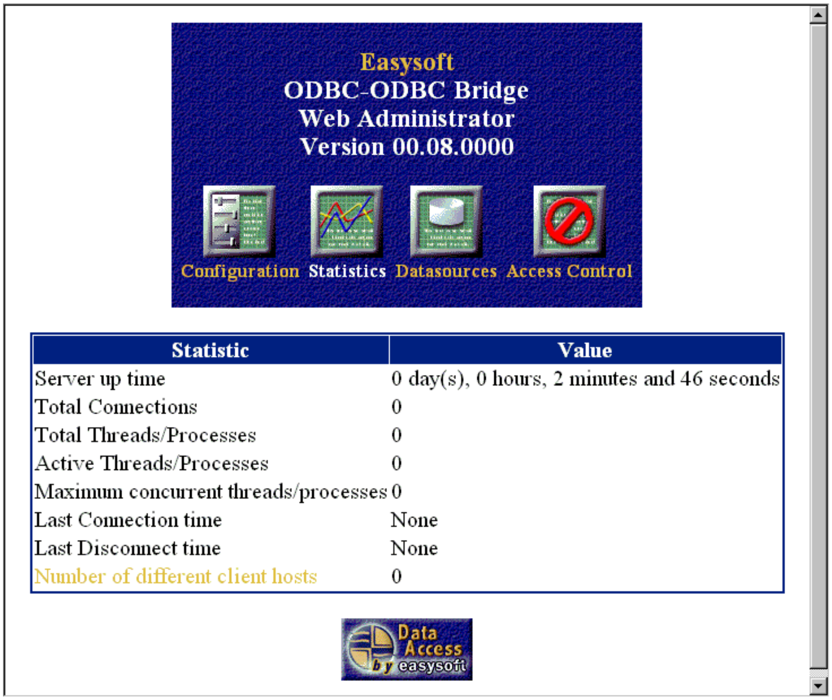
NB

If you forget the HTTP port setting for your OOB server you must look in the registry. Run `regedit.exe` and look in:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Easysoft ODBC
-ODBC Bridge\Configuration\System\setting
s
```

for the `HTTPPort` setting. This is the port you should connect to.

The web page returned is generated by the server process, and displays runtime statistics for the latest run of the server. There are four large icons at the top of the screen, marked Configuration, Statistics, Datasources and Access Control.



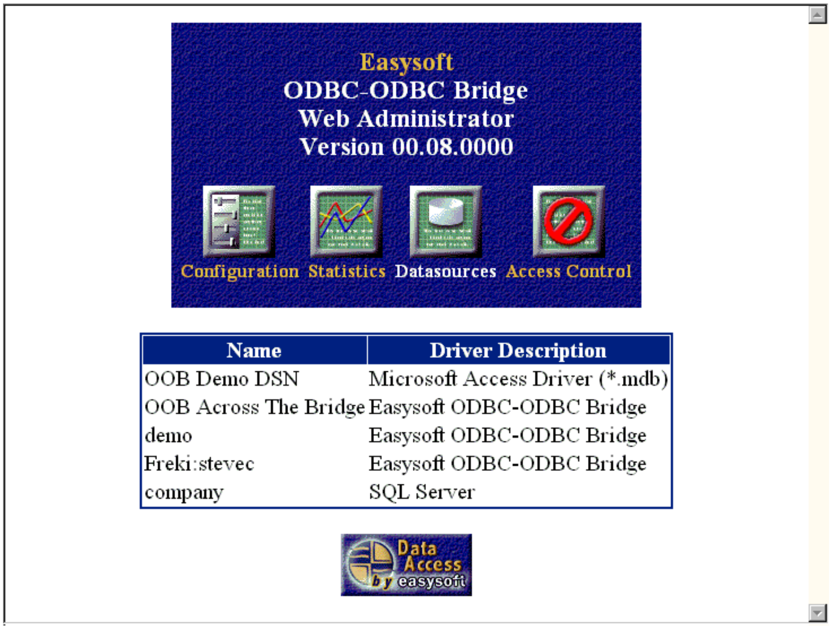
Statistic	Value
Server up time	0 day(s), 0 hours, 2 minutes and 46 seconds
Total Connections	0
Total Threads/Processes	0
Active Threads/Processes	0
Maximum concurrent threads/processes	0
Last Connection time	None
Last Disconnect time	None
Number of different client hosts	0

Figure 4.1: The Statistics Screen of the HTTP Interface

Statistics (the current screen) and Datasources are only used to display information, but the Configuration and Access Control screens allow you to modify the server’s runtime parameters.

2. Click on the **Datasources** icon in the browser window.

The screen displays the ODBC Datasources found by the server, and their applicable drivers .

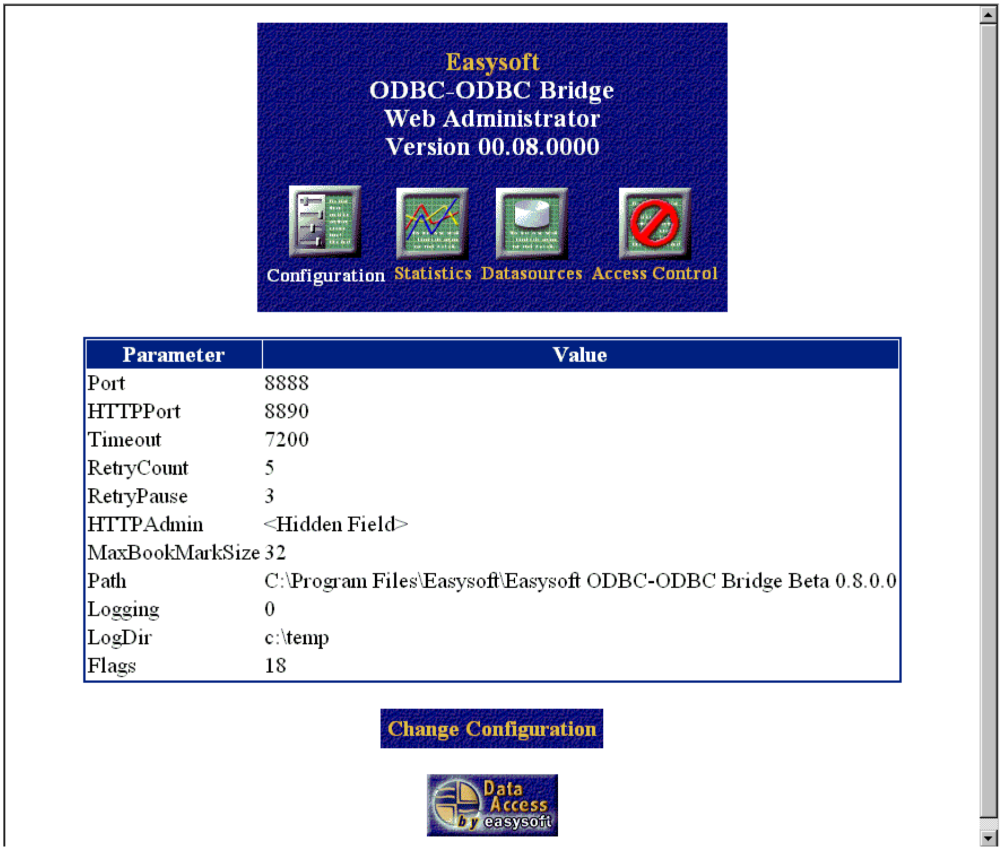


Name	Driver Description
OOB Demo DSN	Microsoft Access Driver (*.mdb)
OOB Across The Bridge	Easysoft ODBC-ODBC Bridge
demo	Easysoft ODBC-ODBC Bridge
Freki:stevec	Easysoft ODBC-ODBC Bridge
company	SQL Server

Figure 4.2: The Datasources Screen of the HTTP Interface

3. Click **Configuration**.

The HTTP Interface shows the configurable parameters of the server process.



Parameter	Value
Port	8888
HTTPPort	8890
Timeout	7200
RetryCount	5
RetryPause	3
HTTPAdmin	<Hidden Field>
MaxBookMarkSize	32
Path	C:\Program Files\Easysoft\Easysoft ODBC-ODBC Bridge Beta 0.8.0.0
Logging	0
LogDir	c:\temp
Flags	18

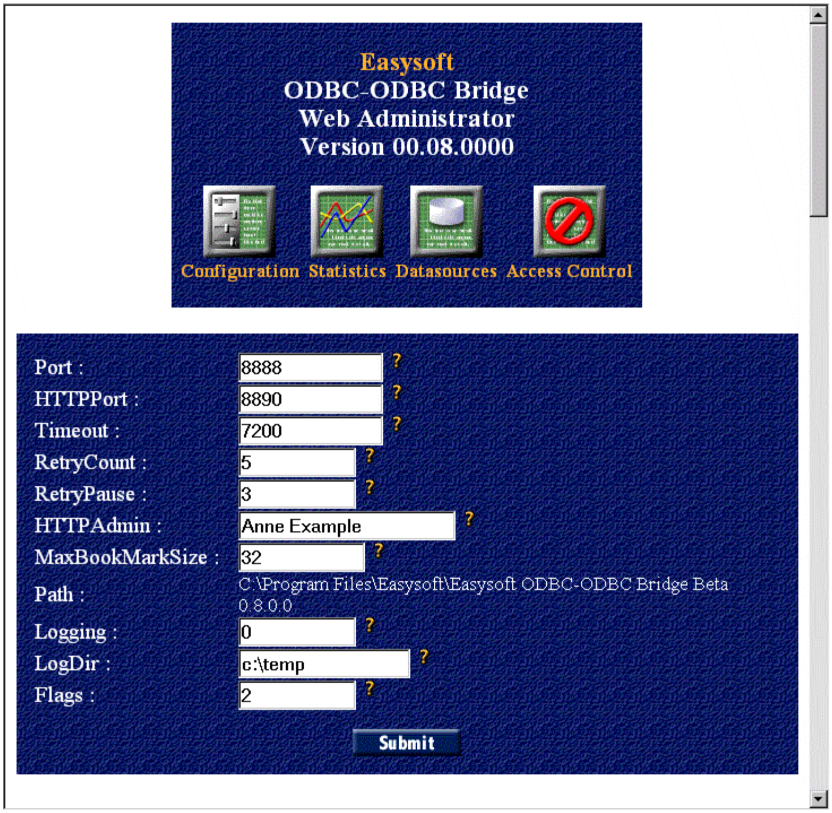
Figure 4.3: The Configuration Screen of the HTTP Interface

This screen shows the configuration state of the server process. There is also a link, **Change Configuration**, to a form that allows you to modify the server settings.

4. Click **Change Configuration**. The browser prompts you for a user name and password.

5. Type the HTTPAdmin user name and password. This is the NT user name that you typed during the install process in the Server

Administrator’s Username Dialog. (**Figure 2.9** on **page 2-20**). The Change Configuration screen is displayed.



Easysoft
ODBC-ODBC Bridge
Web Administrator
Version 00.08.0000

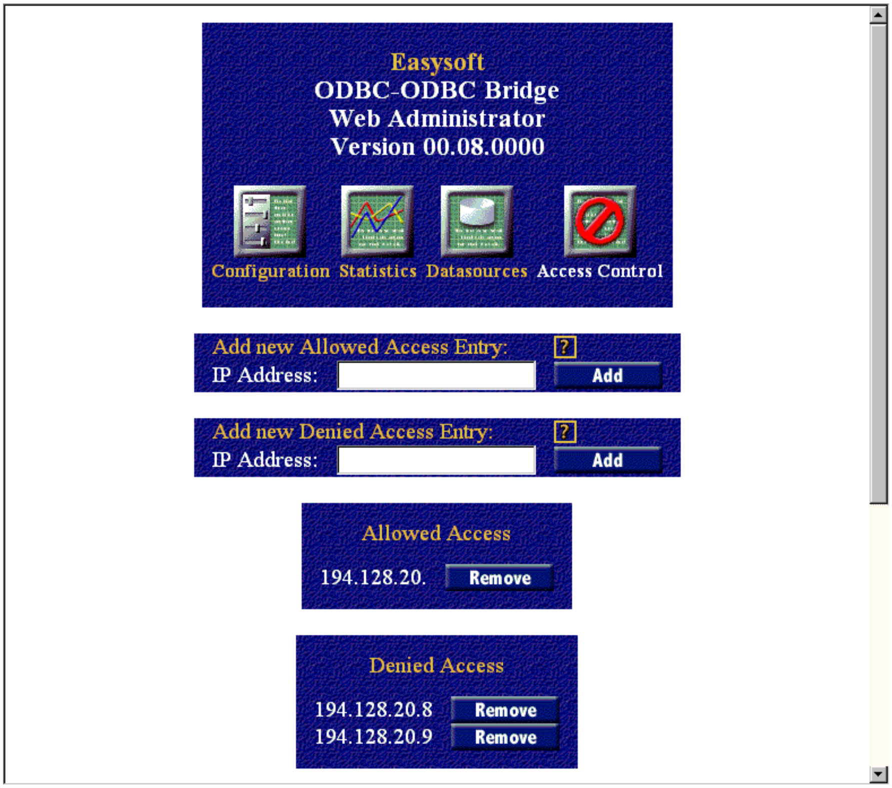
Port :	8888
HTTPPort :	8890
Timeout :	7200
RetryCount :	5
RetryPause :	3
HTTPAdmin :	Anne Example
MaxBookMarkSize :	32
Path :	C:\Program Files\Easysoft\Easysoft ODBC-ODBC Bridge Beta 0.8.0.0
Logging :	0
LogDir :	c:\temp
Flags :	2

Figure 4.4: The Change Configuration Screen of the HTTP Interface

6. If you want to make any changes to these parameters then you should refer to “Server Configurable Parameters (Windows and Unix)” on **page 4-92**. Make your changes and click **Submit**.

7. Click **Access Control** to view or change the set of hosts that are allowed to connect to the ODBC-ODBC Bridge. If you haven’t already

entered them, then you are prompted for the HTTPAdmin user name and password. The following screen appears:



Easysoft
ODBC-ODBC Bridge
Web Administrator
Version 00.08.0000

Add new Allowed Access Entry:
IP Address:

Add new Denied Access Entry:
IP Address:

Allowed Access

194.128.20.

Denied Access

194.128.20.8
194.128.20.9

Figure 4.5: The Access Control Screen of the HTTP Interface

If you want to change the contents of the access control lists, you should first refer to “Access Control (Windows and Unix)” on **page 4-98**.

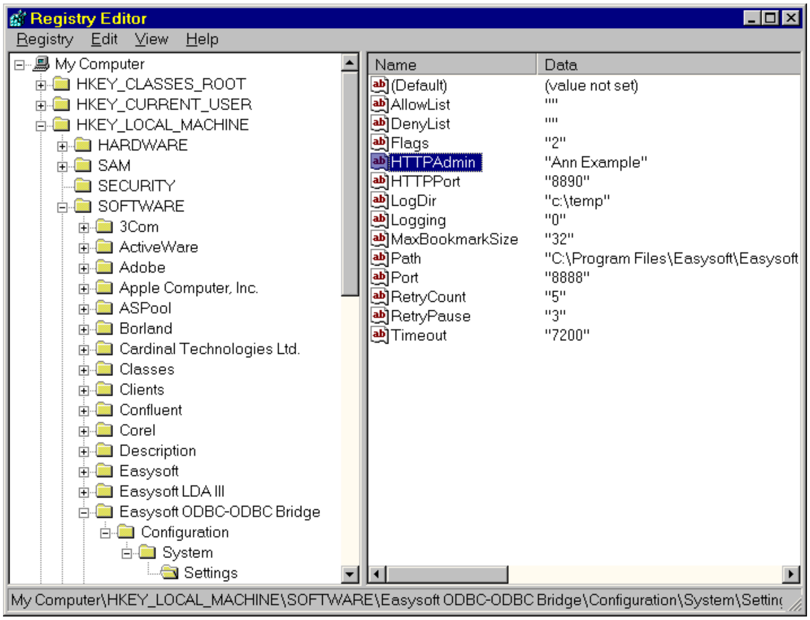
To add an IP address or set of IP addresses to the lists, type the address or address-stem into the **Add New Allowed Access Entry** or **Add New Denied Access Entry** box and click **Add**. At the confirmation screen, click the back button to return to this screen and see the changes you have made.

CONFIGURATION USING THE REGISTRY

For experienced users, the server can be configured by editing the registry directly. No users, and so no step-by-step guide will be provided here. “Server Configurable Parameters (Windows and Unix)” on **page 4-92** gives the names and values to use to achieve various results. These can be found in the registry under the key:

`\HKEY_LOCAL_MACHINE\SOFTWARE\Easysoft ODBC-ODBC Bridge\Configuration\System\Settings`.

Note that the bitmask values are stored as a *string* representing the *decimal value* of the bitmask.



```
My Computer
  HKEY_CLASSES_ROOT
  HKEY_CURRENT_USER
  HKEY_LOCAL_MACHINE
    HARDWARE
    SAM
    SECURITY
    SOFTWARE
      3Com
      ActiveWare
      Adobe
      Apple Computer, Inc.
      ASPool
      Borland
      Cardinal Technologies Ltd.
      Classes
      Clients
      Confluent
      Corel
      Description
      Easysoft
      Easysoft LDA III
      Easysoft ODBC-ODBC Bridge
        Configuration
          System
            Settings
```

Name	Data
(Default)	(value not set)
AllowList	""
DenyList	""
Flags	"2"
HTTPAdmin	"Ann Example"
HTTPPort	"8890"
LogDir	"c:\temp"
Logging	"0"
MaxBookmarkSize	"32"
Path	"C:\Program Files\Easysoft\Easysoft
Port	"8888"
RetryCount	"5"
RetryPause	"3"
Timeout	"7200"

My Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Easysoft ODBC-ODBC Bridge\Configuration\System\Settin

The access lists are stored as strings. IP addresses and address stems are separated by spaces.

Configuring the OOB Server under Unix

On Unix systems, the OOB Server is by default installed as a network service under `inetd`, but it may be configured in its stand-alone 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

The `inetd.conf` file entry instructs `inetd` to listen on behalf of the OOB Server. The services file specifies that it should listen on port 8888. When a client attaches to port 8888, `inetd` consults the file `inetd.conf` and finds the line relating to the service name (esoobserver). It runs the shell /bin/sh — given in inetd.conf — passing it the arguments on the rest of the line.

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 this example, the arguments on the line cause sh to run the Easysoft ODBC-ODBC Bridge server startup script /usr/local/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 in to the server executable. This notifies the server that it is running under inetd rather than at the command line. The server inherits the sockets from inetd and begins communicating with the OOB client. inetd returns to listening on port 8888 for any new connections.

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 Bridge server, then you will need to edit the configuration files by hand.

The port number appears only in /etc/services, but the service name appears 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

When the OOB Server is installed standalone, it has some subtly different properties. First, it no longer uses `inetd` to listen on the socket, so this approach might bypass any firewall implemented as a tcp wrapper. Secondly, you should note that if the server is started by a non-root user then it will always operate as that user. If a client specifies the LogonUser and LogonAuth attributes then they will be disregarded.

When the server is not present in the `/etc/inetd.conf` and `/etc/services` files (or is commented out), it needs to be started manually — or the root user can add the startup script to the kernel boot sequence.

To start the OOB Server in this mode you need to run it with the argument "standalone":

```
# path/esoobserver standalone
```

This notifies the server that it must listen on the socket itself, and fork its own child process when a connection is made.

Changing Server Configurable Parameters under Unix

There are a few DSN-independant configurable parameters in the server (See “Server Configurable Parameters (Windows and Unix)” on **page 4-92**). In Unix, the settings are held in the file named *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.

Server Configurable Parameters (Windows and Unix)

This section explains the configurable parameters in the server, which apply either to the server itself, or as a default value for all DSNs accessed via the server. In Windows these settings are held in the registry, and may be accessed through the HTTP interface (see “The HTTP Configuration Interface” on **page 4-81**). In Unix, they are held in a file — see “Changing Server Configurable Parameters under Unix” on **page 4-91** above.

The configurable parameters are shown below, each as a *key*=*value* pair, with its default value and an explanation. Keys are not case-dependent, so for example `Port`, `PORT` and `pOrT` are treated as the same key. Values are case-dependant if the operating-system is, so it is best to match case where possible.

- `Port=8888`

  This is the port at which the server will listen for connections, if run in standalone mode. This may need to be changed if you already have a service on port 8888.

- `HTTPPort=8890`

  This is the port at which the HTTP Configuration interface will listen, if enabled. See **page 4-81** for details on the interface.

- `Logging=0`

  This is a bitmask controlling the degree of logging to perform. It should only be used as directed by Easysoft support as it will slow the OOB down considerably if set. See “Tracing And Logging” on **page 4-94**.

- `LogDir=c:\temp` (Microsoft Platforms),
  `LogDir=/tmp` (Unix), or
  `LogDir=sys$scratch` (VMS)

  This setting determines the location in which to write the log files when logging is enabled.

NB `Logging` and `LogDir` can also appear in the DSN attributes when a client connects, overriding the settings in the server attributes.

- `Timeout=7200`

  This is a time period, measured in seconds, after which the OOB server will close a connection if it has not received any requests. The value should be high enough to allow users some idle time without losing their connections. The timeout is present because of an occasional problem with abnormal session terminations leaving orphaned server processes or threads on the host, using up resources indefinitely.

- `MaxBookmarkSize=32`

  ODBC 2.0 only supported fixed, 4-byte bookmarks. ODBC 3.0 deprecates fixed-width bookmarks and supports variable-width bookmarks. No upper limit is defined on the size of variable-length bookmarks but Easysoft have not yet encountered an application requiring more than 32 bytes. If you have such an application you should increase this setting. We would be grateful if you would also report the application to Easysoft so that we may update the default value.

- `Flags=2`

  To set the flags, choose your required options and place the sum of their corresponding values in the configuration dialog or file. For help choosing whether to set each flag, see the subsection “Server Flags” on **page 4-96**.

- `RetryCount=5`

  The number of times the OOB server should try to get a license slot for a new connection. If set to `0` or `1` then one attempt is made. If your system is having trouble creating threads then this is how many times the server will try to create a new thread before returning a failure.

- `RetryPause=3`

  The number of seconds between retries when trying to create a new thread to handle an incoming connection or create a new thread. Note that if your connection cannot be made, the OOB will take `RetryCount` x `RetryPause` seconds before giving up and reporting the failure.

TRACING AND LOGGING

The OOB contains comprehensive tracing and logging support — which should be turned *off* as it seriously impairs bridge performance. The client and server sides each carry a bitmask dictating what items should be logged. The actual logging mechanisms vary but the bitmask has the same meaning on any platform. The required value for this bitmask is derived by adding the values in the following table:

NB These flags, while correct at the time of writing, are subject to change without notice. It is unlikely that any of these flags will be removed, but it is likely that more will be added, and the precise functionality may change.

Hex	Decimal	Name	Effect
0x001	1	ESOOB_LOGENTRY	Log all ODBC API function entry points
0x002	2	ESOOB_LOGEXIT	Log all ODBC API function exit points
0x004	3	ESOOB_LOGRPC	Log EasyRPC operations
0x008	4	ESOOB_LOGSPECIAL	Special notifications (see below)
0x010	16	ESOOB_LOGVALUES	Log all bound parameter and column values possible
0x020	32	ESOOB_LOGINTERNAL	Internal functions (see below)
0x040	64	ESOOB_LOGDRIVER	Log calls to the ODBC driver
0x080	128	ESOOB_LOGDIAGS	Log ODBC diagnostic routines and results
0x100	256	ESOOB_LOGQUERIES	Log queries - text posted to SQLPrepare/SQLExecDirect
0x200	512	ESOOB_LOGCONNECT	Log Connection stage
0x400	1024	ESOOB_LOGATTRIBUTES	Log connection attribute setting and retrievals

ESOOB_LOGSPECIAL enables some miscellaneous diagnostic messages in the bridge. For example, the server will record a message when it detects it is connected to SQLServer or Access drivers.

Note also that ESOOB_LOGINTERNAL does nothing unless ESOOB_LOGENTRY or ESOOB_LOGEXIT is set.

Unix

The Logging option may be set in the file `/usr/local/easysoft/oob/server/esoobserver.ini`, e.g:

```
{Settings}
LogDir = /home/user100/logs
Logging = 511
```

This affects both the client and the server. The log files are called `esoobclient.log_`*PID* and `esoobserver.log_`*PID* and will appear in `/tmp`. Alternatively, if the log files cannot be created there, an attempt will be made to create the log files in the current working directory in which case they have "`fallback`" appended to the filenames. As with Windows, you can redirect the log file to a different directory by specifying the LogDir attribute (see the example above).

NB

Tracing may also be turned on for EasyRPC by setting the environment variable `ESRPC` to `c`, `a` or `ca`, where `c` stands for *calls*, and `a` stands for *arguments*. EasyRPC logging is sent to the file `easyrpc.log`.

Server Flags

This subsection documents the server flags in terms of their effect on the service. This is a bitmask with the following effects:

Val	Bitmask	Name	Effect
0	0000	(no name)	
1	0001	(Reserved, do not use)	
2	0010	ESOOB_HTTP	HTTP Enabled (Windows), No effect (Unix).

Val	Bitmask	Name	Effect
4	0100	ESOOB_MULTI_PROCESS	Create a new process on connection rather than a thread (Windows NT Only).
8	1000	ESOOB_NO_METADATA_BF	Disable automatic block-fetch mode for metadata calls.
16	10000	ESOOB_HIDE_HTTPADMIN	Disable automatic block-fetch mode for metadata calls.

ESOOB_HTTP (0x02) should be set if you wish to use the HTTP-based server reporting and configuration utility. This is *set* by default, as the HTTP interface is more flexible than editing the registry manually. The server configuration screen is protected by a user id and password screen, but you may wish to prevent users from accessing the utility altogether. If this is the case, reset this flag.

NB	Once the HTTP Configuration interface is disabled, you will need to edit the registry by hand to bring it back online.

ESOOB_MULTI_PROCESS (0x04) is only effective in Windows NT. It causes the service to create new *processes* rather than *threads* on receipt of new connections. The default is to use threads, as they are lighter on system resources. Though the ODBC standard states that drivers must be thread-safe, in practice some are not. If you have an ODBC driver that is not thread-safe then it is able to harm other connections unless you set this flag.

ESOOB_NO_METADATA_BF (0x08): As an efficiency-boosting measure, the OOB will normally use a block cursor to get read-only metadata. If you do not want this, for instance if your driver behaves badly with block fetches, then you can set this flag to inhibit the feature. Meta data is then retrieved one column at a time.

ESOOB_HIDE_HHTPADMIN (0x10): If set, then the name of the HHTP Admin user (see “The HTTP Configuration Interface” on **page 4-81**) will be hidden on the HTTP Admin interface. This is a security feature to prevent anyone using the HTTP Admin interface to learn an NT user name. By default this is off.

Access Control (Windows and Unix)

In addition to you system’s usernames and passwords, and those of the database management system, the OOB provides another layer of security with access control lists. Unix administrators will recognise 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 than those for `hosts.allow` and `hosts.deny`. Please read the rest of this section to avoid misunderstanding.

OOB Access Control takes the form of two lists of IP addresses. When a host attempts to connect to the ODBC-ODBC Bridge Server, access is only granted if:

- The `allowed` list is empty and the IP address does not appear in the `denied` list, or
- The IP address is in the `allowed` list and does not appear in the `denied` list.

The lists can be edited either in their relevant flat files (Unix servers) or via the HTTP Interface (Windows servers). Entries must be in the IP ‘dot’ notation, and those having fewer than four fields stand for all addresses matching those fields that do appear, e.g. `163.141.23.`

(note the trailing dot) matches all IP addresses from `163.141.23.0` to `163.141.23.255`.

Building a driver for use with the OOB Server (Unix)

The server is distributed as an executable linked with the shared object `libodbc.`*suffix*; where *suffix* is the file suffix your system uses for shared objects. Assuming unixODBC is installed — it is included in the OOB distribution — then `libodbc` is the unixODBC Driver Manager and the OOB will function ‘out of the box’. This section shows how to prepare a driver for use without a driver manager.

If you have no driver manager then you can create a `libodbc.`*suffix* object by linking the driver with the server object code to create a new server executable.

Caution!

> This is an extremely complex area and it is impossible to document it exhaustively. You should not attempt this unless you are comfortable with compiling and linking code, and you know what driver you have and that it is built properly.
>
> In any event, we *strongly recommend* making a backup of the executable `esoobserver`.

1. Back up the server executable:

```
$ cd install-path/easysoft/oob/server
$ cp esoobserver esoobserver.bak
```

In order to support a wide range of clients, the OOB contains all the ODBC 3.0 API and some of the ODBC 2.0 API too. For this reason you will probably need to create a few stubs. First, we must find out what symbols are missing.

You should verify the name and location of your desired driver. Refer to the documentation for the driver and satisfy yourself that it is correctly built and that all the necessary components are on your system.

The library `libserver.a` contains the actual server. The command line for building the driver is given below. Note the italics, which denote that you should substitute some specific text that depends on your installation.

2. Enter the following, from the `easysoft/oob/server` directory:

```
$ cc -o esoobserver -L../../lib -L. -Ldriver-
     path libserver.a -ldate -lsupport -lesrpc
     -lextra -ldriver
```

In a typical installation, *install-path* will be `/usr/local`, but it is in no way unusual for the `easysoft` directory to be located elsewhere. *driver-path* should be a path to the driver library, and the driver library is named `lib`*driver*`.a`, (the suffix varies depending on the platform and how the driver was built).

NB	The driver name given to the linker excludes the filename suffix and the preceding `lib`. For example, if your driver is in the file `libodbc.so` then the corresponding command-line option is `-lodbc`.

This command is likely to fail. If it completes successfully, then your chosen driver is very comprehensive and is probably a driver manager. In this case you have created the executable `esoobserver` in the directory *install-path*`/oob/server`, and the bridge is ready for configuring. Refer to **Chapter 3** for information on how to do this.

In most cases, however, this first attempt will fail with a number of "undefined reference" errors, as the OOB server library depends on a lot of symbols being present in the driver library. Functions such as

`SQLAllocEnv`, `SQLAllocConnect` and `SQLAllocStmt` are in ODBC 2.0 but not ODBC 3.0. They are likely to be absent from an ODBC 3.0 driver, but the bridge server must contain calls to them in order to support clients using the ODBC 2.0 API.

You can create stubs (functions that do nothing except return) for the missing names, and link them in with the driver and bridge server. As long as your client and driver support the same version of ODBC then the stubs should never be called and everything should function correctly.

3. Create a file in the `server` directory called, for example `stubs.c`, and open it in your editor.
4. Insert `#include "odbc.h"` at the top of the file.
5. The remainder of the file should consist of definitions for the missing functions. You will have to look in the header files to locate their prototypes, and you must give each a body:

```
{
  return SQL_ERROR;
}
```

So for illustration, you can expect to have a file looking something like this:

```
#include "odbc.h"
SQLRETURN SQL_API SQLAllocConnect(
                    SQLHENV EnvironmentHandle,
                    SQLHDBC *ConnectionHandle)
{
  return SQL_ERROR;
}

SQLRETURN SQL_API SQLAllocStmt(
                    SQLHDBC ConnectionHandle,
                              SQLHSTMT *StatementHandle)
{
  return SQL_ERROR;
}
```

6. Compile this into object code (substituting for `.o` according to platform):

   ```
   $ cc -L../client/include -o stubs.o -c stubs.c
   ```

7. Then build the server again, this time including the `stubs` object code.

   ```
   $ cc -o esoobserver -L../../lib -L. -Ldriver-
        path libserver.a stubs.o -ldate -lsupport
        -lesrpc -lextra -ldriver
   ```

   If your `stubs.c` file is still not correct, you will still get messages about functions beginning `SQL...().` If you get other messages then it is possible that your driver is not correctly compiled or linked.

INTERFACING

5

Preparing to write applications for the Easysoft ODBC-ODBC Bridge

This chapter shows you how to begin utilising the Easysoft ODBC-ODBC bridge in your own applications. The distribution comes with several example programs which can be found in the `examples` directory. These can serve as a tutorial to get you started writing OOB client applications. 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 the OOB. Examples shown were executed under Linux, and serve as a guide for all unix platforms. Windows developers should refer to the manual for their chosen development platform.

Chapter Guide

A Simple OOB Client in C

In “The Demo Client” on **page 3-69**, you compiled and/or ran a simple C program that connected through the OOB. 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 some SQL and passes it to a datasource for execution.

Before you can use the program you must edit the call to `SQLDriverConnect()` to indicate a valid DSN. You can then `make` the program, following the instructions on **page 3-69**.

This program, the source to the demo program, and the Makefile are enough to show you how to create C programs that connect through the OOB. If you need to know more, you should refer to an ODBC Programmer’s Manual.

Apache/PHP

Building Apache with PHP and the OOB allows your web content to be closely integrated with a database running on a separate machine to your web server. This is particularly useful because many offices have MS Access databases that would benefit from up-to-date information from the webserver, but the web server will typically be a linux box running Apache.

NB	Please read the INSTALL files in the PHP and Apache distributions before following these instructions as they may provide more up-to-date information specific to Apache and PHP.

Building PHP with Apache and OOB approximately follows this sequence (subject to changes due to the PHP or Apache distributions).

1. Download Apache.
2. Dowload PHP.

   NB Apache can be found at **http://www.apache.org**, and PHP at **http://www.php.net**.

3. Refer to the instructions in the `INSTALL` files in each distribution.

   The PHP `INSTALL` file gives step-by-step instructions for building PHP with the `mysql` database.

   To build PHP with the Easysoft ODBC-ODBC Bridge you must define the environment variable `CUSTOM_ODBC_LIBS`:

   ```
   # CUSTOM_ODBC_LIBS="-lesoobclient"
   # export CUSTOM_ODBC_LIBS
   ```

   Next, follow the PHP installation instructions replacing `--with-mysql` with `--with-custom-odbc=`*InstallDir*`/easysoft/oob/client`.

   (*InstallDir* is the OOB installation directory, not PHP's!)

4. When configuring Apache you need to define the environment variable `LDFLAGS`:

   ```
   # LDFLAGS="-L/base_oob_install_dir/oob/easysoft/c
       lient"
   # export LDFLAGS
   ```

...then follow the Apache installation instructions.

Caution!

Currently, OOB shared objects are not built with a dynamic linker run path so if you are on a Unix machine (except linux) you need to set

`LD_LIBRARY_PATH=$LD_LIBRARY_PATH:`*InstallDir*`/easysoft/oob/client:`*InstallDir*`/easysoft/lib`

...and export it before building or running Apache. *InstallDir* refers to the OOB installation directory (usually `/usr/local/`).

More information can be foud in the file *InstallDir*`/doc/Apache_PHP`.

Perl DBI DBD::ODBC

This section shows how to build Perl DBI with the DBD::ODBC module and connect to the Easysoft ODBC-ODBC Bridge client. It assumes that you have Perl and the Perl modules DBI and DBD::ODBC. These may be obtained from your nearest CPAN archive.

NB

If you are building DBD::ODBC version 0.20 then you must apply a patch (supplied with OOB in the `extras` subdirectory) to the file `Makefile.PL`. If you do not have the `patch` command then you need to apply the changes by hand. To do this, look for lines in the patch file beginning with `+`. Insert these lines in `Makefile.PL`.

Driver Managers

The DBD::ODBC 0.21 distribution contains the iODBC driver manager and the OOB will work with this driver manager. However, if you wish to use a driver manager then Easysoft recommend the unixODBC driver manager. Easysoft have close links with the unixODBC project

— the project is now maintained by an Easysoft developer. This brings the following benefits:

- We are able to provide better support for OOB running under unixODBC.
- If you find a problem in unixODBC it is much easier for us to get it fixed.
- At the time of writing, unixODBC supports ODBC 3.5 and is thread-safe whilst this is not true for iODBC.

NB DBD::ODBC prior to version 0.21 did not require a driver manager, but this changed in version 0.21. You still do not need a driver manager however to use DBD::ODBC with the Easysoft ODBC-ODBC Bridge: The bridge provides the necessary driver manager functionality.

Building Perl DBI, DBD:ODBC with OOB

1. Unpack DBI and build as per the instructions in the README file in the Perl DBI distribution.
2. Install DBI.
3. Unpack DBD-ODBC version 2.1 or later. Earlier versions can be made to work, with a patch. Consult the files in the easysoft/oob/docs directory for more information.
4. If you are using a driver manager, refer to the documentation for that driver manager and set up a Data Source Name (DSN) for the OOB client.

   – OR –

   If you are not using a driver manager, you must create an `odbc.ini` file for the client. In this file go all the DSN attributes for the OOB Bridge DSN.

5. Read the README accompanying the DBD:ODBC distribution. You must define and export the following environment variables to execute the test:

Name	Description
`ODBCHOME`	The installation directory of the OOB client: `/usr/local/easysoft/oob/client` or the directory where your driver manager was installed
`DBI_DSN`	The DBI data source, e.g. `dbi:ODBC:MYDSN`
`DBI_USER`	The username to use to connect to the database, if required
`DBI_PASS`	The password to use to connect to the database, if required

> **NB** By default, the ODBC-ODBC Bridge is installed in `/usr/local/easysoft` and the OOB client is in the subdirectory `oob/client`. For the default installation, `ODBCHOME=/usr/local/easysoft/oob/client`. The value for `DBI_DSN` depends on the datasource you have set in the `odbc.ini` file.

6. Type:

```
$ perl Makefile.PL
```

You may get some warnings of the form:

```
Warning: LD_LIBRARY_PATH
```

... which you should ignore for now.

```
$ make
...
$ make test
```

7. You may get errors like:

```
install_driver(ODBC) failed: Can't load 'blib/arch/auto/DBD
/ODBC/ODBC.so
```

This generally means that the Easysoft ODBC-ODBC Bridge client shared object cannot be found by the dynamic linker.

If you get errors like this you should ensure the path to `libesoobclient.so` is on the dynamic linker search path. The method for doing this depends on the OS you are running but often means adding a path to the environment variables `LD_LIBRARY_PATH` or `LD_RUN_PATH`. In linux you should normally add the path to the file `/etc/ld.so.conf` and run `ldconfig`. Try `make test` again.

8. You can check the server is responding using `telnet`.

   ```
   $ telnet server port
   ```

   ... where *server* and *port* are as specified in the `odbc.ini` file (or driver manager DSN). You should receive a string like the following (though it will differ from server to server):

   ```
   87FA9694x1:0:1998-1112-0000000001+01:NT^02:4.0.1381^03:Inte
   l^04:4^05:^06:<99> p^
   ```

   Once any problems with the server or the connection are resolved, any further problems should be easily identified by the ODBC error message output.

NB	The Perl tests are somewhat dependent on the ODBC database and driver you are using on the remote machine, for example, the MS Access ODBC driver does not have `SQLDescribeParam()`. This also impacts upon attempting to use perl bound parameters across the OOB to MS Access.

9. Once the tests succeed you can install Perl DBI:ODBC with:

   ```
   $ make install
   ```

Once Perl DBI:ODBC is installed, you can try the example files in the examples directory of the OOB install tree. There are also examples given in the Perl DBD::ODBC distribution but beware that at the time of writing, some of these were quite out of date, so they are liable to fail when there is nothing really wrong with your installation or build.

mxODBC

mxODBC is a database API for the Python scripting language which provides an interface to ODBC datasources. The Easysoft ODBC-ODBC Bridge (OOB) has been tested with mxODBC-1.0.1 and mxODBC-pre1.0.2 with Python-1.5.1.

When building mxODBC, there is a configuration section specifically for OOB which makes the build very straightforward. You should refer to the web page below and follow the installation instructions.

REF mxODBC and instructions for building it are available at **http://starship.skyport.net/~lemburg/mxODBC.html**.

Rexx/SQL

Rexx/SQL provides Rexx programmers with a consistent, simple and powerful interface to SQL databases. This section explains how to build Rexx/SQL with ODBC support provided by the Easysoft ODBC-ODBC Bridge and shows you how to connect a small test program.

This section assumes that you have an OOB service running with a test database on a remote machine, and that you have a local DSN set up for an OOB connection to a test database. Refer to “Connecting to the Bridge Client in Unix” on **page 3-65** to set up a local DSN.

REF

The Rexx/SQL home page contains download links, documentation, mailing list instructions and links to other Rexx resources.

The URL for Rexx/SQL's World Wide Web home page is **http://www.lightlink.com/hessling/**.

The Rexx/SQL anonymous ftp site is mirrored in the US at **ftp://ftp.lightlink.com/pub/hessling/REXXSQL**.

Easysoft used the Rexx interpreter ‘Regina’ to test the ODBC-ODBC Bridge with Rexx/SQL. Regina is available at **ftp://ftp.lightlink.com/pub/hessling/Regina/**.

BUILDING REXX/SQL WITH OOB

You will need a Rexx interpreter (such as Regina — see above reference) to run the Rexx/SQL test code and your Rexx programs using Rexx/SQL. You may use `./configure --help` in Rexx/SQL to see which Rexx interpreters are supported.

These instructions assume you are building from a source distribution of Rexx/SQL. You must have already installed the ODBC-ODBC Bridge before building Rexx/SQL. This is essential as Rexx/SQL needs an ODBC driver to compile and link with. Make a note of the installation path used when OOB was installed as you will need it when building Rexx/SQL.

1. Unpack the Rexx/SQL distribution.

> **NB** Read the INSTALL files in Rexx/SQL and OOB distributions. There is also a FAQ distributed with the OOB if you need further assistance.

2. Unpack Rexx/SQL and run

```
$ ./configure --help
```

...to locate the option required for your Rexx interpreter. For instance, with Regina you would use `--with-regina`. Depending on your interpreter you may need to add extra configure options.

3. You need to pass `configure` a minimum of:

- an interpreter option (e.g. `--with-regina`)
- `--with-easysoft-oob`
- `--with-dblibdir=`*clientpath*

  (where *clientpath* is the location of the OOB library and header files,) and

- `--with-dbincdir=`*clientpath*`/include`

If you installed the OOB in the default directory and are using Regina as the Rexx interpreter you might configure Rexx/SQL with:

```
$ ./configure --with-regina --with-easysoft-oob
      --with-dbincdir=/usr/local/easysoft/oob/client/includ
      e --with-dblibdir=/usr/local/easysoft/oob/client
```

4. Once Rexx/SQL is configured type

```
$ make all
```

...to build it.

5. You should then consult the section of the Rexx/SQL INSTALL file entitled "Testing the Installation".

To perform the tests in this section you need to define an OOB Data Source local to the machine where the OOB client and Rexx/SQL are installed.

To do this you must create an `odbc.ini` file containing the datasource (see `DSN_definition.txt` in the OOB distribution).

Once you have defined the local datasource you can try the Rexx/SQL tests, e.g:

```
$ ./rexxsql user=db_user pass=db_passwd data=rexx_sql_test
```

Running this against MS SQLServer you should see something like this:

```
Rexx/SQL Version: rexxsql 2.4 21 Jul 1999 UNIX ESOOB

Database Name:    Microsoft_SQL_Server

Database Version: 07.00.0255

Disconnect succeeded!
```

Rexx/SQL contains other test code such as `samples/tester.cmd`.

You can run this by defining and exporting the following environment variables:

Name	Description
`REXXSQL_DATABASE`	The ODBC data source name
`REXXSQL_USERNAME`	The username to use to connect to the database, if required
`REXXSQL_PASSWORD`	The password to use to connect to the database, if required

```
$ ./rexxsql samples/tester.cmd setup
$ ./rexxsql samples/tester.cmd
```

The `setup` argument creates the necessary tables for the test routine. Then run the same command without the setup argument. The `tester` program exercises Rexx/SQL and the OOB more thoroughly.

A

TECHNICAL REFERENCE

Technical Reference

ODBC Versions Supported

The Easysoft ODBC-ODBC Bridge supports most of ODBC 2.0 and ODBC 2.5, and all of ODBC 3.0. All the ODBC 2.0 and 2.5 API required to run Perl DBD:ODBC, PHP or mxODBC is present, but the full API of these deprecated versions is not supported as new applications will be written using the ODBC 3.0 API. The ODBC 3.0 API is supported in its entirety.

ODBC API Supported

This section documents the ODBC-ODBC Bridge API where it differs to the ODBC API.

MODIFICATIONS TO THE API

On non-Windows platforms `SQLDrivers` is implemented by the ODBC-ODBC Bridge even though it is a driver manager function. This is primarily for Perl DBD::ODBC 0.21 support. The ODBC-ODBC Bridge only returns details about itself and no other drivers.

`SQLBrowseConnect` is a slightly uncomfortable API in the context of an ODBC-ODBC bridge. 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 datasources (defined in your odbc.ini file or the registry) but these point to another real datasource on a remote machine.

For this release of the Easysoft ODBC-ODBC Bridge, SQLBrowseConnect only supports browsing the OOB data sources on the local machine. If the user then selects one of these to connect to, they will be prompted for 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 OOB will return a list of DSNs retrieved by calling SQLDataSources. The browse stops here and so the remote datasource must already be set up with sufficient information to allow a connection.

This implementation avoids complications with possible clashes of attributes between the OOB and the remote ODBC driver which would make it impossible to return a final connection string which allows a later connection without browsing.

Extra functions

The non-Windows version of the Easysoft ODBC-ODBC Bridge contains extra functions which are not part of the ODBC API. These were created as a substitute for the valuable functionality of

SQLDataSources which was not implemented in early pre-releases. They are listed below:

- void *enumerate_dsns(void *, unsigned long type); where type is 0=user DSNs and 1=System DSNs. Call this function to enumerate the list of DSNs in the odbc.ini file. The return value is used in subsequent calls to get_next_dsn() below. The return value will be NULL if no DSNs can be found (e.g. if there is no odbc.ini file).
- dsn_details_t *get_next_dsn(void *); This function returns a pointer to a dsn_details_t which contains all the details of the next DSN. You should pass the value returned from enumerate_dsns() as a parameter.

The ODBC-ODBC Bridge now supports SQLDataSources so these functions are deprecated.

A TECHNICAL REFERENCE

THE LICENSE MANAGER

B

Guide to the Easysoft License Manager

Products in the Easysoft Data Access 2000 range running on the Windows platforms all ship with the Easysoft License Manager, which provides a consistent interface for applying for, receiving and deleting licenses to use Easysoft software.

This appendix provides an overview and guides you through the task of adding licenses with the license manager.

Chapter Guide

Easysoft Licensing Overview

Easysoft software is set up so that it is possible to complete the installation process without having obtained a license. Some of our products can be used without a license, but most will not operate until you activate one of the two types of license:

A *trial license* is free, but comes with certain built-in limitations. Commonly, this will be a fixed expiry date or a limit on the number of concurrent sessions allowed. This license is designed for beta testers and customers wishing to evaluate Easysoft products.

A *purchase license*, as the name implies, is a license that you purchase from Easysoft. A purchase license has fewer restrictions and enables the software to be used in your production system.

NB	Only the license is purchased. The software product itself always remains the property of Easysoft.

THE LICENSE MANAGER AND LICENSE SERVER

The License Manager runs on the machine that you are running the software on. It handles license verification requests from Easysoft software.

The License Server runs on a machine at Easysoft. The License Manager on your PC contacts the License Server when you request or apply licenses. No marketing information is passed to Easysoft; only your contact details and license request identifier.

THE TRIAL LICENSE

The trial license is very simple. When you request one, the Easysoft License Manager generates a number based on the software you wish to evaluate and some other factors. This number is sent to the

Easysoft License Server which grants the license, unless for instance it is already in use or has expired.

THE PURCHASE LICENSE

To activate a Purchase License, you must commit to paying the purchase price. This is easily achieved via the Easysoft website **http://www.easysoft.com/**. Alternatively, you can phone, fax or email sales@easysoft.com — see **Appendix D**, “Contacting Easysoft,” for further details.

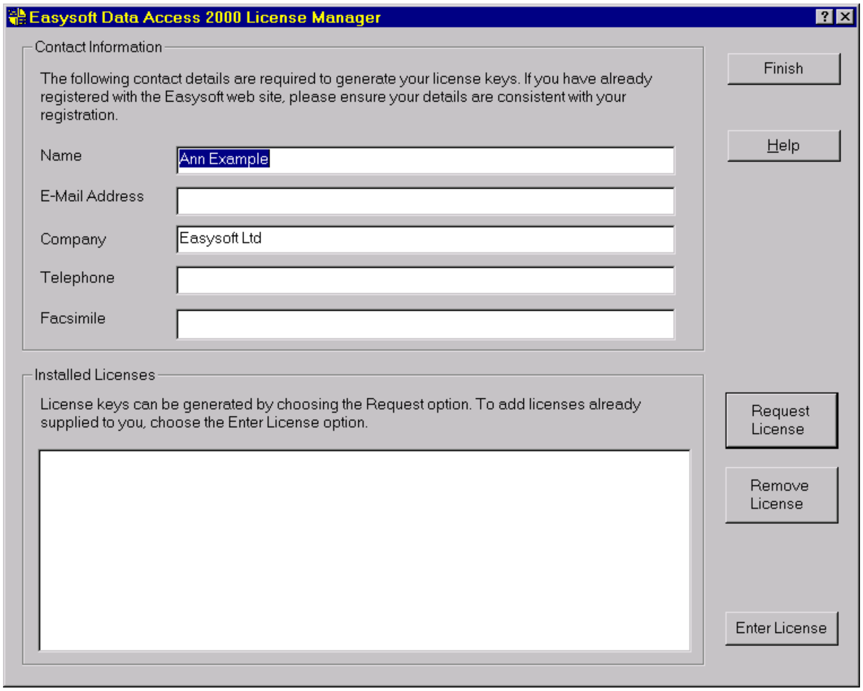
When you purchase a license, you are given an *Authorisation Number*. The Authorisation Number acts like a receipt — you give it to the Easysoft License Manager when you request a purchase license, and it allows the License Manager to verify the purchase you just made with the License Server.

Starting the License Manager in Windows

1. Select:

   Start > Programs > Easysoft >
   Easysoft Data Access 2000 Licensing > License Manager.

   The License Manager window appears.



Easysoft Data Access 2000 License Manager

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: Ann Example

E-Mail Address:

Company: Easysoft Ltd

Telephone:

Facsimile:

Installed Licenses

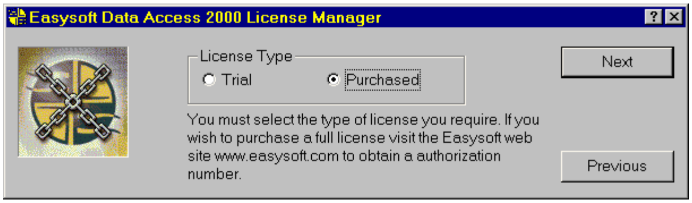
License keys can be generated by choosing the Request option. To add licenses already supplied to you, choose the Enter License option.

2. Enter your contact details if they are missing. You should include at least your name, email and company name. The **telephone** and **facsimile** fields are important if you want us to contact you by those methods.

   You are now ready to request or activate licenses. If you wish to activate a license you have already requested, then refer to “Activating an Existing License” on **page B-7**.

To Request a Trial or Purchase License

3. Click **Request License**. You are asked for a license type.



You must select the type of license you require. If you wish to purchase a full license visit the Easysoft web site www.easysoft.com to obtain a authorization number.

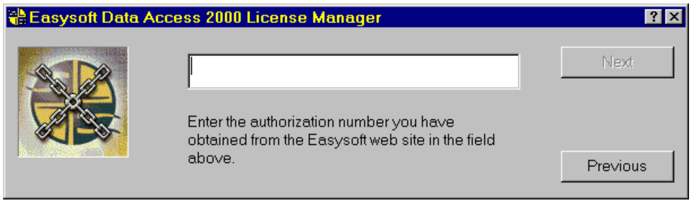
The next step depends on the type of license you want.

4. If you are installing the software on an evaluation basis you should click **Trial**, followed by **Next**. Skip to **step 6**.

– OR –

If you have purchased the software and obtained an authorisation code, you should select **Purchase** and click **Next.**

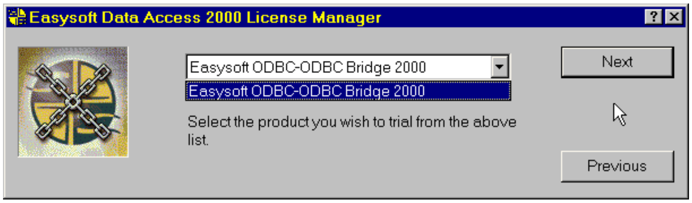
The License Manager requests your authorisation number.



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

5. Enter the authorisation number you received when you completed your purchase, and click **Next**. Skip to **step 7**.

6. The License Manager asks what software you are licensing.



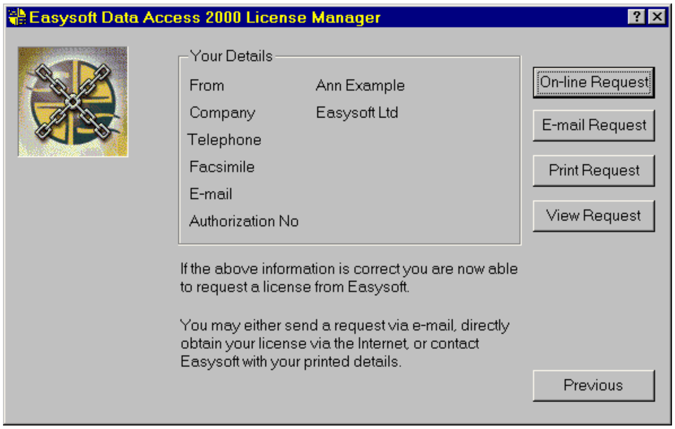
Easysoft ODBC-ODBC Bridge 2000

Easysoft ODBC-ODBC Bridge 2000

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

Check that `Easysoft ODBC-ODBC Bridge` is selected in the drop-down list, and click **Next**.

7. The license manager displays a summary of the information you entered and allows you to choose the method of applying for your license.



Easysoft Data Access 2000 License Manager

Your Details

From	Ann Example
Company	Easysoft Ltd
Telephone	
Facsimile	
E-mail	
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.

Four options are offered here:

- **On-line Request**: This is the quickest method of applying for a license. The License Manager transmits a network packet to the license server at the Easysoft office. The whole process is automatic and invisible.
- **E-mail Request**: You can use email to obtain your license. The license server at Easysoft replies immediately with your license number, and you should receive it without delay.
- **Print Request**: You can use this option to apply for a license by fax or mail. This is useful if you do not have an internet connection.
- **View Request**: You can obtain your license by telephoning Easysoft. The **View Request** button displays the license code you need to give to the Easysoft staff. This is the same as the information sent by the **On-line Request** button.

If you choose On-line request, then click **On-line Request** now. The license transaction is complete and you can skip to **step 5**.

– OR –

To use email, fax or phone, click E-mail Request, Print Request or View Request now.

NB	Though the on-line request is often the most efficient method, many networks' firewalls block the transmission. If you chose **On-line Request** and got a failure, try email instead.

8. Read the information window and close it. You are returned to the license manager window.

   You must wait until you receive a reply in order to proceed with the next step. Normally this will be in a matter of seconds, so you can go straight on to the next subsection — alternatively you can quit the License Manager and return later to activate the license.

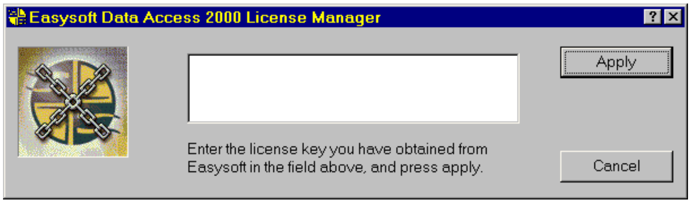
Activating an Existing License

When Easysoft receive a valid request for a license, the License Server generates a License Key. Normally you will receive this in an email. If you can execute the attachment in the email then it will automatically activate the licenses you applied for, and you can skip the rest of this section.

If you did not receive the key by email, or for instance your company policy forbids executing attachments, then you will have to activate the license manually.

1. When you receive your License Key(s), refer to “Starting the License Manager in Windows” on **page B-4** and start the license manager.

2. Click **Enter License**. The following dialog appears:



Enter the license key you have obtained from Easysoft in the field above, and press apply.

3. Enter your license key or keys in the box. If you received the license in an email, you can use Windows' Copy and Paste functions. You do not have to be very precise with the mouse, as this dialog will be quite tolerant of extra characters before and after the license keys.

NB	Many licenses can be activated at once in this way: simply paste all the keys into the text box in the dialog.

4. Click **Apply**.

   A dialog box pops up to confirm that the licenses have been added.

5. Click **OK**.

   You have activated the license or licenses. You can now quit the license manager and begin using the product.

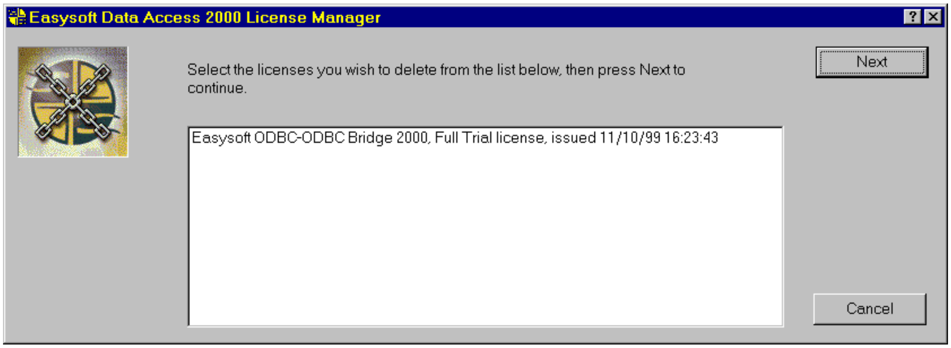
Removing a License or Licenses

To remove a license, for instance if it has expired, follow the steps in this section.

1. Refer to "Starting the License Manager in Windows" on **page B-4** and start the license manager.

2. Click **Remove License**.

A window appears containing a list of licenses for selection.



Select the licenses you wish to delete from the list below, then press Next to continue.

Easysoft ODBC-ODBC Bridge 2000, Full Trial license, issued 11/10/99 16:23:43

3. Select the licenses that you want to remove by clicking on each in turn.
4. Click **Next**.

   The License Manager pops up a confirmation box for each license you selected.
5. Click **Delete** or **Do Not Delete** for each license.

B THE LICENSE MANAGER

C

EASYSOFT LICENSE AGREEMENTS

End-User License Agreements for Client installation and Client-and-Server installation.

This Appendix contains two different license agreements. The first is applicable if you install only the OOB client and the second is applicable if you install the server, with or without the client.

Appendix Guide

End User License Agreement for Easysoft Client Software

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

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

SOFTWARE PRODUCT License

The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is Licensed, not sold.

1. Grant of License.

Easysoft client SOFTWARE. Your License grants you the following non-exclusive rights to the client SOFTWARE:

- You may only use the client SOFTWARE in conjunction with server SOFTWARE installed on a server owned by you. You may install and use the client SOFTWARE on an unlimited number of computers operating remotely or on one or more networks.
- The client SOFTWARE is made available in various formats compatible with different computer systems. You may install the client SOFTWARE for use with any of those systems at

any given time. You may make copies of the SOFTWARE in machine-readable form, provided that you reproduce all proprietary notices on the copy.

2. RESTRICTIONS.

You may not use the client SOFTWARE in conjunction with Easysoft server SOFTWARE not owned or Licensed by you or installed on a server which is not under your control.

- You must maintain all copyright notices on all copies of the SOFTWARE.
- Limitations of Reverse Engineering, Decompilation and Disassembly. You may not reverse engineer, decompile or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
- Rental. You may not rent, lease, sub-lease or incorporate the whole or any part of the use of the software in any other program or system to receive payment from third parties for the use of this software without the express written agreement of Easysoft.
- SOFTWARE Transfer. You may not transfer any of your rights under this agreement.
- SUPPORT Services. Easysoft may provide you with support services related to the SOFTWARE ("Support Services"). Use of support services is governed by the standard Easysoft Support Contract.

Any attempt to defeat or circumvent these SOFTWARE-enforced restrictions is a breach of this agreement.

3. TERMINATION.

Without prejudice to any other rights, Easysoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. In such event, you must destroy all copies of the SOFTWARE and all of its component parts.

4. COPYRIGHT.

All title and copyrights in and to the SOFTWARE and the accompanying written materials are owned by Easysoft. Unauthorised copying of the SOFTWARE, including SOFTWARE that has been modified, merged, or included with other SOFTWARE, or of written materials, is expressly forbidden. Any alteration, change or removal of any identification from the SOFTWARE, including but not limited to copyright, trademark or other notices or proprietary legends is forbidden.

5. WARRANTIES.

To the maximum extent permitted by applicable law, Easysoft expressly disclaims any warranty for the SOFTWARE. The SOFTWARE and any related documentation are provided "As Is" without warranty of any kind, either express or implied, including, without limitation, the implied warranties of merchantability or fitness for a particular purpose. The entire risk arising out of use or performance of the SOFTWARE remains with you.

6. LIABILITY.

To the maximum extent permitted by applicable law, in no event shall the owner be liable for damages for loss of data, profits or use of the SOFTWARE or for special, indirect, incidental or consequential damages except personal injury or death resulting from the owner's negligence or otherwise except as expressly provided herein. In the event of any violation of this Agreement or failure to comply with any of the terms and conditions set forth herein, or any other violation or infringement of any of Easysoft's rights, Easysoft shall have available all rights and remedies at law and in equity. Without limiting the foregoing, in the event of any such violation or failure to comply, then in addition to its other rights and remedies, if Easysoft so elects Easysoft shall have the right to seek damages of not less than £1,000,000.00.

7. MISCELLANEOUS.

This Agreement constitutes the entire agreement between the parties with respect to the subject matter hereof. This Agreement supersedes any written or oral agreement between Easysoft and you. This Agreement shall apply, unless Easysoft and you have agreed in writing to variations and additions by an authorised representative. Any modification of the terms of this Agreement must specifically refer to this Agreement and must contain the written signature of an authorised representative of Easysoft and you. No representative of Easysoft has been authorised to make any representation, warranty or promise not contained in this Agreement. In the event that any provisions of this License are held to be illegal or otherwise unenforceable, the entire License shall fail because of the severance of unenforceable provisions. This Agreement is governed by and interpreted in accordance with English Law.

End User License Agreement for 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 License Agreement ("Agreement").

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

SOFTWARE PRODUCT License

The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is Licensed, not sold.

1. Grant of License.

Easysoft Server SOFTWARE. Your License grants you the following non-exclusive rights to the Server SOFTWARE:

- You may install the server SOFTWARE on a single computer or network. NOTE: A License for the SOFTWARE may not be granted if no purchase consideration for fees has taken place between you and the supplier for the server SOFTWARE. You may copy the Licensed software for back up purposes. A License for the SOFTWARE may not be shared or used concurrently on different computers.

- Evaluation SOFTWARE. Easysoft may grant you the right to use the SOFTWARE on a royalty-free basis for evaluation purposes only and for a period of not more than 30 days. At the end of this period if you do not purchase the SOFTWARE, you must destroy all copies of the SOFTWARE supplied to you, including copies installed on any computer, and all related documentation. You disclaim all warranties, either express or implied, including but not limited to implied warranties of merchantability, fitness for a particular purpose and non-infringement, with respect to the Evaluation SOFTWARE and the accompanying written materials. In no event will Easysoft be liable to you for damages, including any loss of profits, lost savings or other incidental or consequential damages arising out of your use or inability to use the Evaluation SOFTWARE.

Easysoft client SOFTWARE. Your License grants you the following non-exclusive rights to the client SOFTWARE:

- You may only use the client SOFTWARE in conjunction with server SOFTWARE installed on a server owned by you. You may install and use the client SOFTWARE on an unlimited number of computers operating remotely or on one or more networks.
- The client SOFTWARE is made available in various formats compatible with different computer systems. You may install the client SOFTWARE for use with any of those systems at any given time. You may make copies of the SOFTWARE in

machine-readable form, provided that you reproduce all proprietary notices on the copy.

- You disclaim all warranties, either express or implied, including but not limited to implied warranties of merchantability, fitness for a particular purpose, and non-infringement, with respect to the Client SOFTWARE and the accompanying written materials. In no event will Easysoft be liable to you for damages, including any loss of profits, lost savings or other incidental or consequential damages arising out of your use or inability to use the Client SOFTWARE.

2. Restrictions.

You may not use the client SOFTWARE in conjunction with server SOFTWARE not owned by you or not under your direct control. Any attempt to defeat or circumvent these SOFTWARE-enforced restrictions is a breach of this agreement.

- You must maintain all copyright notices on all copies of the SOFTWARE.
- Limitations of Reverse Engineering, Decompilation and Disassembly. You may not reverse engineer, decompile or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
- Rental. You may not rent or lease or lend the SOFTWARE.
- SOFTWARE Transfer. You may not transfer your rights under this agreement.
- Support Services. Easysoft may provide you with support services related to the SOFTWARE ("Support Services").

Use of Support Services is governed by the Easysoft Support Contract.

- Termination. Without prejudice to any other rights Easysoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. In such event, you must destroy all copies of the SOFTWARE and all of its component parts.

3. COPYRIGHT.

All title and copyrights in and to the SOFTWARE and the accompanying written materials are owned by Easysoft. Unauthorised copying of the SOFTWARE, including SOFTWARE that has been modified, merged, or included with other SOFTWARE, or of written materials, is expressly forbidden. Any alteration, change or removal of any identification from the SOFTWARE, including but not limited to copyright, trademark or other notices or proprietary legends is forbidden.

4. LIMITED WARRANTY.

Unless otherwise expressed in this agreement, Easysoft warrants that the SOFTWARE will conform substantially to the documentation in effect for the SOFTWARE when delivered, for a period of sixty (60) days (the warranty period) from the date of delivery to you. Easysoft warrants that it has a right to grant the License. Providing that the SOFTWARE is used on the computer and with the operating system for which it was designed and is used in the manner set out in the user manual with the SOFTWARE and hardware equipment specified in the SOFTWARE product description, any demonstrable errors in the SOFTWARE or related documentation notified to Easysoft within the warranty period will be corrected within a reasonable time or at Easysoft's option provide or authorise a refund of the price of the

SOFTWARE (against return of the SOFTWARE, and it's related documentation). Easysoft makes no other warranties, express or implied, on any other Licensed SOFTWARE, including but not limited to all implied warranties of merchantability and fitness for a particular purpose.

5. LIMITED LIABILITY.

Unless otherwise expressed in this agreement Easysoft's entire liability to you for any cause whatsoever is limited to the amount paid for the SOFTWARE that is the subject matter of the cause of action. In no event shall Easysoft be liable for damages for loss of data, profits or use of the SOFTWARE or for special, indirect, incidental or consequential damages except personal injury or death resulting from Easysoft's negligence or otherwise except as expressly provided herein.

6. PATENT AND COPYRIGHT INDEMNITY.

Easysoft will defend or settle at it's expense any action brought against you to the extent that is based on a claim that the SOFTWARE, used within the scope of this agreement, infringes a copyright or existing patent. Easysoft will pay any cost, damages and legal fees awarded against you in such actions, which are directly attributable to such claim, provided that you notify Easysoft promptly in writing of the claim, that Easysoft is given complete authority and co-operation by you and all information necessary for it to conduct the defence or settlement of such claim. Should the SOFTWARE become, or in Easysoft's opinion be likely to become, the subject of a claim for infringement of a copyright or patent, Easysoft may procure for you the right to continue to use the SOFTWARE or replace or modify the SOFTWARE to make it non-fringing. In such an event, Easysoft shall have no further liability to you in respect thereof. In the event of any violation of this Agreement or failure to comply with any of the terms and conditions set forth herein, or any other violation or infringement of any of Easysoft's rights, Easysoft shall have available all rights and remedies at law and in equity. Without limiting the foregoing, in the event of any such violation or failure to comply, then in addition to its other rights and remedies, if Easysoft so elects Easysoft shall have the right to seek damages of not less than £1,000,000.00.

7. MISCELLANEOUS.

This Agreement constitutes the entire agreement between the parties with respect to the subject matter hereof. This Agreement supersedes any written or oral agreement between Easysoft and you. This Agreement shall apply, unless Easysoft and you have agreed in writing to variations and additions notwithstanding any proposed variations or additions, which may appear in any purchase order to Easysoft by you. Any modification of the terms of this License must specifically refer to this License and must contain the written signature of an authorised representative of Easysoft and you. No representative of Easysoft has been authorised to make any representation, warranty or promise not contained in this License. In the event that any provisions of this License are held to be illegal or otherwise unenforceable, the entire License shall fail because of the severance of unenforceable provisions. This Agreement is governed by and interpreted in accordance with English Law.

D

CONTACTING EASYSOFT

By Phone or Email

You can request licences, report problems and obtain support by email, fax, web, mail or telephone.

For information on our current and emerging products, prices and terms, contact **Sales**:

Tel:	+44 (0)113 222 0480
Fax:	+44 (0)113 222 0580
Email:	sales@easysoft.com

For after-sales support, and advice on installing or configuring our beta releases, contact **Support**:

Tel:	+44 (0)113 222 0450
Fax:	+44 (0)113 222 0550
Email:	support@easysoft.com

To obtain a license key upon expiry of any trial period or beta release, contact **Licensing**:

Tel:	+44 (0)113 222 0450
Fax:	+44 (0)113 222 0550
Email:	licensing@easysoft.com

All lines are open 8:30-5:30 UK time, Monday to Friday, except for public holidays. UK time is GMT+1 from April to September, and GMT during the winter months. The current UK time is shown at the bottom of every Easysoft web page.

Other Methods

The **Easysoft Web Site** (**http://www.easysoft.com/**) provides a round-the-clock service, offering free patches, upgrades, documentation and beta versions of Easysoft products. It can also be used to contact us.

As an alternative to email, you may wish to complete the online **Information Request Form**. This is quicker than email, and automatically enters your query in our database so that we can respond more effectively.

If you would prefer us to phone you as soon as someone is available, you can click the **Contact Me** button.

Should you wish to contact Easysoft by mail, please use the following postal address:

Easysoft Limited
3 The Embankment
Sovereign Street
LEEDS
LS1 4BJ
United Kingdom
Tel: +44 (0) 113 222 0400
Fax: +44 (0) 113 222 0500

Information to Include When Contacting Easysoft

When requesting support please include the following information where possible:

- your name and your company name,
- the name and version number of the Easysoft product,
- the operating systems and their version numbers, and
- indicate whether similar problems have arisen previously.

Registering at the Website

To download final-release products, you must fill in a registration form online. Not only does this allow you access to the download area, but it also makes the website easy to use by enabling the **Contact Me** button, and automatically completing certain fields on the information request form and download request forms. Registration is *not* mandatory for patches, documentation and Beta releases, but is nevertheless recommended.

D CONTACTING EASYSOFT

E

GLOSSARY

API

Application Programmer Interface. An API is a published set of function calls and constants allowing different programmers to utilise 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 datasource.

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 datasource, but provision is made for the driver to negotiate with the application or the user for any missing information.

Datasource

In the context of ODBC, a datasource is a database or other data repository coupled with an *ODBC driver*.

DLL

Dynamic Link Library. Windows' mechanism for shared object code. See also *Shared Objects*.

to **Download**

To retrieve data from a remote machine (or "the internet") to your local machine. Mechanisms for achieving this include FTP and the World Wide Web.

Driver

See *ODBC driver*.

Driver Manager

Software whose main function is to load ODBC drivers. ODBC applications connect to the Driver Manager and requests 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 quite simply a name associated with an ODBC data source. Driver Managers, such as unixODBC or the one that ships with Microsoft Windows, 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 MS 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 the HTTP Interface.

HTTP Interface

The OOB server under Windows runs a lightweight webserver which can be used to display and amend certain server statistics and parameters.

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 now becoming available on a wide number of platforms, and Easysoft's 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 standardised view of the data to ODBC.

Operating System

Academics still haven't agreed on the actual definition of an operating system, but a working definition can be:

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 centralised component of some task, for example extracting information from a corporate database. See *client-server*.

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 Datasource

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.

Table

A data set in a relational database, composed of rows and columns. For example:

software	
vendor	name
Easysoft	Easysoft ODBC-ODBC Bridge
MeSoft	My ODBC Client Application

This table has two columns; `vendor`, and `name`. It has two rows: one corresponding to Easysoft ODBC-DOBC Bridge, and the other corresponding to MeSoft's ODBC client software. 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. Compare *System Data Source*.

INDEX